On the Use of Functionals on Boundaries in Hierarchical Models of Object Recognition

by

Ian Hyla Jermyn

A dissertation submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

Department of Computer Science

New York University

September 2000

Davi Geiger

©Ian Jermyn

All Rights Reserved, 2000

If there were any real proof that the sun is in the centre of the universe and the earth in the third heaven, and that the sun does not go around the earth, but the earth round the sun, then we would have to proceed with great circumspection in explaining passages of scripture which appear to teach the contrary, and rather admit that we did not understand them, than declare an opinion to be false which is proved to be true. As for myself, I shall not believe that there are such proofs until they are shown to me. Nor is it proof that, if the sun be supposed to be at the centre of the universe and the earth in the third heaven, everything works out the same as if it were the other way around.

Cardinal Roberto Bellarmino, Master of Controversial Questions at the Collegio Romano, in a letter of 12 April 1615 to Paolo Antonio Foscarini, Carmelite monk, replying to an enquiry about the truth of the Copernican system (*Opere*, Vol. 12, p. 165).

* * * * * * * * *

Indeed, someone who does philosophy or psychology will perhaps say "*I* feel that I think in my head". But what that means he won't be able to say. For he will not be able to say *what* kind of feeling that is; but merely to use the expression that he 'feels'; as if he were saying "*I* feel this stitch *here*". Thus he is unaware that it remains to be investigated what his expression "I feel" means here, that is to say: what consequences we are permitted to draw from this utterance. Whether we may draw the same ones as we would from the utterance "I feel a stitch here".

Ludwig Wittgenstein, *Remarks on the Philosophy of Psychology*, Vol. 1, # 350, Basil Blackwell, Oxford, 1980. Translated by G. E. M. Anscombe.

DEDICATION

To Leslie, who has shown me what is truly important,

and

In Memory of my Beloved Grandmother,

Elizabeth Norah Jermyn.

Acknowledgements

My greatest thanks go to Davi Geiger who, after indulging my inclination to other things, showed me the computer vision light. As well as all the advice, encouragement, psychotherapy, and stern admonishments that he has given so well and so freely over the years, we have also become friends. I thank him very much for everything.

Hiroshi Ishikawa is my great and good friend. We have shared many things, particularly our adventure in Rio, during which, fuelled by cafezinhos and caipirinhas, we began the collaboration that produced much of the work in this thesis. I thank him very much for his friendship, his humour,and his thought.

I would like to thank Pete Wyckoff, who befriended me when I was a novice Englishman in New York, who introduced me to Leslie, and who has been a dear and steadfast friend ever since. My life now would be quite different without him.

Thank you to Fabian Monrose for being my fellow un-American and for sharing my love of music. When New York becomes too much, I know I can always rely on him for sympathy and great chicken.

When I visited India with Leslie in 1995, Laxmi Parida invited us into her home and showed us the beauty of her country. A growing friendship was sealed. Thank you so much to her for being a listening ear, a stimulus to thought and a deadly pool opponent.

Thank you to Ken Pao for his friendship and gentle company in the office and outside it.

Au revoir to all these good friends. Remember that Antibes is a good place to visit.

Thank you very much to David Jacobs, who besides being my kindly boss during the summer at NECI, has acted as a second advisor. His calm support has been a great help.

Thank you to Ernie Davis for his perhaps unknowing encouragement to me when I needed it. I apologize to him for the lack of real AI in this thesis.

Thank you to Nava Rubin, for serving on my thesis committee, and for providing me with food for thought through her work.

Thank you to Alan Siegel for his attempt to convince me of the error of my ways.

Thank you to everyone at Courant, particularly Anina, Rosemary and Lourdes, who are continually friendly and helpful and kind.

I would like to thank the Instituto de Matemática Pura e Aplicada in Rio de Janeiro for their generous hospitality during the above-mentioned sojourn. I will never forget the monkeys and butterflies and the steamy sounds of the tropical forest floating through my office window.

Finally, thank you to my family, my parents Richard and Leonie, and my brother and sister Phil and Anna, for always being there, quietly supporting and encouraging me.

New York City, September 3, 2000

Abstract

Object recognition is a central problem in computer vision. Typically it is assumed to follow a sequential model in which successively more specific hypotheses are generated about the image. This is a rather simplistic model, allowing as it does no margin for error at any point. We follow a more general approach in which the various representations involved are allowed to influence one another from the outset. As a guide and ultimate goal, we study the problem of finding the region occupied by human beings in images, and the separation of the region into arms, legs and head. We approach the problem as that of defining a functional on the space of boundaries in images whose minimum specifies the region occupied by the human figure.

Previous work that uses such functionals suffers from a number of difficulties. These include an uncontrollable dependence on scale, an inability to find the global minimum for boundaries in polynomial time, and the inability to include region as well as boundary information. We present a new form of functional on boundaries in a manifold that solves these problems, and is also the unique form of functional in a specific class that possesses a non-trivial, efficiently computable global minimum. We describe applications of the model to single images and to the extraction of boundaries from stereo pairs and motion sequences.

In addition, the functionals used in previous work could not include information about the shape of the region sought. We develop a model for the part structures of boundaries that extends previous work to the case of real images, thus including shape information in the functional framework. We show that such part structures are hyperpaths in a hypergraph. An 'optimal hyperpath' algorithm is developed that globally minimizes the functional under some conditions.

We show how to use exemplars of a shape to construct a functional that includes specific information about the topology of the part structure sought. An algorithm is developed that globally minimizes such functionals in the case of a fixed boundary. The behaviour of the functional mimics an aspect of human shape comparison.

Contents

List of Figures

List of Tables

CHAPTER I

SEEING MACHINES

A background is drawn for the work. The study of vision is difficult both philosophically and practically, but the notion of seeing machines clarifies the issues somewhat. A definition of a visual system as a module of a seeing machine is given, and this necessitates a discussion of image semantics as the appropriate output of a visual system. The ideas discussed are formalized using probability theory and working assumptions used to render the problem tractable. We then consider briefly what it means to test a visual system empirically.

THE nature of vision is obscure. To a great extent this reflects the difficulties associated with any discussion of mental phenomena, whether in the biological/psychological sciences or in computer science. Indeed the very use of the word phenomena here is misleading. What we refer to as mental phenomena are exclusively experiences of ourselves, unless we count particular physical and chemical measurements that may be made on our brains and whose connection to the first kind of mental phenomena is largely unknown. These experiences are not phenomena in the same sense that the behavior of a falling object is a phenomenon. Others do not observe my 'mental phenomena'. They may hear me speak as if I have observed something, but we do not observe ourselves as we observe a physical event or even as we observe others, except metaphorically. It is not clear what we mean when we say that we 'see' something or that we 'recognize' an object, once we step outside the normal realms of discourse and attempt to analyze such statements in the abstract. For example, what does it mean to ask the questions "do we recognize every object in our field of view?" or "do we see every object in our field of view?"? Avoiding the dilemmas and confusions raised by these issues is not always easy.

By way of contrast, computer vision is the attempt to construct *seeing machines*. In full generality, a seeing machine is any machine that uses images to help accomplish a task. Such tasks are extremely varied. They range over almost all of human and animal activity: counting widgets passing by on

a conveyor belt; navigating through a complex environment; extracting the region corresponding to a human being in an image; animation; copying a design; handwriting recognition; and on and on. Human beings allegedly devote a third of the volumes of their brains to visual processing, which gives some indication of the problems facing computer vision. Nevertheless, by approaching the study of vision in this operational way, it is to be hoped that we can avoid the philosophical concerns mentioned in the first paragraph, and eventually shed some light on what we are talking about when we discuss human vision, as well as constructing useful technology along the way.

The first thing we will do however, is to make a simplifying assumption that reduces the operational content of our model. We will postulate a separation between those parts of the machine that deal with the images themselves and those parts that perform other tasks such as planning or locomotion. The picture is of a 'module' (called the *visual system*) that takes images as input (the images are made available according to a plan formulated elsewhere in the machine), and that produces as output statements about the image. Such a picture has advantages and disadvantages. On the positive side, it is a useful abstraction since we are not forced to contemplate general intelligent behaviour in addition to the already formidable difficulties of image understanding, and it opens the possibility of discovering task-independent methods. On the negative side, the separation means that we must now test the performance of the visual system independently of a specific task. In what could such a test consist? We are forced to refer the notion of image understanding to human performance, since that is the only visual system to whose output we have access.

1. Image Semantics

In performing a given task, the images used by the seeing machine will be endowed with a semantics. This semantics encodes what the seeing machine as a whole does with the images it acquires: what consequences it can draw from these images. A semantics can be thought of as a collection of statements about the image that are true. In general the semantics will clearly depend on the task. The job of the visual system is to output a statement from the semantics on receiving an image as input.

In order for the semantics to be testable in any meaningful way, the relevant people must agree on the statements in it: *ground truth* is established by human consensus. This may be because the semantics is agreed upon for a specific type of image and task, for example a blueprint, but often this is not the case. For example, the statement that there is a black rectangle at such and such a location in figure 1 is unlikely to produce disagreement among observers. On the other hand, the statement that this image is a picture of a

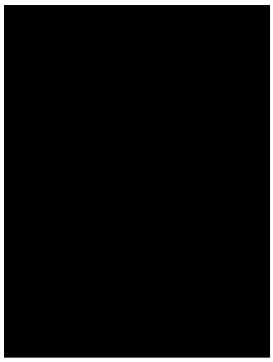


FIGURE 1. A black rectangle or a book?

black book might well, and yet it is not an unreasonable interpretation of the image. While this may seem to subjectivize the notion of the meaning of an image, in practice it is all that we have once we separate visual understanding from task performance. In the future, given a theory as to why we divide the world into the objects and concepts that we use (such a theory is not inconceivable: perhaps there is an informationally optimal way to do this, to which human understanding is an approximation), this situation might be changed. In the meantime, human consensus is what we mean by image understanding.

In order to compare two visual systems, we must have not only the notion of ground truth provided by human consensus, but also a notion of how 'close' to correct a given statement is. Given the output of a visual system on a particular image, this latter notion (an *evaluation function*) will compare the statement to the image semantics and output a real number, the evaluation of the output. Two visual systems can then be compared by, for example, using a probability distribution of possible inputs and computing the mean evaluations. The evaluation function is not given *a priori*. It too must be agreed upon, and will in general be task-dependent. In fact, in a typical task the evaluation function will depend upon a number of other factors that only logically become available to us once we consider the task itself. For example, the resources needed for the visual system to output its statement might be extremely important in reality, and may offset the accuracy of the

result. These factors are completely task-dependent and we do not consider them further except to ensure that they are not prohibitive (for example, an algorithm that takes time exponential in the size of the input).

It is hard to give a clearly defined semantics for many images. For example, depictions of real scenes can be given a semantics by making statements about possible scenes prefixed by "If a real scene had generated the image, then in that scene . . . ". The problem is that in some cases there may not be enough consensus to render such statements free of their dependence on the speaker. One way is to choose a semantics based on the author's intent, but this may not have been expressed in enough detail to be useful. More pragmatically, statements about such images must be referred back to the task that generated them. The successful (or otherwise) completion of the task will then be the best measure of the correctness of the visual system.

Rather than go into the details of different types of image semantics, we will restrict ourselves immediately to the case of *real* images, which is to say those images formed by a camera or cameras (including the case of biological vision). By the *scene* we mean the world including the camera(s). The scene itself has a semantics (which may differ from person to person, and certainly from culture to culture), which can include statements about time and about the relations of the views of different cameras (for example, "the volumes of the world viewed by these two cameras intersect"). This semantics may be translated into a semantics for the image by prefixing every statement in the scene semantics by "In the scene that generated this image . . . ".

The semantics of the scene must be testable in the same sense as discussed above for it to generate a testable image semantics. In many cases there is no problem. People will agree by and large on many things. The physics of the scene for example is agreed upon, including the geometric structure and motion of the objects and surfaces in the scene. Thus questions such as those asked in stereo vision or optical flow computation can be tested with reference to this semantics. People will also identify the human beings in the scene and the volumes occupied by them reliably, so that statements about object recognition can also be tested. Other properties, for example human facial expression, are less likely to produce unanimity, and again a referral back to the practical application may be necessary.

1.1. *Internal Representation of Images*

Internally to the visual system, for the purposes of model building and computation, an image will be represented by a map from some domain to some co-domain. We give some examples. A *single image* is a real-valued function on a domain $\mathcal{D}$ in the plane. An image may be $\mathbb{R}^3$-valued in the case of coloured images, and it can be more exotic, such as S^1-valued if we have

a phase image for example. We can consider n such images at once, which we need to do in the case of stereo for example, in which case the image is a product map $\mathcal{D}^n \to \mathbb{R}^n$ (or alternatively $\mathcal{D} \times \mathbb{N} \to \mathbb{R}$). (We could also consider products of images with different domains.) In the case of motion, we deal with maps from $\mathcal{D} \times \mathbb{R} \to \mathbb{R}$, where the domain now includes time. It is not inconceivable that the domain might also be topologically more complicated: a 360° view for example, or a 4π solid angle view. Of course we can also combine these and any other possibilities in various ways.

2. Probabilistic Models

In order to begin to formalize these notions, we turn to probability theory. Probability theory is the extension of logic to uncertain statements, or in other words to cases in which our state of knowledge can be described no longer by 1 when we know something to be true, or 0 when we know it to be false, but must be described by the interval $[0, 1]$. Despite many attempts to devise alternative pictures, probability theory remains in practice the pre-eminent method, because it seems to be the only system that satisfies certain simple desiderata of consistency.

2.1. *Decision Theory*

Suppose we have an hypothesis space $\mathcal{H}$, and a data space $\mathcal{D}$, and probability distributions $\Pr(D = d|H = h)$, which describes our knowledge of how the model represented by the hypothesis generates the data, and $\Pr(H = h)$, which describes our prior knowledge of the likelihood of the hypothesis h.[1] Then we can form the posterior probability that hypothesis h is correct given data d and our prior knowledge using Bayes theorem: $\Pr(H = h|D = d) = \Pr(D = d|H = h)\Pr(H = h)/\Pr(D = d)$.

If we wish to make a decision as to which hypothesis to pick (and use in any subsequent adventures), we must choose a *loss function*. This is a function L on $\mathcal{H} \times \mathcal{H}$, whose value $L(h, h')$ gives the loss consequent on assuming that the true hypothesis is h when it is in fact h'. The choice of a loss function is entirely disjoint from the probability calculation. Two different uses of the same data and same hypotheses might have different loss functions, and hence the decision we would make about which hypothesis to use might change with the situation. Even given a loss function, there are still some choices we can make as to what to do with it. We can choose to minimize the maximum

[1] By and large we will use calligraphic letters for spaces, capital letters for variables over spaces, and lower case letters for points in spaces. We will often abbreviate the notation $\Pr(H = h)$ by $\Pr(h)$ where the context makes it clear what is meant. We treat every object as an arrow, so that function composition will be denoted by juxtaposition. The only exception will be for points in spaces, where parentheses will be used. Thus $f(g(h(x)))$ will be written $fgh(x)$.

loss, or maximize the minimum loss or, most usually, minimize the expected loss given the data. Some hypothesis spaces have relatively natural choices of loss function. It is well known that for vector spaces for example, choosing an Euclidean norm with respect to some basis means that the minimum expected loss is achieved by using the expectation of the hypothesis as an estimate. If a Manhattan norm is used, the resulting estimate should be the median. Some loss functions, of William Tell type, that are small when the hypotheses are the same, grow as the difference between them grows, and shrink again for large differences, offer effectively no good estimate at all.

Unfortunately, all these possibilities assume a great deal of structure on the space of hypotheses. In the absence of anything except the preconditions for probability theory, there is only one obvious loss function, the delta function. This says that a "miss is as good as a mile". It is large and negative when the estimated hypothesis is the correct one, and zero otherwise. It is easy to see that the requirement to minimize the expected loss with this loss function leads to a best estimate given by $\arg\max_{h \in \mathcal{H}} \Pr(H = h | D = d)$. This is the MAP (maximum a posteriori) estimate. It has three great advantages. The first is that it is always applicable. The second is that it is not necessary to compute integrals over vast hypothesis spaces. The third is that it can be re-expressed in a useful way by minimizing the negative logarithm of the probability rather than maximizing the probability. The negative logarithm of the probability is known as the *energy* by analogy with statistical physics, and its use simplifies the notation involved considerably. The disadvantage of the MAP estimate is that it throws away all of the probability distribution. It is the saddle-point approximation to the estimate using the mean and, interpreted in that way, is only really accurate if the distribution is extremely peaked about the maximum. Nevertheless, because of its computational tractability the MAP estimate will be used throughout this thesis.

It is in principle possible to compute how reliable the estimate is for any of these scenarios by considering the mean value of fluctuations around the estimate. This is normally extremely difficult to do for the models considered in computer vision.

2.2. *Real Image Semantics*

As discussed at the beginning of the chapter, each real image has a semantics, unknown to the visual system, derived from the semantics of the scene. Call the space of all possible complete scene semantics $\mathcal{S}$, and the space of all possible images $\mathcal{I}$. *Complete* here means that the scene semantics is sufficiently large to make the image a function of the semantics, up to noise introduced by the microphysics of the scene (recall that the camera is a part of the scene), and to contain the statements in which we are interested. The

notion of completeness depends upon the camera: for a 'shadow' camera that takes silhouette pictures, a complete semantics would be much smaller that that required for a conventional camera for example. Given such a semantics, we can imagine a probability distribution $\Pr(I = i|S = s)$ whose form is in principle known from the physics of the scene and the camera.

Of course just because the image is a function of the scene does not mean that the reverse is true. Nevertheless, given a prior probability distribution on possible complete semantics, we can generate the joint distribution of images and semantics, $\Pr(I = i \,\&\, S = s) = \Pr(I = i|S = s)\Pr(S = s)$. From here we can generate the posterior probability of the semantics, $\Pr(S = s|I = i) = \Pr(I = i \,\&\, S = s)/\Pr(I = i)$, where $\Pr(I = i)$ can be found by marginalization.

In general, it will be true that $\Pr(I = i|S = s) = \Pr(I = i|T = t)$, where $t \subset s$. Many aspects of the scene do not affect the resulting image. These depend on the camera. The example of the shadow camera has already been given. Properties of the scene in different spectral ranges may also be irrelevant because the camera does not record its interaction with most spectral ranges. Of most importance in practice however, are the physical limitations imposed by the transparency or otherwise of parts of the scene to the physical carrier producing the image. The most obvious example is the gathering of light by a 'normal' camera. The camera gathers light passing through a small cross-section in certain directions only due to the opacity of its body and transparency of its lens. This means that most of the world is out of view. In the part of the world that is in view, opaque objects block the light from objects further from the camera but projecting to the same point in the image domain. Information about any of these parts of the scene is irrelevant to the formation of the image. For visible light, such occlusion is very important. It is less important for X-ray images for example, and assumes an entirely different form for ultrasound images. If the frequency of the carrier is low enough, there may be diffusion effects that render the useful idea of image formation as projection an approximation. The study of image formation is thus rather complicated. We will deal with one or more monochromatic cameras operating in the visible spectrum. We assume that at each instant they behave as orthogonal or projective maps from $\mathbb{R}^3$ to $\mathbb{R}^2$, creating a single image $\mathcal{D} \to \mathbb{R}$, and that the intensity of the light at each point of $\mathcal{D}$ is faithfully preserved in the value of the map.

Note that just because the image does not depend on $S \setminus T$ does not mean that we can find out nothing about this part of the scene semantics, because we have the prior on scene semantics to consider too. In fact the posterior distribution on $R = S \setminus T$ is given by $\Pr(R = r|I = i) = \sum_t \Pr(R = r|T =$

$t)\Pr(T = t|I = i)$. Unless R is independent of T, we can acquire knowledge about R. We can frequently fill in missing objects in images for example.

The discussion so far must all be taken formally of course. We do not know what the space of complete semantics is, and still less do we know how to define a measure on it. This does not prevent it being a useful way of thinking, but it does prevent us doing any computations with it. To do that, we must simplify. By looking at the enormous space $\mathcal{S} \times \mathcal{I}$, we are looking at the 'microstates' of the system, the most detailed picture that we can have. To render this useful, we need a thermodynamics, dealing with much smaller spaces of 'macrostates'. Just as in physical thermodynamics, the choice of macrostate is up to the modeller. Some properties can be held constant while others are allowed to "flap in the breeze". What aids us, is that in the performance of a particular task, we are generally not interested in the totality of information about the scene, even if such could be acquired. There will in general be one frame of discernment, Q, indexed by a (possibly continuous) value q in a space $\mathcal{Q}$, so that each complete image semantics contains one (and necessarily only one) proposition $Q(q)$ from the frame of discernment. The frame of discernment then defines a macrostate by inducing an equivalence relation on $\mathcal{S}$. Two semantics are equivalent if they both contain $Q(q)$. This produces a projection from the space $\mathcal{S} \times \mathcal{I}$ to the space $\mathcal{Q} \times \mathcal{I}$, and hence we can push forward the probability distribution from the larger space. This is not an approximation if our loss function depends only on the true and estimated values of q. In that case, in making the MAP estimate, all other degrees of freedom will integrate out, leaving the pushed forward probability distribution as the one to maximize over.

We make a similar looking assumption about the dependence on the data, but in this case it represents a real approximation. We typically assume that

$$\Pr(Q(q)|I = i) = \Pr(Q(q)|F(q, I) = F(q, i)). \tag{I.1}$$

In other words, the probability that we make the statement $Q(q)$ about image i depends only on the value of a specific functional from a family of functionals on the image space, the member of the family being determined by q. (We may have to deal with continuous spaces and densities but we stay with the discrete notation for simplicity. Nothing is changed.) It is not clear that dependence of the data used on the hypothesis always makes sense, but in our case it does: it will amount to a drastic independence assumption. A simple example is the following. We write the space of images as a product, $\mathcal{I} = \mathcal{I}_1 \times \mathcal{I}_2$. For the moment we assume that this factorization does not

depend on the value of Q. Now, in line with the previous equation, we write

$$\text{(I.2)} \qquad \begin{aligned} \Pr(Q(q)|I = i) &= \Pr(Q(q)|I_1 = i_1 \,\&\, I_2 = i_2) \\ &= \Pr(Q(q)|I_1 = i_1) \end{aligned}$$

This equation is equivalent to the assumption

$$\text{(I.3)} \qquad \Pr(I_2 = i_2|Q(q) \,\&\, I_1 = i_1) = \Pr(I_2 = i_2|I_1 = i_1)$$

If the factorization depends on the value of Q then these independence relations must hold for all q. We discuss this further in relation to our specific model in the next chapter.

3. TESTING

It is not necessary that the semantics of real images takes the form described above. If photographs are places in an art gallery for example, a whole new type of semantics can come into play that concerns the image directly, with no real relation to the scene that generated it. Comments about the relation between the colours in the image, their density, their placement, the intensities, may often be made without reference to the scene. This becomes true per force for extremely abstract images.

Nevertheless, in most applications considered, statements about the scene are the ones in which we are interested. If the visual system outputs statements that do not lie in this semantics, what are we to make of them? Let us call such systems *visual processes*. They may make mathematically correct statements about the image. For example, the partitioning of the image domain into subsets that are homogeneous in some (perhaps well-defined) way, creates a statement about the image function. This may be an interesting mathematical fact, but unless a correspondence has been established between such partitions and statements about the scene, nothing is being said that can be tested. A tautology is being stated instead.

Some areas of computer vision, for example stereo and motion computations, do not suffer from this apparent problem. The statements that they make are about the scene (stereo for example talks about minimum distances from the camera to the objects in the scene along the lines of projection, with some provisos), and can in principle be measured and tested. Of course, it is still necessary to define an evaluation function, and it is not always clear exactly what this should be in the absence of a defined task, as we discussed earlier. Similarly, visual systems that output statements of the form "In the scene that generated this image there was a human being in the volume back projected from this region in the image, and his pose was one among such-and-such family of poses" (specifying positions of arms and legs etc.) are

testable: they will be agreed upon by the vast majority of people. In contrast, statements such as "In the scene that generated this image there were such-and-such number of objects visible" are not testable. It is very unlikely that people would agree upon such a statement simply because 'object' is not well-defined.

Having made these points, it must be stressed that they are not arguments against the development of visual processes. Indeed, it is essential to develop many and varied types and versions of visual processes. It is equally essential to try and create visual systems within which they can operate and produce testable results.

4. POSTSCRIPT

In the remainder of the thesis, we concentrate on a problem in object recognition. The next chapter describes this problem and justifies the approach we will take to trying to solve it. We then move on to describe the novel visual processes to which the attempt to construct this visual system leads us.

CHAPTER II

OBJECT RECOGNITION

Object recognition is a central problem in computer vision. We define an object recognition task we would like to solve and hence a visual system that we would like to build. This will serve to guide the work and as its ultimate goal. We specify the structure of the statements we would like to make, and hence the structure of probability distributions for those statements. We then argue through a combination of examples and a review of previous work that the problem of image ambiguity forces us to compute the MAP estimate for the full distribution rather than using a greedy approach involving marginalization to break the estimation into stages. We argue that this means the use of models at several levels of specificity and abstraction simultaneously, in contrast to a commonly held position that sees a visual system as a sequence of processing stages.

FIRST we must specify the visual system that we would like to build. This system will make statements about single real images of the form "The region in the image domain projected to by the volume occupied by one of the human beings in the scene is R, and this region divides into the following parts corresponding to the projections of the arms, legs, torso and head". Any configuration of a human being in the scene generates a statement of this form, and thus a solution to the problem is testable.We will assume that the scenes that generated the images input to the visual system do all contain

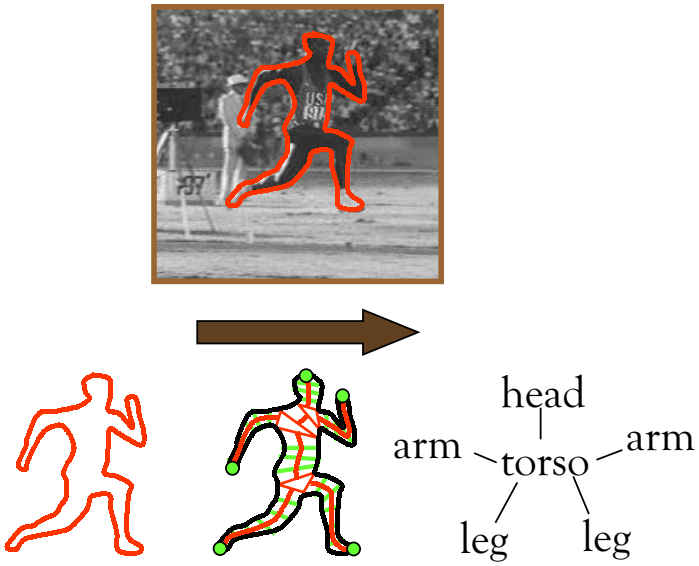


FIGURE 2. The hierarchy of models.

at least one human being at least partly visible in the image.[1] This problem may seem rather specific or artificial in form. The reasons for this specificity will be developed in this chapter. It does not mean that the visual processes with which we will primarily be concerned are that specific; indeed they are rather general. It is nevertheless useful to have this example in mind for the sake of concreteness, and as a guide to approaches and models, even if in the end more general examples can be given using the same apparatus.

We choose this problem because, first of all, it is useful. Automated information retrieval for example would benefit from the ability to perform this or similar tasks. Surveillance and miliary applications also abound. Second, it is important. Object recognition is a central problem in computer vision for good reason. Much of what we say about images concerns the

[1]We are not dealing with the question of whether or not a human being is present. We do not expect our model to answer this question, and in the abstract the probability distribution that we write down will not refer to this possibility. In practice however, the model that we construct will be applicable to images that do not contain human beings. We do not expect the answers found for such images to make sense, but given an answer on a particular image, it may be possible to perform further tests to see if the answer found makes sense as a human being in respects beyond those considered in our model. For example, we can look for eyes in the region corresponding to the head. If the image contains a human being, we might expect them to be present. If the image does not, it would be very unlikely that something like an eye would be present.

objects in them. Human beings are not trivial objects to recognize in images, especially if colour is not used. Describing regions in general and their shape in particular becomes important, and it leads us to investigate how to describe shape and how to use those descriptions in models and algorithms. The knowledge gained will be of use in the construction of systems to recognize other objects.

That shape is extremely important for object recognition may seem obvious, but psychological work confirms and illuminates this importance. According to Rosch, Mervis, Gray, Johnson, and Boyes-Braem [RMG+76], in human behaviour there is a significant difference between object classification into *basic level* categories and into more refined groupings. Basic level categories are best defined by example. Examples are 'dog' as opposed to 'mammal', 'Retriever', or 'Fido', and 'chair' as opposed to 'furniture', 'Ottoman', or 'Shaker Rocker'. According to [RMG+76], basic level categories are "the most inclusive categories whose members: (a) possess significant numbers of attributes in common, (b) have motor programs which are similar to one another, (c) have similar shapes, and (d) can be identified from averaged shapes of members of the class". They are also "the most inclusive categories for which a concrete image of the category as a whole can be formed", are "the first categorizations made during the perception of the environment", are "the earliest categories sorted and earliest named by children", and are "the most codable, most coded, and most necessary in language". For our purposes (c) and (d), and the fact that an image can be formed of the category as a whole are the most important properties. These suggest that shape is the critical determining factor in membership of a basic level category, and as such justify the use of a shape-based model for such types of recognition. This is to be contrasted with more inclusive categories, whose members may have widely varying shapes, "mammal" for example, and which seem to be created by cognitive abstraction; and more concrete categories, whose members need attributes beyond shape to distinguish them, for example "Labrador retriever" or "Ian's face".

There is also support for the use of regions (or equivalently *boundaries* or closed curves) to represent objects in image domains. Psychological work has emphasized the importance of closure in perception since the Gestalt movement. Work in illusory contours has also shown the importance of the Gestalt concept of closure to the perceptual organization involved in these phenomena [Kan71, Kan79]. Illusory contours demonstrate the concept of *completion*, which is a simple example of the image function in one part of the image domain being correlated with, and hence informing us about, structure in another part of the image domain, as discussed in a slightly different context in the next section. More recent work by Kovács and Julesz, and Elder and

Zucker has demonstrated that closure is a very important determinant of contour saliency [KJ93, EZ93, EZ94].

1. Probabilistic Model

In chapter I we laid out the bare bones of a probabilistic model for the inferences we wish to make about the scene from the image. We will now specialize this model further to the case at hand. This will enable us to discuss the model of the visual system we use and to contrast it with previous work.

Let us call the space of regions together with their named (in the sense of 'arm', 'leg', etc.) part structures, $\mathcal{Q}$. Then the set of statements "The volume occupied by one of the human beings in the scene projects to the structure $q \in \mathcal{Q}$ in the image" is our frame of discernment.[2] We will therefore be concerned with constructing probability distributions of the form $\Pr(Q = q|I = i)$, and we will need to find the maximum of such a distribution over the space $\mathcal{Q}$. To do this, we need to know a lot more about the structure of $\mathcal{Q}$. It seems natural to break up this space according to the following logic. The names attached to a part structure clearly depend on that part structure, which in its turn clearly depends on the region whose parts it is supposed to describe. This suggests that the space has a double bundle structure. From the full space $\mathcal{Q}$ there is a projection to a space $\mathcal{P}$ that 'forgets' the names attached to the part structure and leaves just the decomposition of the region into parts, and the region itself. Then from $\mathcal{P}$ there is a projection that forgets the part structure, leaving us with the space of regions, which we will call $\mathcal{B}$:

$$\mathcal{Q} \xrightarrow{\pi_{\mathcal{Q}}} \mathcal{P} \xrightarrow{\pi_{\mathcal{P}}} \mathcal{B} \tag{II.1}$$

This structure is more or less as generic as it can be if the notions of part structure and region are to have a meaning separate from the meaning of q itself, which we must certainly demand that they do in order to make the kind of statements we wish to make. Given such a structure, we can of course push the probability distributions forward from $\mathcal{Q}$ to $\mathcal{P}$ and $\mathcal{B}$. Then we have trivially

$$\begin{aligned}\Pr(Q = q|I = i) = {} & \Pr(Q = q|\pi_{\mathcal{Q}}(Q) = \pi_{\mathcal{Q}}(q)\,\&\,I = i) \\ & \times \Pr(\pi_{\mathcal{Q}}(Q) = \pi_{\mathcal{Q}}(q)|\pi_{\mathcal{P}}\pi_{\mathcal{Q}}(Q) = \pi_{\mathcal{P}}\pi_{\mathcal{Q}}(q)\,\&\,I = i) \\ & \times \Pr(\pi_{\mathcal{P}}\pi_{\mathcal{Q}}(Q) = \pi_{\mathcal{P}}\pi_{\mathcal{Q}}(q)|I = i)\end{aligned} \tag{II.2}$$

This equation is the normal factorization of a probability, but its structure is very suggestive. We will discuss it further in the next section. In terms of

[2] Actually this is not a frame of discernment unless there is only one human being in the scene that generated the image visible. We will therefore assume this in order to avoid dealing with the issue of multiple objects.

energy functionals, it corresponds to the following:

$$E(q,i) = E_{\mathcal{Q}}(q,p,i) + E_{\mathcal{P}}(p,b,i) + E_{\mathcal{B}}(b,i) \tag{II.3}$$

where it is understood that $p = \pi_{\mathcal{Q}}(q)$ and $b = \pi_{\mathcal{P}}(p)$.

We will make an assumption about the dependence of the image on the scene of the form given in equation I.1. We will say that

$$-\ln \Pr(Q = q | I = i) = E(q, i_{R_q}) + \ln Z(i). \tag{II.4}$$

As discussed in chapter I, the functional E will be called the energy. If we need to refer to it, which will be seldom, Z will be known as the *partition function*. The notation i_{R_q} denotes the restriction of the image to $R_q = \pi_{\mathcal{P}}\pi_{\mathcal{Q}}(q)$. We are saying that the probability that the structure q corresponds to a human being depends only on the properties of the image in R_q, up to a normalisation factor $Z(i)$ that depends only on the image. Note that we are not saying that the probability depends only on R_q. The q entering into the functional E contains all the information about part structure and so on that we require. We are saying that the only data we need to consider is that in R_q. Equation I.3 implies that the fact that a human being projects to the region R_q does not affect the probability of the image outside this region. These assumptions are not likely to be correct. They imply that the MAP estimate depends only on i_{R_q}. This is plainly false, since the context provided by the rest of the image may be crucial, especially if the human figure is largely occluded. Indeed there seems almost no limit to the amount that the rest of the image can tell us about q. It is possible to take a picture and excise the human figure entirely, and yet we can still make a good guess that there was a human figure occupying a certain region standing in a certain pose, and perhaps even dressed in a certain way. If the surroundings are familiar enough, we may even be able to identify the person. For examples, see figure 3.

Nevertheless, we will use this approximation simply in order to move forward. It seems likely that the rest of the image can only help significantly at the level of the semantics, in which case we are faced with the task of achieving an understanding of the whole image simultaneously. In any case, it may turn out to be sufficient to consider the region alone, with the rest of the image providing confirmatory evidence, since the MAP estimate can be the same even for widely differing distributions.

Given this assumption, the task before us is to construct the functional $E(q, i_{R_q})$. To do that, we must clarify the spaces $\mathcal{Q}$, $\mathcal{P}$ and $\mathcal{B}$ that are involved (we will call them *representations*) and define the probability distributions (or equivalently their energies) on these spaces. Before going on to do that over the next few chapters, it is interesting to spend some time discussing object recognition in general. This will turn out to justify the complex and specific

FIGURE 3. We know where the missing person is.

nature of the statements we wish to make, and shed more light on the structure of equation II.2.

2. HIERARCHIES OF REPRESENTATIONS

A common picture of object recognition is as a sequence of processing stages dealing with increasingly complicated entities, the output of one stage being passed to the input of the next. The output of the final stage is the required answer. To give an example: perhaps the first stage produces an *edge map*, a thresholded gradient image. The next stage tries to link some of the points belonging to edges into a coherent structure, perhaps a closed curve. Then a stage finds the part structure of this closed curve before a final stage compares the part structure to that of a human being. Notice that this picture is analogous to the structure of $\mathcal{Q}$ and to the factorization given in equation II.2, if we envisage estimating q using an approximate *greedy* approach in which we first maximize over $\Pr(B = b|I = i)$, and then use the result b' to maximize over $\Pr(P = p|\pi_{\mathcal{P}}(P) = b' \,\&\, I = i)$ to get p', and then use p' to maximize over $\Pr(Q = q|\pi_{\mathcal{Q}}(Q) = p' \,\&\, I = i)$.

Previous work in computer vision has by and large favoured this sequential approach to image understanding. A classic and extreme expression of this

point of view is given in Marr and Nishihara [MN78]. They say that up to the stage of the 2.5D sketch, "no higher-level information is yet brought to bear: the computations proceed by utilizing only what is available in the image itself".[3] It is not clear what can be meant by "available in the image itself". Since the image is simply a function, it could presumably exist in any world whatsoever. When photographed, some of these worlds might produce images exactly like our own, but for entirely different reasons, leading to entirely different image semantics. Even a seemingly innocent operation such as edge detection is assuming that the model of an edge that is inevitably used in the process is relevant to the world in a particular way. This does not imply that there is no well-defined sense in which edges contain more information than (for example) homogeneous areas. This information might not be relevant for object boundary recognition however. Consider a world composed of black and white rectangles moving against a grey background. The strongest intensity edges are those within the body of the object. They would be extremely useful for object recognition, but not interpreted as object boundaries. In such a world, this assumption would lead to all sorts of terrible injuries.[4]

If we discard the idea of model-independent visual processing, what is wrong with the sequential picture, if anything? Certainly it cuts the amount of computation we must do, since we are only ever maximizing over a small fraction of Ω at a time. The problem is that a mistake at an early stage will prevent successful recognition. If an incorrect region is identified at the first stage, nothing can be done later on to correct this, meaning that we must find the correct solution at the first stage. It is as if the first stage must know that it is looking for a human being before it begins. A powerful demonstration of the difficulties that images can create for such systems is given in figure 5. This image is very well-known, but familiarity should not breed contempt. The lesson of this image cannot be stressed enough. The image is so hard

[3]The 2.5D sketch is essentially image segmentation information plus depth information.

[4]The terms 'top-down' and 'bottom-up' are often used in discussing visual processes. These terms are rather insidious. They encourage the idea that there are two, qualitatively distinct types of visual process: 'bottom-up' processes are 'data-driven', whereas 'top-down' processes contain *a priori* knowledge of the world. This in turn leads to the notion that we can deal first with 'bottom-up' processes, and only invoke 'top-down' processes later in the visual system. The terminology also provides support to the notion that 'bottom-up' processes are 'objective' since they (allegedly) depend on the data alone. In fact, as we have noted in the text, even the most trivial of 'bottom-up' processes, such as edge extraction, assume *a priori* knowledge of the world in the form of a model, in this case a model of what constitutes a 'good' edge and its relevance to object boundaries. Thus the difference between 'bottom-up' and 'top-down' is not qualitative, but if anything is quantitative. The terms lower and higher level therefore seem more appropriate. The text describes what this implicit ordering might consist in.

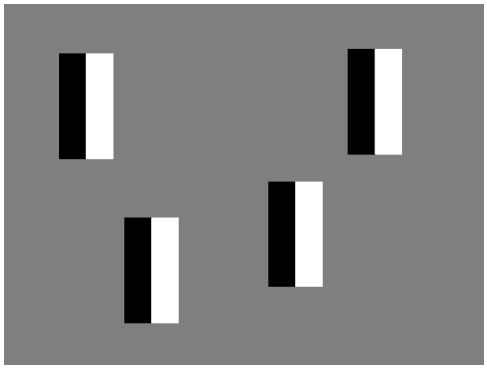


FIGURE 4. A world in which assuming that intensity edges represented object boundaries would lead to problems.

FIGURE 5. What is in the picture? A well-known but continually important image.

to interpret that human beings can normally only see the primary object in the image after being given explicit verbal clues. Nevertheless, when viewed with the appropriate knowledge, the image does contain a dalmatian dog. It is important that there is no doubt about this. One cannot see whatever one likes in the image. Being told that the image contains a motor car will not produce the same effect. Admittedly the image is an extreme example, but the difficulty it presents, which can be summarized in the phrase 'local image ambiguity', exists in almost all real world images.

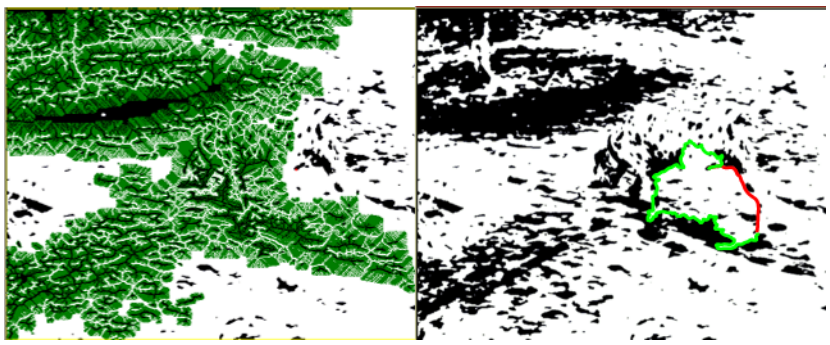


FIGURE 6. The model uses Dijkstra's algorithm to find the curve between the two end points that globally maximizes a measure of the total image gradient magnitude along the curve, while also trying to keep the curve short. The left hand figure shows a trace of the algorithm. The green areas are those points in the image that have already been explored. The right hand figure shows the path found in green (lighter), and the semantically correct path in red.

Further to illustrate the problem, figure 6 shows the result of running a typical curve detection algorithm on this image. This process finds the curve between the two end points that globally maximizes a measure of the total image gradient magnitude along the curve, while also trying to keep the curve short. Application of other techniques would lead to equally nonsensical results. It is clear that local, generic knowledge about such quantities as image gradients will never be enough to find the boundary of the dalmatian on its own. This in turn tells us that greedy schemes such as that outlined above cannot work in general.

One possible solution to this problem is that the output of the greedy sequential process is compared to the image again in some way, and the initial stages of the process altered in accordance with this comparison to produce an improved estimate for the output of the first stage. This is then passed up the various stages and the process repeated. Examples of this kind of model are the SCERPO system of Lowe [Low85], and the FORMS system of Zhu and Yuille [ZY96b]. The latter build a system to recognize object silhouettes by first splitting the object into a number of parameterized parts using their own version of the symmetry axis (to be discussed in the next chapter). The system then extracts candidate objects from a 'database' based on the part decomposition. A number of operators are then used to deform the original part decomposition so that it matches as well as possible to one of the models. If the initial part decomposition is too unlike all the models in the database,

a new object is assumed found, and added to the existing models. This is too strong a reaction, but is really the only possible one in a system where it is still possible for the low-level models to produce completely the wrong answer. Unfortunately the work lacks a theoretical framework within which it can be analyzed and developed. For example, it does not explicitly optimize a functional; it is rather a set of procedures. Because of the lack of a theoretical construct, it is unclear how to extend the system beyond the space of closed contours with which it deals and into the domain of real images, except by assuming that the contours are extracted by some ideal low-level process, exactly what we are trying to avoid.

The architecture described and implemented in the SCERPO system performs a sequence of tasks similar to FORMS: extraction of structure from the image, comparison with a set of models and choice of likely candidates, and subsequent verification and extension of the representation (in this case of edges) by projection back into the image.

In the work of Sarkar and Boyer [SB93], a Bayesian network is used for the perceptual organization of an image. The system is designed to identify simple geometric figures in gray-scale images. Like the SCERPO and FORMS systems, the work lacks a clear theoretical formulation, making it hard to analyze, but out of the three models described, all of which can be viewed as as approximating a true MAP estimate, the use of Bayesian networks seems the most principled way to incorporate feedback into an algorithm.

Another possibility is to pass a number of hypotheses to the next stage, or to be more formal, to pass a probability distribution over the more likely hypotheses. This may be a satisfactory solution in some cases, and is clearly an improvement over the previous approach, but in the case of the dalmatian we can envision the downfall of this process too. One problem is that the probability distribution can only be efficiently described by a few samples if it has low entropy, or in other words, if the number of reasonably likely hypotheses is not too large. Clearly however, the number of hypotheses in the dalmatian case is enormous, and most importantly, is not likely to narrow very much until the specific shape of a dalmatian (or at least a dog) is taken into account. This will be near the final stage of processing. In the meantime we have to process the vast number of hypotheses generated by the image, or risk making an irreversible mistake. A second problem is that it is not at all clear that the correct region need be among the most likely hypotheses at this first stage, which suggests that we must keep the whole distribution. In this case however, we are computing the exact MAP estimate for equation II.2, and we might as well admit it and attack the problem directly. The point is not that passing up multiple hypotheses is an impossible approach, but merely to suggest that things would be a lot more efficient if the information

from all the stages were available at once, to narrow the distribution from the beginning and hence direct the search.

The latter assertion can be made more formal. If we have a joint probability distribution $\Pr(X = x \,\&\, Y = y)$, then we have the following two ways to form a distribution over x. One is to marginalise, creating $P(x) = \Pr(X = x)$ by summation or integration over y. Another is to maximize over y for each x, giving a function $y^*(x)$. We can then create $\Pr(X = x \,\&\, Y = y^*(x))$ and re-normalize to give a distribution $Q(x)$. Note that maximizing $Q(x)$ over x to give x^* gives the full MAP estimate for $\Pr(X = x \,\&\, Y = y)$, $\langle x^*, y^*(x^*)\rangle$, so that we have lost nothing in terms of our estimating power. The distribution $Q(x)$ always has a lower entropy than $P(x)$. To see this, note that the entropy of $P(x)$ does not change if we re-label the y values in a different way for each x, because it is a summation. We can do this re-labelling in such a way that the labels of the $y^*(x)$ become the same. In that case, the distribution $Q(x)$ becomes just the conditional probability that $X = x$ given that Y has value y^*. But conditioning always reduces the entropy, so that $H(Q) \leq H(P)$. Thus in a model like equation II.2, which is general enough to cover almost all possibilities, the extra information supplied by the models for Q and P can always be viewed as defining a more complicated probability distribution for B. This distribution always has a lower entropy than the marginalized distribution $\Pr(B = b|I = i)$ does on its own, and it defines the region sought with more certainty. For this reason we need the higher-level models, even if all we are interested in is the boundary. We pay a price for this specificity however, which is that the entropy of the full distribution $\Pr(Q = q|I = i)$ over which we must maximize is always larger than that of both the conditional and the marginal distributions. We have gained accuracy at the expense of increased computational load. An interesting question is whether there is a characterization of when and to what extent this is worth doing.

There is a hierarchical structure to the concepts we are using that is reflected in the structure of the distribution in equation II.2. A region may be used to describe the projection of any object whatever, and we can anticipate that the probability distribution on regions alone will depend on generic properties only, by which we mean properties that could describe the projection of any object and that are not specific to human beings. The idea of a part structure on the other hand narrows the field considerably. We will define part structure more carefully later, but for now it suffices to say that by part structure we mean articulated parts such as arms and legs, rather than surface features such as eyes or hair. Such a structure does not describe well all types of object, and we might expect that the probability distribution on the combined space of part structures and regions will narrow the search to 'articulated objects' using some characteristic geometric properties of such

objects in addition to the generic region properties. In the final stage, which corresponds to the naming of the parts, we expect the probability distribution to contain information specific to human body structure.

We therefore have a progression from a *generic* model of 'Everyobject' to a *specific* model of a human being, via a model of articulated objects. At the same time, however, as the models are becoming more specific, they are also becoming more abstract. By this we mean that in some sense (and making this sense precise is not easy) the representations become 'more invariant'. Certainly a region is invariant to changes in the image produced for instance by changing illumination. A part structure will be independent of certain changes in the region, and so on. This is possible because the more generic models takes care of the dirty details, leaving the more specific models free to describe higher level structures. Both the progressions 'generic to specific' and 'concrete to abstract' are useful. The generic models should be useful not just in the particular task considered but in other similar tasks, thus avoiding redundancy. Abstraction is convenient for a couple of reasons. It leads to economy of representation, because the more abstract representations keep only the information relevant to the task at hand, and it also lends its power to the description of the relations between objects. If, as seems likely, we would eventually need some kind of reasoning engine to take into account context and encode high-level knowledge of the world, this engine will take a far simpler form in terms of abstract descriptions than it would using B-splines.

3. POSTSCRIPT

All the approaches discussed above can be seen as methods for computing approximations to the exact MAP estimate, but the arguments suggest that these approximations are likely to fail badly in general. We will therefore concentrate on computing the full MAP estimate if possible. The picture of the visual system that emerges is then not one of successive stages, but of a number of interacting representations, each of which influences all the others. There is still a notion of hierarchy, from generic to specific and simultaneously from concrete to abstract, and this is important. The generic parts of the system can be re-used, thus preventing redundancy. They also narrow the search considerably (think of matching a dog template directly into the dalmatian image). The generic parts of the system are also the concrete parts, and as such they serve to couple the image to the more abstract representations. In fact every representation serves this purpose, coupling a more concrete to a more abstract representation. The existence of the concrete representations also enables the abstract ones to exist by modelling aspects of the object that the abstraction ignores. The utility of abstract representations becomes clear as soon as we begin to consider relating

objects to one another or reasoning about them. We now proceed with the first stage of the development of the visual system we have been discussing.

CHAPTER III

GENERIC OBJECT MODELS

As a first step, we focus on the space $\mathcal{B}$ of regions, the simplest of the representation types described in the previous chapter. After defining the representation space and some possible functionals on this space, we review previous work that uses these or similar concepts. Based on criticisms of this work, we propose a new form of energy functional on boundaries in the image domain. This functional takes the form of an energy density, and solves many of the problems with previous work in the area through a combination of pleasing formal properties: it is scale-invariant, can incorporate both region and boundary information, and can be globally minimized in polynomial time for any choice of data. The application of the functional to single images is demonstrated.

THE guiding task we have set ourselves is that of finding the region of the image domain occupied by a human being in the scene, assuming that one is present, and of finding certain additional information about this region, namely its part structure and the labels of its parts. We wish our visual system to return one of the following set of statements from the image semantics: "The volume occupied by one of the human beings in the scene projects to the structure $q \in \mathcal{Q}$ in the image". The space $\mathcal{Q}$ of named part structures of regions has a double bundle structure, projecting first to a space $\mathcal{P}$ of part structures and regions, and second to a space $\mathcal{B}$ of regions. Because in our model, the region in the image is a geometric projection of a volume in the scene, such a statement enables us to say a great deal about the scene semantics independent of the viewpoint and camera: the location of the human being in the scene, their posture and so on. The last chapter argued that to do this successfully meant using different representations of the region simultaneously, to capture different types of information at different levels of specificity and abstraction, rather than following a greedy step-by-step approach. The representations play a dual role: they enable the construction of more sophisticated models connecting the semantics to the image, while at the same time encoding the statements in the semantics in which we

are interested. Thus paradoxically, even in order to obtain information about the region alone, it is necessary to introduce extra structure on that region first. The visual system then produces a statement of the above form by choosing those representations that maximize the probability over their combined space.

In this chapter, we focus on what can be done by considering the region alone, in the absence of any extra structure, or in other words we look at the distribution $\Pr(B = b|I = i)$. We pursue the extra structure in chapter V.

1. Representations and Functionals

Mathematically speaking, we are interested in functionals $\mathcal{B} \times \mathcal{I} \xrightarrow{E} \mathbb{R}$, the energy of the distribution. In order to define such a functional, we must first define exactly what we mean by the space of regions $\mathcal{B}$. We choose to represent a region R by its boundary ∂R. Although mathematically it makes little difference whether we choose to think in terms of regions or boundaries, at least in two dimensions, algorithmically it makes a big difference as we will see later in this chapter.

We think of regions and their boundaries in the image domain (single or otherwise) as two- and one-chains, since we will primarily need to integrate over them. We also have the space of zero-chains, which are collections of weighted points. The *boundary operator* ∂ takes the space of two-chains to the space of one-chains and the space of one-chains to that of zero-chains. As usual, chains in the image of the boundary operator are called *boundaries*, while those in its kernel are called *closed*. A closed chain is known as a *cycle*, and all boundaries are cycles ("the boundary of a boundary is zero": $\partial^2 = 0$). In general the reverse is not true, but if the homology is trivial, which it will be in almost all the spaces we consider, then all cycles are boundaries. In two dimensions, the boundary operator on two-chains has a well-defined inverse because there are no non-trivial closed regions. This means that regions and boundaries are interchangeable. In higher dimensions this is not usually the case.

We will restrict the form of boundaries that we consider. We will almost always force the boundaries to have a single component, so that there is no sub-chain of the boundary that is itself a boundary. In two-dimensions, this means that the regions we consider will be simply connected. (At the end of the next chapter we discuss what can be done if we relax this constraint.) In fact, we will do more than this. We will restrict ourselves to the space of equivalence classes of continuous piecewise-smooth embeddings of the oriented circle, S^1, in the image domain $\mathcal{D}$ (a generic element of the latter

space of maps being denoted γ).[1] Members of an equivalence classes are related by the continuous piecewise-smooth automorphisms of S^1. A generic member of this latter space will be denoted ϵ, while the space itself will be denoted Diff(S^1). The orientation-preserving subgroup will be denoted by Diff$^+(S^1)$. An equivalence class therefore defines the same subset of the image domain independent of the particular map and orientation used to represent it.

Although in this chapter we will write as if all boundaries are in the image domain $\mathcal{D}$ for single images, equipped with a Euclidean metric, our notation will apply to the case of a general Riemannian manifold (and beyond in fact).[2] This means that we have the freedom to change the metric on the image domain as required, and to deal with domains more exotic than subsets of the plane. For example, in the next chapter we deal with multiple images without having to rethink the formalism. It costs nothing to deal with this more general case, and in fact treating the material this way may be beneficial, since it encourages us to think geometrically and to express the invariances that we need in a clear fashion.

1.1. *Invariance Issues*

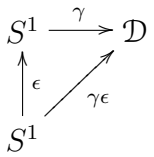
Any probability distributions we write down should be invariant under a change of representative map in the equivalence class corresponding to a boundary, because all members of a class have the same image set, and hence encode the same geometric information. Therefore, replacement of γ by $\gamma\epsilon$

[1] We choose this class of boundaries for two reasons. First, they are relatively easy to describe and visualize, but general enough to include everything we might need. In particular, they can have corners. Second, the boundaries that we obtain when we embed a graph into the domain of an image lie in this class, and the optimizations of the models to be described rely on embedded graphs.

[2] For completeness and later use we define two-dimensional Euclidean space, E^2, of which $\mathcal{D}$ is a subset. E^2 is an affine space $\langle X, T\rangle$, where X is a manifold isomorphic to $\mathbb{R}^2$ (and henceforth identified with it) on which the two-dimensional vector space T acts by translation. The space T is the *tangent space* to X, and is also isomorphic to $\mathbb{R}^2$. The translations generated by T identify the fibres of the tangent bundle to X in a canonical way, and T can be identified with each fibre. Tangent vectors can thus be viewed as elements of T and compared at will. The tangent space T will be equipped with an inner product $(.,.)$ (and hence X is equipped with a metric). We write $\|v\|$ to mean $(v,v)^{1/2}$. The inner product is used to define an antisymmetric product $(.,.)_\times$, defined as $(u,v)_\times = (u,v^\perp)$, where $(v,v^\perp)=0$, and $(v^\perp,v^\perp)=(v,v)$. Clearly the definition of $v^\perp$ requires a choice of orientation. Nothing will depend on this choice.

should have no effect:

(III.1)



It is not hard to write down functionals that do not have this behaviour, and it is worth pointing out that this is not a trivial matter. Nothing in the image can distinguish between γ and $\gamma\epsilon$, and the choice of one or the other is arbitrary. Energy functionals that do depend on this choice are meaningless.

We would also like the functionals we consider to be invariant under isometries of the image domain, assuming that the data possesses such symmetries also.[3] Thus in the Euclidean case, we would like the functional to be invariant under translations and rotations, if the data is. If $\mathcal{D} \xrightarrow{\theta} \mathcal{D}$ is an isometry of the image domain, then we require that our functionals be independent of the change from γ to $\theta\gamma$.

The functionals should also behave sensibly under global changes in the scale of the metric. We will require that scaling the image domain by a constant factor merely scale our functional by a constant factor also. This amounts to invariance to a change in the size of the image (or in the discrete case, to a change of sampling rate). It is essential if the energies are to have any real meaning, since image size may be varied without varying the semantic content. (There is another meaning to the phrase 'scale invariance' that is as important as this one. We will return to it later in the chapter.)

1.2. *Linear Functionals*

The simplest functionals on the space of chains in a manifold are the linear functionals. These are the functionals that are expressible by integration over the chain. They are linear in the sense that the integral over a linear combination of chains is the linear combination of the integrals over the constituents. Such functionals correspond to probability distributions with simple dependence characteristics: they are first-order when viewed as Markov random fields on the domain of the chain.

[3] Note that being told 'which way up' an image should be is data, and probability distributions built using this data may violate the symmetry condition. We do not make use of this possibility.

The following equations give four useful and general forms of linear functional. We have written them as functionals of γ, but they are all well-defined on boundaries.

$$E(\gamma) = \int_{S^1} \gamma^* \mathbf{A} = \int_{\partial R} \mathbf{A}$$

$$E(\gamma) = \int_{S^1} *_\gamma \gamma^* f$$

$$E(\gamma) = \ll \gamma^* \phi, \gamma^* \phi \gg = \int_{S^1} \gamma^* \phi *_\gamma \gamma^* \phi \qquad \text{(III.2)}$$

The first is constructed from a one-form $\mathbf{A}$ on $\mathcal{D}$, and need not involve a metric.[4] The second is constructed from a function f on $\mathcal{D}$, and involves a metric via the Hodge star. In the last example, ϕ can be a function *or* one-form on $\mathcal{D}$. This integral is orientation-independent and positive. The brackets $\ll \cdot, \cdot \gg$ denote the inner product on the space of forms described in appendix A. We will always assume that the metric on S^1 is the metric pulled back from $\mathcal{D}$, so that the Hodge star $*_\gamma$ on S^1 is determined by the particular representative γ chosen from the equivalence class of embeddings described above. This is to ensure the necessary invariance under the replacement of γ by another member of the same equivalence class. It is defined using the metric pulled back from $\mathcal{D}$ by γ, and has the property that $*_{\gamma\epsilon} = \epsilon^* *_\gamma \epsilon^{-1*}$, thus ensuring the required invariance. All the functionals are integrals of one-forms on S^1 of course, but the point of the different types is to show how such one-forms can be constructed from data on $\mathcal{D}$.

Before giving some examples of one-forms, it is worth pointing out the wealth of more complicated possibilities inherent in the idea of functionals on boundaries. Given the map γ, we can construct product maps $S^{1n} \stackrel{\gamma^n}{\to} \mathcal{D}^n$ and construct functionals for these maps. The linear functionals on such spaces of maps take the form for example of multiple integrals in which different points of the boundary interact with one another. This can lead to fascinating behaviour: finger-like structures that bear some resemblance to objects with articulated parts are one possibility [Gol97]. (Our approach to constructing functionals for such shapes is different superficially, but perhaps leads to the similar types of functional.) Since the human ability to identify the regions occupied by human beings in images *defines* such a functional (modulo assumptions we are making about the dependence on the rest of the image and on prior contextual knowledge, which we are not attempting

[4]We use the language of differential forms throughout because it is the most correct and concise, and because it reveals the invariances of the quantities involved most clearly. Appendix A provides a (very brief) description of differential forms and a dictionary to convert to the language of vector calculus.

to model), there is no reason to expect that these and yet more complicated examples are not of relevance to the problem. This is another argument for getting to the point where the visual processes can be taken seriously as visual systems and tested, however limited the universe of images.

Continuing with our simple linear functionals, we give some concrete examples of equations III.2 in the case of single images on a domain $\mathcal{D}$. The one-form A can be taken to be $*\mathbf{d}i$, where $*$ is the Hodge star on $\mathcal{D}$ (rather than the induced Hodge star on S^1). In the Euclidean case, using a concrete (but arbitrary) coordinate system on the circle, t, and a Euclidean coordinate system x^a on $\mathcal{D}$, the functional becomes $\int_{S^1} dt\, \epsilon_a{}^b \dot{\gamma}^a(t)(\vec{\nabla} i)_b(\gamma(t))$, where $\epsilon_a{}^b$ is the unit determinant antisymmetric tensor in two dimensions. This integral (which is orientation-dependent) will have a large magnitude when the image gradient is directed perpendicular to the boundary and of large magnitude. If we were to minimize this energy over all boundaries in the image, we would find the region with the most image gradient flowing into or out of it. Alternatively, we can use the magnitude of the image gradient squared as a function ϕ on $\mathcal{D}$. This form is orientation-independent, but does not consider the relative directions of the image gradient and the boundary. It will prove useful to have a positive orientation-independent energy that decreases with increasing image function gradient. An example that we will use later is given by taking $\phi = |\mathbf{d}i|^{-1}$.[5]

2. Previous Work

These examples are not new (although it is hard to find the directed version in the literature, perhaps because it is orientation-dependent). That a large image gradient on the boundary of a region makes the region more likely to be the projection of an object has long been accepted in computer vision. There is a great deal of previous work that relies on this fact. This work suffers from a number of problems though, as we now discuss.

2.1. *Boundary Extraction Processes*

The earliest work in the area focuses on edge maps, or atomic edge segments known as *edgels*, and attempts to link the points in the edge map or the edgels together using various criteria, particularly co-circularity and low curvature. This approach, which involves thresholding the image gradient to find the edges, and hence selecting the 'best' edges at the outset, incorporates all the problems discussed in the last chapter. For example, it is not at all clear that by choosing the 'best' edges at the first stage, we actually capture the information we need for subsequent processing. A great deal of the image is

[5]The modulus of a form, $|\phi|$, is defined is $|\phi|^2 = *(\phi * \phi)$. It is a positive function.

being thrown away.[6] In addition, a lot of this work suffers from the defect that it is described in a concrete algorithmic way only, no abstract characterization being given. This does not prevent it being tested (assuming that it meets the criteria discussed in chapter I), but it does prevent a clear analysis of the properties of the model.

Ullman [Ull76] develops a simple model of contour completion using two circular arcs, one tangent to edge elements at either end of a contour gap. A minimum curvature criterion is used to select between these curves. Parent and Zucker [PZ89] use a relaxation labeling process to compute a most probable assignment of discrete tangent and curvature values to each point in the image, with a constraining relation built on co-circularity. The process has the effect of de-noising images with respect to the contours they contain, or equivalently, of making contours more salient. The latter is also the subject of Shashua and Ullman's work [SU88]. They compute a transformation of the image intensity, creating a new image in which the intensity at a point is a measure of the saliency of that point. The saliency function at point P is defined in terms of curve segments passing through P: it is the maximum over all curve segments passing through P of a function defined on each curve segment. The value of this function increases as the length and smoothness of the curve segment increases. Like most similar early work it suffers from the lack of a clear theoretical description of what the algorithm achieves. Luckily, Alter and Basri [AB97] provide a thorough analysis of [SU88], pointing out the method's preference for a σ shape over an o for example, since the extra spur on the σ would produce higher saliency due to lower curvature. Guy and Medioni [GM96] describe another similar system. They define an extension field, a vector field whose orientation at a point describes the preferred direction of continuation of the line germ generating the field, and whose magnitude describes how much influence the given germ has on the line direction at that point. At each point in the image, a voting system, using the contributions from all the extension fields at that point, computes the preferred direction and more importantly, a measure of saliency of that point. The results are somewhat similar to the work of Shashua and Ullman. Elder and Zucker [EZ96] develop a process for finding closed contours using chains of tangent vectors, but they drastically prune the search space to render tractable the exponential problem they have set themselves.

A better-defined approach emerges with the work of Kass, Witken and Terzopoulos [KWT88] on *active contours*, more concisely and popularly known as 'snakes'. In this work, functionals are defined on a space of boundaries or open curves, and are optimized locally using gradient descent methods. (The

[6] Ironically, in the Dalmatian image the edge information is enough, because the image is binary.

evolution of the snake during this minimization leads to the epithet 'active', although it has no significance in terms of image semantics.) The functionals typically used have gradient terms as already described (usually those versions that do not depend on the relative orientation of the gradient and the curve), and a length term (which is simply the last of equations III.2 with $\phi = \mathbb{I}$, where $\mathbb{I}$ is the function that is everywhere unity.). They also have a slightly more complex term involving the curvature of the curve, designed to smooth the curve. This results from lifting the curve from the image domain to its tangent bundle using some choice of frame on the circle. This frame is usually taken to be the unit tangent vector field on the circle as defined using the pulled-back metric, although the functional must be invariant to this choice.

Later work used level set methods for the optimization problem. This type of process allows for the possibility of topology change, which seems to have led to the misapprehension that such methods are global. This is not the case. The principle reason for their use is algorithmic: they are numerically more stable. The possibility of topology change is an added bonus.

Unfortunately the global optimization problem for these functionals is NP-hard, which forces all these processes to rely on local optimization.[7] Local methods bring with them an unknown dependence on initial conditions, and an unspecified mechanism for choosing those conditions. They do not provide any information about the significance of the boundary for the image as a whole. Viewed as a MAP estimate they are particularly bizarre. Why should a local maximum of a probability distribution matter at all? Kass, Witken and Terzopoulos were well aware of this, as they were of the limitations of the types of local image data that they use. In their original paper, they quote Marr and Nishihara, whose view they describe as typical of computer vision at that time (and possibly still), as saying that up to the 2.5D sketch "no higher-level information is yet brought to bear: the computations proceed by utilizing only what is available in the image itself." But as we discussed in chapter II, nothing is present in the image alone: the image is a function and nothing else; a choice of models must be made. Not to do this explicitly is merely to fail to acknowledge your assumptions. Kass, Witken and Terzopoulos then go on to say that "This rigidly sequential approach propagates mistakes made at a low level without opportunity for correction." As the paper stresses, this places "stringent demands on the reliability of low-level mechanisms." Instead the paper proposes that low-level processes should

[7] To see that the problem is NP-hard in the discrete case if the cycle is required to be simple, consider a graph G. Equip each edge of the graph with a negative weight. Now the minimum weight simple cycle is a Hamiltonian cycle. Allowing non-simple cycles renders the problem ill-defined due to the presence of a negative cycle. Restricting the weights to be positive renders the problem tractable but the solution trivial.

provide "alternative organizations among which higher-level processes may choose," and suggests that selection of an organization from the alternatives should be "accomplished by the addition of energy terms that push the model towards the desired solution." The authors end with a strong statement of their view on the structure of the visual system: "We believe that the ability to have all levels of visual processing influence the lowest-level visual interpretations will turn out to be very important." This is essentially the view of this thesis.

Subsequent work attempted to overcome the locality constraint by using graphical methods. Unfortunately, here too the computational complexity of finding a non-trivial solution to the minimal boundary problem meant that either the work concentrated on open curves or it used various ad hoc mechanisms to ensure closure that then destroyed the optimality of the algorithms. Amini et al. used dynamic programming as part of a gradient descent procedure. Their process could be summarized as 'evolve, minimize, repeat'. Montanari used dynamic programming to find the minimum energy path between given end-points, a problem that is not NP-hard and that possesses a non-trivial solution. Geiger et al. use an initialization with a series of points, and a choice of window around those points, to delineate the space of curves considered. They then use dynamic programming and an ad hoc mechanism to close the curve. The process does not find the global optimum.

The dynamic programming approach relies on the fact that the curve is one-dimensional, so that a directed acyclic graph can be created and the topological ordering used to decide in what order to process the vertices. Dijkstra's algorithm, which is essentially the same thing except that it includes the brilliant idea of using the current vertex weight to order the vertices, can be used to solve all of these problems more efficiently. All of this work uses initialization and restricted regions of the image to limit the space of curves over which the optimization proceeds, and the algorithms find local minima or approximations to global minima over a limited set of curves. It would seem that to find a non-trivial global minimum requires open curves, but unfortunately open curves do not segment anything, so that further work is needed to group them. This frustrating choice was one of the prime motivations for the work described in this chapter. As will be discussed a little later, obtaining a non-trivial solution for boundaries in polynomial time rules out models of all the above forms, and specifies a new and essentially unique form of functional.

Mumford provides an interesting overview of the history of energy minimizing curves, and elaborates the connection to certain types of stochastic process [Mum94]. Focusing on "elastica", curves that minimize a linear combination of the length and integrated squared curvature, he shows that they can be viewed alternatively as the most probable paths taken by a particle

undergoing a particular type of random walk in the plane, and hence as diffusion processes. The advantage of the probabilistic formulation is that it lays open the full distribution of paths rather than just the most probable one.

Williams and Jacobs [WJ97] provide a similar approach to the grouping of edge maps into closed curves. À la Mumford, they model curve completion as a random walk in the tangent bundle of the image domain. The emphasis is on computing the *stochastic completion field*, a measure of the likelihood that a path will pass through a given point on its way from a given source to a given sink. The authors stress that the model they use is Markov, and this allows them to split the stochastic completion field into two contributions, one from the source and one from the sink, and compute the full field as a product of the two, each of which is computable by a linear operation. They also emphasize the important point that the full probability distribution of paths is available to them via their method, rather than just the minimum energy path, and that this information may play an important role, for example in describing the sharpness of illusory edges. As is shown in [WJ97], their model is equivalent to an energy minimization formulation using an energy that is a weighted sum of the squared curvature and the length, but the formulation as a random walk enables the authors to emphasize the neurological basis of their model. They identify a group of cells within V2 that seem to have the behavior required of the stochastic completion field.

Williams and Thornber [WT98] provide an interesting analysis of the transitions that take place in the distribution of stochastic paths as parameters in the image are varied, and in particular, the transition from differentiable contours to contours with corners. They do not give an energy formulation of their model, but such an analysis would be most interesting. The phase transitions should be predictable from the energy also and perhaps more said about their universality.

The work of Cox, Rao, and Zhong [CRZ96] is closely related to the work in this chapter. They use a graph algorithm known as the pinned ratio algorithm to find closed contours in an image. The process can be made initialization-free, and finds a global minimum under some weak constraints. It suffers from the severe drawback that the region information used must be strictly positive everywhere in the image, which rules out many quantities of interest.

2.2. *Energy Densities*

All of this work (with the possible exception of [WT98, WJ97, CRZ96, TW96]) suffers also from an unacknowledged difficulty: an uncontrolled dependence on scale. By this, we do not mean the scaling behaviour of the energy functional under a re-scaling of the metric discussed at the beginning

of this section. To illustrate the differences, we refer the reader to figure 7. This figure shows three type of scale invariance. The bottom left is metric scaling. The bottom right is invariance to an expansion in the size of the image domain: simply the addition of more data. By assumption and subsequent design, our energies are invariant under such changes.

The third type of scaling shown at the top is the one we are addressing now. If we have two boundaries in the image domain that are scaled versions of one another,[8] and if the data at the boundaries is the same, what reason would we have to pick one boundary over the other? Of course there are many reasons, but they are all task-dependent. What we do not want is a preference for one or the other to creep into our models in an uncontrolled and unnoticed way especially if it is parameter-dependent. Yet this is true of all the work discussed so far. The energy functionals used are all *extensive* quantities, a physical example being mass, that scale with the length of the curve. It is as if we were to ask "Which weighs more, lead or water?" and then to answer the question by putting some arbitrary samples of lead and water on the scales. Another problem associated with extensive energies is that of 'skipping', shown in figure 8. In order to join two points x and y in the image domain, the curve may choose a short path that the data does not support at all to a longer one that is well supported simply because of length. Note that we are not considering the case in which an explicit length term is added. This behaviour is solely due to the extensive nature of the energy.

To overcome these problems, it is necessary to use an *intensive* quantity: an energy density. This change, motivated by the above argument, turns out to bring with it a quite remarkable benefit:

> For energy densities on boundaries, the global optimization problem can be solved in *polynomial* time.

That what seems like a more complicated functional should be easier to optimize is an interesting lesson. In fact, it turns out that in many problems, converting the quantity involved to a density renders the problem polynomial. For further details concerning this interesting fact, we recommend the paper of Meggido [Meg79].

The following analysis shows that orientation-dependent energy densities on boundaries are really the only ones we should consider if we are interested in global optima, as they are the only ones that have both efficiently computable and non-trivial solutions. To see this, let us divide up problems of this nature into eight categories, depending upon whether they search for open or

[8]Note that we must restrict ourselves to a vector space for this to mean anything. It also raises the interesting question of invariance under conformal transformations, or local changes of scale. As it stands this group is far too large, but it is true that depth creates a local scale change.

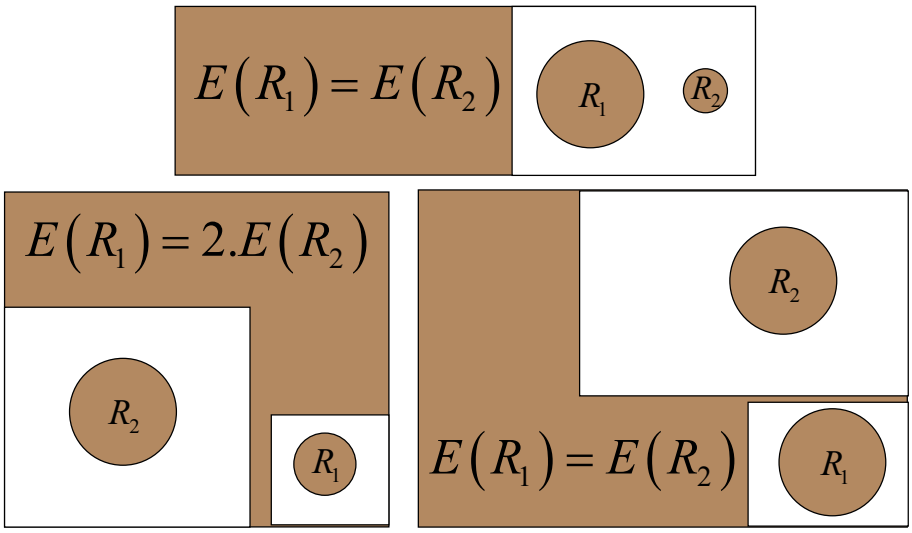


FIGURE 7. On the left, the first type of scale invariance is a global change in the metric. Under such a change, we would like energy functionals simply to scale, so that the MAP estimate is unchanged. Otherwise our results would depend on image size. In the second type, on the right, we do not change the boundary, but simply enlarge the image around it. We are assuming that our functionals depend only on the image function restricted to the region that we would like to find, so that such a change will not affect the energy. In other approaches, notably image partitioning, this invariance does not hold. While it seems likely that the probability of a region does depend on the rest of the image, as we discussed in chapter II, this dependence will be via the image semantics rather than via the details of the image function itself. This is further discussed in section 2.3. In the third type of invariance, at the top, we simply scale the boundary (restricting ourselves to Euclidean space). Unless we have a reason for preferring large or small boundaries (in which case we will explicitly incorporate this preference), our energy functionals should not be biased towards one or the other.

closed curves, whether the energy functional is a density or not, and whether it is orientation-dependent or not ('oriented' or 'un-oriented').[9] We can form

[9]We ignore the possibility of orientation-independent but non-positive energies.

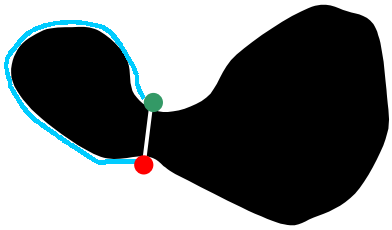


FIGURE 8. Instead of taking the long way round to join the red to the green dot, the model recommends the short white line that has no image support. This is due to the extensive nature of the energy, not a term that explicitly penalizes length.

the chart shown in figure 1, listing these possibilities and the properties of their global solutions.

We thus see that only intensive orientation-dependent energies on the space of boundaries have non-trivial global optima that are simultaneously efficiently computable. The new form of energy that we discuss here has exactly this form, but before going on to discuss such energy densities, it is interesting to consider another problem that is often categorized as similar to the type we have been discussing: image partitioning.

2.3. *Image Partitioning Processes*

Image partitioning processes, as their name suggests, partition the image domain. The competition between image partitioning approaches and boundary extraction has a long history, although the problems involved are very different, and it is not obvious that they should be lumped together. The former tends to use properties of regions such as texture and homogeneity, while the latter looks at the properties of boundaries, using information such as image gradients. One difficulty with boundary extraction processes that we did not mention in the last section is their failure to incorporate region information. Image partitioning processes on the other hand typically do incorporate region information and often, although not always, are initialization-free. They suffer however from difficulties of their own. The most important technical problem is that they usually find it hard to include important boundary properties such as smoothness and local gradients. One of the most interesting properties of the new models described in this chapter is that they allow the use of both region and boundary information while still being efficiently globally optimizable.

Extensive	Un-oriented	Open	Trivial $\downarrow 0$. Finds smallest weight edge.
		Closed	Trivial $\downarrow 0$. Finds smallest weight edge.
	Oriented	Open	Trivial $\uparrow \infty$. Wraps around negative cycle, or NP-hard if simplicity is enforced.
		Closed	Trivial $\uparrow \infty$. Wraps around negative cycle, or NP-hard if simplicity is enforced.
Intensive	Un-oriented	Open	Trivial $\downarrow 0$. Finds smallest ratio edge.
		Closed	Trivial $\downarrow 0$. Finds smallest ratio edge.
	Oriented	Open	Trivial $\downarrow 0$. Finds largest ratio edge.
		Closed	Non-trivial—finds minimum ratio weight cycle in polynomial time.

TABLE III.1. The different possibilities for energy functionals on boundaries. The notation $\downarrow 0$ means that in the continuous case the trivial solution is simply a point, while the notation $\uparrow \infty$ means that it is an infinite path or cycle. The notes in the last column refer also to the discrete case, which we discuss is section 4. We assume that we are minimizing the energy for both orientation-independent and orientation-dependent energies. The latter case is the same as maximizing an orientation-independent energy, since cycles come in pairs with opposite orientations and hence signs. The search over open curves is assumed to be over all open curves in the image domain without constraint. If the space of curves is constrained to be that between two fixed points, then the problem becomes polynomial.

The paper of Zhu and Yuille [ZY96a] uses region and boundary information within one energy optimization model to partition the image domain. The work brings active contour and region growing techniques together, which in itself is interesting. The model and the algorithm are rather complex however, and only local minima can be found. Nevertheless the model is important for the serious attempt to bring together these two types of information.

Causing quite a stir over the last couple of years has been the work of Shi and Malik [SM97]. They use the graphical concept of a *normalized cut* to partition the image into two parts. The image domain in the discrete case is converted into a weighted complete graph with the pixels as vertices. The weights typically fall off sharply with Euclidean distance. This dependence on distance means that the resulting partition of the vertices into two sets makes sense as often as not. Without it, any subset of the image domain could be

a partition. For a given partition of the vertex set into two subsets A and B, the weight of the cut is given by $\frac{E(A,B)}{E(A,A)} + \frac{E(B,A)}{E(B,B)}$, where $E(C, D)$ is the sum of the edge weights of the edges with one end in C and the other end in D. Since the graph is undirected (it is not clear what a directed version could mean), $E(A, B) = E(B, A)$. The advantage of a normalized cut as against, for example a minimum cut, is that it is normalized. One might think that this means that it is scale-invariant in a modified version of sense three from section 2.2. In fact this is not really so. If the data is ignored, the method tends to prefer partitioning the vertex set into two equinumerous subsets. (In contrast, use of the minimum cut tends to find partitions consisting of one very small and one very large region.) The process bears some resemblance to the Fisher linear discriminant, since it compares 'between-group' differences to 'within-group' differences. Unfortunately, the problem of finding the partition with the minimum normalized cut value is NP-hard. Shi and Malik come up with an approximate process by relaxing the characteristic function of one of the subsets to a real-valued function (essentially an unnormalized probability). The relaxed problem has a generalized eigenvalue solution corresponding to modes of oscillation of the graph viewed as a system of masses and springs. They then threshold the real-valued function with the largest eigenvalue to obtain a solution. Functions with smaller eigenvalues are regarded as alternative partitions.[10]

As it stands, the normalized cut process says nothing about the boundaries of the regions it finds. Indeed the notion of geometry is only weakly coded into the edge weights. Leung and Malik [LM98] extend [SM97] by incorporating weak contour continuity information into the region-based model. The problem is that by creating a model so unconcerned with geometry (it is really a generalized clustering algorithm), the incorporation of geometric information is hard and unnatural. Many subsequent papers, one of which we discuss in the next chapter, have applied the technique to slightly different problems. The most recent papers are still working on the incorporation of boundary information.

There is a mass of other work that uses fields defined on the image domain to partition it. Mumford and Shah describe a variational formulation for image approximation by a piecewise-smooth function [MS85, MS89]. One could then view the discontinuities as a partition, providing they took a sensible form. Markov random fields, which are essentially the discrete version

[10]There is another 'normalised' cut known as the minimum quotient cut. The input is a graph $\langle V, E\rangle$, a vertex-weight function w, and an edge-weight function c. The solution is the subset $C \subset V$ that minimizes $\frac{c(C)}{\min(w(C),w(V\setminus C))}$, where $c(C)$ is the sum of the weights of the edges into or out of C, and w is applied to a set of vertices by summation. This problem is NP-hard but approximable within $\log(\text{card}(V))$ [MT90].

of fields defined on the image domain, are used by Geman and Geman [GG84] primarily for image restoration. All these processes use variants of gradient descent or Monte Carlo methods to optimize locally the functionals they use. The whole field of anisotropic scale spaces, which involves evolving the image function under the action of a non-linear operator, attempts to remove noise and other clutter from images while preserving for example important edges. While not strictly speaking partitioning processes, they can be used to aid in such processes and the spirit is similar.

One problem with image partitioning processes concerns scale invariance in sense two of section 2.2. Image partitioning processes are not so invariant. Expanding the image domain can frequently change the results of such processes. This dependence is not the same thing as using image data outside the occupied region in a semantic way to influence the probability distribution over regions. It is instead a kind of action at a distance, for which there seems little justification in semantics. Although in certain extreme circumstances expanding the image domain may change the probabilities of certain objects being distinct, image partitioning processes suffer from this problem in all circumstances. This brings into question what such processes are actually saying about the image.

We argued in chapter I from another direction that it is not clear, except in a few instances, what type of statement partitioning processes are trying to make about the image, or even what they are aiming for as part of a larger visual system. The idea of partitioning a scene into objects does not seem to be a testable one, in the language of chapter I, except in very limited circumstances. The idea of using a field on the image to recognize objects is counter-intuitive anyway, since normally we think of objects as being *in* space, rather than as continuous with it, but with some added value.[11] Embeddings of volumes in space naturally push forward to the image; the same is not true of fields. Fields are more appropriate for describing surface properties, so that if the image consists of a single object in whose surface properties we are interested, fields may be a good bet. The analysis of satellite imagery often falls into this category. Even here however, discontinuities may make a picture in turns of embedded objects more suitable.

[11] In physics it is an article of faith that we can in principle analyse every object in terms of its constituent quantum fields, and hence represent the state of the world accurately by fields on space. Unfortunately, what concerns us here is not physics but image semantics, which in its turn derives from how we talk about the world. It is not clear how contingent our concepts and language are. Even if they are determined by physics, we do not know how.

3. Globally Optimal Regions and Boundaries

We have argued that the extensive energy functionals previously used for the extraction of boundaries from images suffer from a number of problems, chief among these being their lack of scale-invariance and the difficulty of finding the global minimum. We then argued that an intensive energy density would solve these problems. We now proceed to introduce such an energy density.

The new energy density functional is defined on the space of oriented boundaries in the image domain, $\mathcal{B}$, defined as above in terms of embeddings of S^1. The definition is

$$E[\partial R] = \frac{N[\partial R]}{D[\partial R]} = \frac{\int_{\partial R} \mathbf{A}}{\ll \gamma^*\phi, \gamma^*\phi \gg}. \tag{III.3}$$

where γ is any representative of ∂R. We use the notation ∂R rather than b to emphasize that the domain of integration is a boundary. As above, $\mathbf{A}$ is any one-form on $\mathcal{D}$, and we use ϕ to denote any function *or* one-form on $\mathcal{D}$ as in equation III.2. The equation defines N and D, the numerator and denominator functionals. Note that the integrand in the denominator is always positive by construction. The denominator can be generalized to be a sum of terms like that shown, each of which is built from any function or one-form. Note that because we are considering oriented boundaries, the values of the energy come in pairs, with opposite signs.

The denominator is a weighted measure of the length of the boundary, and has the effect of damping the scaling behaviour of the energy. Indeed, under certain conditions on the one-form in the numerator, the energy functional is scale-invariant in the second manner discussed above: it has no inherent bias towards large or small boundaries. In the case that the one-form is derived from a linear operator applied to the image function, the condition is that the operator have compact support. This means that the operator will scale appropriately along with the data. The condition is true for example for image function gradients.

Energy densities have an obvious but intriguing relation to extensive energies of the form $H[\partial R] = N[\partial R] - \beta D[\partial R]$, where $\beta \in \mathbb{R}$. Indeed this relation is the basis for the optimization algorithm and we discuss it further in section 4.4. The relation amounts to a choice of β for which H is positive with minimum value 0.

3.1. *Region Information*

Under certain conditions, which include the Euclidean case, much more can be done with the energy in equation III.3. We would like to be able to say something about the region that is the interior of the boundary; indeed that is where we began. For example, it would be useful to find the boundary that, in addition to having high image function gradients along its length, also contained a region of highly homogeneous image function, or that contained a particular texture. This is exactly the type of extra information that we can incorporate into our model using Stokes' theorem.[12] This says that the integral of a one-form $\mathbf{A}$ along a boundary is equal to the integral of its derivative $\mathbf{dA}$ over the region contained by the boundary: $\int_{\partial R} \mathbf{A} = \int_R \mathbf{dA}$. Thus if the measure of optimality of a region R can be expressed as the integral of a two-form $\mathbf{F}$ over the region, $E[R] = \int_R \mathbf{F}$, and if we can find a one-form $\mathbf{A}$ such that $\mathbf{dA} = \mathbf{F}$, then we can re-express the measure of region optimality $\int_R \mathbf{F}$ as an integral over the boundary $\int_{\partial R} \mathbf{A}$, and use this to measure the optimality of a region in equation III.3.

Given any *exact* two-form $\mathbf{F}$ on M, we can by definition find a one-form $\mathbf{A}$ whose differential $\mathbf{dA} = \mathbf{F}$. In addition, if the 2-cohomology group $H^2(M;\mathbb{R})$ of M is trivial (as it is, for example, for $\mathbb{R}^2$), then all closed forms $\mathbf{F}$ (a form is closed if $\mathbf{dF} = 0$) are exact. In two dimensions all two-forms are closed, meaning that we can take *any* two-form $\mathbf{F}$ and find a one-form $\mathbf{A}$ such that $\mathbf{dA} = \mathbf{F}$. What is more, the construction of this one-form is trivial. It involves an integration along the x direction with y held fixed, and an integration along the y direction with x held fixed (or a similar procedure in another coordinate system) to generate the two components of the one-form. In the discrete case, this means scanning the image just twice. In higher dimensions the cohomology may still be trivial, but now not all two forms are closed. The two forms that *are* closed are exactly those whose integrals over a region depend only on the boundary of that region. These quantities are not usually the most interesting. In particular, this restriction prevents us doing all that we might wish in the case of multiple images in the next chapter.

The above discussion means that *any* measure of region optimality expressible as an integral over the region can be re-expressed as an integral over its boundary. Thus the new functional can include region information such as homogeneity and texture as easily as it can boundary information such as image function gradients. For example, suppose the one-form $\mathbf{A}$ measures

[12] The vector calculus version of this theorem is known as Green's theorem in two dimensions, and Gauss' theorem in three dimensions. The original Stokes' theorem is also subsumed into the differential form version.

some boundary information, whereas the two-form $\mathbf{F}$ measures region information. We want a measure of the form $\int_{\partial R} \mathbf{A} + \int_R \mathbf{F}$. Since we can always find a $\mathbf{B}$ such that $\mathbf{dB} = \mathbf{F}$, we can rewrite this expression as $\int_{\partial R}(\mathbf{A} + \mathbf{B})$, which is in the form of the numerator of equation III.3.

It is obvious from this discussion that even if the energy were not a density, and consisted solely of the numerator, the above argument would still apply. What is remarkable about the new model is that even with all this generality, the functional can still be globally optimized by a single algorithm.

It is interesting to consider the region and boundary information in the following way. Boundary information, expressed as a one-form, necessarily has many similarities to a gradient, to a change in some quantity. This fits well with the idea of the boundary as the interface between the region and the rest of the image domain. The boundary information describes the differences between the region and its surroundings. In contrast, the interior of the region has no contact with the rest of the image domain. The types of properties that can be expressed here are essentially functional (in two dimensions there is a bijection between functions and two-forms given by the Hodge star), and as such describe 'absolute' properties of the region rather than properties relative to the environment.

3.2. *Gauge Invariance*

Equation III.3 has an interesting invariance. If we add any closed one-form to $\mathbf{A}$, then when we transform to a region integral by taking the exterior derivative, this term will disappear, meaning that the energy functional is unchanged. This type of invariance to local transformations is known as *gauge* invariance.

Taking the simplest example: suppose that the one-form $\mathbf{A} = *\mathbf{d}i$, as in the example in section 1.2. Suppose we add Δi to the image function. Then the numerator of equation III.3 is changed to $\int_{\partial R}(*\mathbf{d}i + *\mathbf{d}\Delta i)$. Now if $\mathbf{d} * \mathbf{d}\Delta i = 0$, the new numerator will equal the old. But $\mathbf{d} * \mathbf{d}\Delta i = 0$ is simply Laplace's equation for Δi. The conclusion is that if we add any harmonic function to the image function, the energies of the boundaries will be unchanged.[13] The reverse is not necessarily true. If we add a closed one-form $\mathbf{B}$ to $*\mathbf{d}i$, we can interpret this as adding a function Δi to i if $\mathbf{d} * \mathbf{B} = 0$. If this is true, $*\mathbf{B}$ is closed, and hence exact, and so there exists Δi such that $*\mathbf{B} = \mathbf{d}\Delta i$. If this is the case, then because $\mathbf{dB} = 0$ ($\mathbf{B}$ is closed), we have that $\mathbf{d} * \mathbf{d}\Delta i = 0$: Δi is harmonic.

[13] We assume the denominator independent of i. We could for example take $\phi = \mathbb{I}$, so that the denominator measures the length of the boundary.

All linear functions are harmonic of course, and so adding a linear function to the image function will not change our results. This can be interpreted as a limited form of illumination invariance if i is the image function. Something similar is discussed in an early paper of Blake [Bla85], which in turn derives from the Retinex model of Land and McCann [LM71, Lan83].

3.3. *Examples of one- and two-forms*

To clarify these abstract ideas, we give some examples of one-forms and functions that can be built from the image function and used in equation III.3.

First we consider the case where we generate a two-form by convolving a filter with the image function i. This includes a large number of familiar cases. Indeed the action of any linear operator on the image function can be thus described. The convolution creates a two-form $\mathbf{F} = i \diamond \mathbf{G}$, where $\diamond$ denotes convolution, and $\mathbf{G}$ is a two-form (the filter). It is easy to see that, if $\mathbf{dB} = \mathbf{F}$, and $\mathbf{dC} = \mathbf{G}$, then $\mathbf{B} = i \diamond \mathbf{C}$. This means that to convert the integral $\int_R \mathbf{F}$ to a boundary integral $\int_{\partial R} \mathbf{B}$, we need only find the one-form $\mathbf{C}$ for which $\mathbf{dC} = \mathbf{G}$. We do not need to filter each image and then integrate it; we can just filter the image with an integrated filter. This simplifies the transition from region to boundary in the most important cases.

We now give a list of specific possibilities for the terms in equation III.3.

- $\mathbf{F} = *i$**:** In this case the model is looking for globally maximum image function regions. It will find bright spots such as specular reflections, as well as large regions of high image function.
- $\mathbf{F} = \mathbf{d} * \mathbf{d}i$**,** $\mathbf{A} = *\mathbf{d}i$**:** This possibility has already been discussed. Viewed as a region integral, this function finds the region with the largest absolute value of the integrated Laplacian of the image function. Such regions correspond to 'lumps' or 'dips' in the image function, since regions with undulations in the image function will make both positive and negative contributions to the region integral, reducing its absolute value. Viewed as a boundary integral, this term measures how much image function gradient flow there is into or out of the region enclosed.
- $\mathbf{F} = *|\mathbf{d} * \mathbf{d}i|$**:** The previous function does not deal well with the case of contrast-reversing boundaries, which introduce both positive and negative contributions to the region integral. We can deal with the case of general boundaries (including contrast-reversing) using the absolute value of the Laplacian, at the expense of not having an analytical expression for the boundary integral. This region function is a better way to deal with contrast-reversing boundaries than the normal process of taking the magnitude of the gradient, since it

preserves the notion that image function change should be normal to the boundary.

$\mathbf{F} = *e^{-|\mathbf{d}i|}$: This is a positive monotonically-decreasing function of the magnitude of the image function gradient. The integral of such a term over a region will be large if the gradient has small magnitude everywhere. It will thus seek out the region with globally most homogeneous image function.

$\mathbf{F} = i * \mathbf{T}$.: A two-form filter $\mathbf{T}$ (or linear combination of filters) that responds strongly to a particular class of textures can be used to segment globally optimal regions of that texture.

An obvious choice for the denominator in equation III.3 is the single function, $\phi \equiv \mathbb{I}$. This means that the denominator measures the Euclidean length of the boundary. A linear combination, for example, of the second and last choices in the list above in the numerator, and this choice for ϕ in the denominator means that the model is searching for a region with high image function gradient flow in or out, and that is filled with a particular texture.

3.4. *Regularization*

If the numerator has a term that grows as the area, then the scale invariance is spoiled. The denominator continues to restrain the scaling behaviour of course, but it also acquires a regularizing function as a consequence of the failed invariance. For a fixed area, the shape with the largest area to length ratio is of course the circle. Similarly, since the energy functional prefers shorter boundaries if the image data is the same, it will tend to behave like a soap bubble and contract around the supporting data thus smoothing the boundary. This is confirmed by the relation to extensive energies mentioned earlier. In those energies the denominator appears as an explicit length term.

4. ALGORITHMICS

Having defined the energy functionals with which we are concerned, we now need to describe how we will solve the MAP estimation problem associated with the corresponding probability distributions, or in other words, how we will minimize the energy. In this section we present two algorithms that can be used to solve the minimization problem. Neither algorithm requires initialization, and both find the global optimum for their problem instances in polynomial time. The first applies in a restricted set of instances, but has the advantage of being extremely parallelizable. The second is completely general, applying to any energy functional of the form of equation III.3,

and on serial machines is much faster than the first algorithm.[14] Neither of these algorithms is new to this thesis as will be clear from the citations, but their application to the solution of the continuous problem described here, and hence their use in computer vision, is new. We describe them here for completeness.

In section 4.1, we set out a discrete problem with obvious similarities to the continuous problem we are considering. Then, in sections 4.2 and 4.3, we describe the two algorithms that solve this problem and compare them. In section 4.5 we describe how we can reduce the continuous problem to the discrete one, and hence solve the continuous problem.

4.1. *Problem*

Given a graph $G = \langle V, E \rangle$ (we denote $\mathrm{card}(V)$ by n, and $\mathrm{card}(E)$ by m), and two maps $E \xrightarrow{\lambda} \mathbb{Z}$ and $E \xrightarrow{\tau} \mathbb{Z}^+$, we define the *ratio weight* $W(C)$ of a set of edges $C \subset E$ as

$$W(C) = \frac{\sum_{e \in C} \lambda(e)}{\sum_{e \in C} \tau(e)} \tag{III.4}$$

Note that the weights are integral: this represents almost no restriction in practice. Note also that the co-domain of τ is the positive integers. This corresponds to the form of the integrand in $D[\partial R]$ in equation III.3, which is positive by construction. When τ is constant, we call W the *mean weight*.

Let $\mathcal{C}$ be the set of cycles in G. The problem is then to find $W^* = \min_{C \in \mathcal{C}} W(C)$ and $C^* = \arg\min_{C \in \mathcal{C}} W(C)$. Call this problem A. Note that the corresponding extensive problem, problem B, where we wish to find the minimum *total weight* cycle with weight given by the numerator of equation III.4, is not clearly defined if the graph contains negative cycles. Remarkably, attempts to alleviate this problem by restricting attention to simple cycles renders the problem NP-hard, since solution of it would allow the solution of Hamiltonian Cycle as described in footnote 7. The case of minimum ratio weights is different. Here, no special effort is needed to restrict attention to simple cycles. No path will wrap multiple times around a negative cycle, since this will not change the ratio weight. In addition, cycles that cross themselves will only occur if both simply connected pieces have exactly the same ratio weight. Not only does this not occur very often, but if it does the cycle can simply be split into two parts once it is found.

[14] There is also a linear programming approach to the problem, described by Dantzig, Blatner and Rao [DBR66], but no bounds can be given on the time to solution in this case, and we do not describe it here.

4.2. *Minimum Mean Weight Cycle Algorithm*

The first algorithm that we describe does not solve the general instance of problem A. The restriction is that the denominator weights τ should all be equal, or in other words that we are solving the *minimum mean weight cycle* problem rather than the more general *minimum ratio weight cycle* problem. Up to an irrelevant factor, the denominator is simply counting the edges in the set $\mathcal{C}$. This discrete problem does not have a continuous counterpart, dependent as it is on the discretization, but it is useful because the corresponding algorithm is extremely parallelizable. Each vertex only needs to read from, and never write to, its neighbours at each stage of the algorithm. The algorithm is due to Richard Karp, and we refer the reader to the original paper [Kar78] for proofs.

We begin with a weighted directed graph G, as described in section 4.1. We have a numerator function λ, and a trivial denominator function $\tau \equiv \mathbb{I}$. We wish to find the minimum mean weight cycle, where the mean weight of an edge progression is defined by equation III.4.

First, define the function F_k taking each vertex $v \in V$ to the weight of the minimum total weight path consisting of $k \geq 0$ edges to v from an arbitrary start vertex s, and define it to be ∞ if no path exists of k edges. Then it can be shown that the weight W^* of the minimum mean weight cycle is given by

$$W^* = \min_{v \in V} \max_{k \in [0..(n-1)]} \left\{ \frac{F_n(v) - F_k(v)}{n-k} \right\}. \tag{III.5}$$

Intuition about this equation is hard to come by. The original paper contains the proof but it is not particularly illuminating. $F_k(v)$ can be computed using the recurrence

$$\begin{aligned} F_k(v) &= \min_{(u,v) \in E} F_{k-1}(u) + \lambda(u,v)) \\ F_0(s) &= 0 \\ F_0(v) &= \infty \ , \ \forall v \neq s. \end{aligned} \tag{III.6}$$

The computation of F for all $k \in [0..(n-1)]$ can be performed in time $O(nm)$. The minimum weight paths can be computed simultaneously. Using a further $O(n^2)$ time we can compute W^* from $F_k(v)$, leading to an overall computation time of $O(nm)$. The cycle itself can be extracted by selecting the minimizing v and k in equation III.5, and finding a cycle of length $n-k$ in the minimum weight path from s to v.

The algorithm is extremely parallelizable, since $F_k(v)$ only depends on $F_{k-1}(u)$ for u in the neighbourhood of v. Thus a network arranged in levels of constant k could compute the values of F_k in parallel from the values of F_{k-1} in $O(1)$ time. To fill the whole table containing $F_k(v)$ for all k and

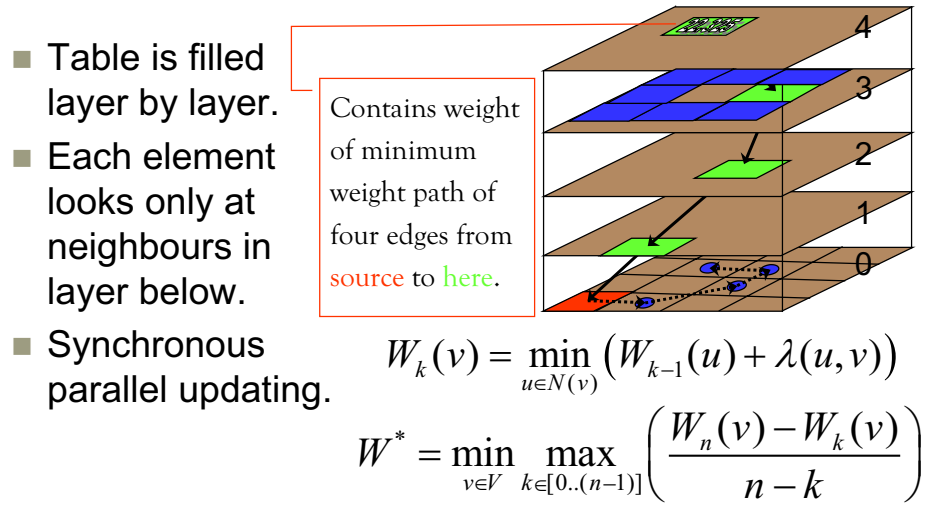


FIGURE 9. The table used to compute the minimum mean weight cycle. This table is easily filled in parallel using one processor for each vertex. The processing is trivial, and it seems likely that purpose-built hardware could perform boundary optimization using this algorithm at very high speed.

v would thus take $O(n)$ time. The minimization would take the same time, and is also easily implementable as part of the network. It is easy to envisage columns of this sort being arranged behind a detector and producing at the 'top' of the table the extracted region. This would be very fast. In addition, manipulation of the network connectivity would allow attention effects to be included. Figure 9 shows the situation graphically.

4.3. *Minimum Ratio Weight Cycle Algorithm*

The minimum mean weight cycle algorithm described above has the drawback that it cannot deal with edge weights τ other than the trivial case in which $\tau \equiv \mathbb{I}$. This has the consequence that it cannot deal with arbitrary functions and one-forms g_α in $D[\partial R]$ in equation III.3. Indeed, it cannot deal even with the case of a single function $g = \mathbb{I}$, since this corresponds to $\tau(e) =$ Euclidean length of e. Instead it uses a discrete measure: the number of edges in the discretized version of the boundary. This lack of generality and dependence on the discretization, coupled with the unfortunate properties of an edge count as a measure of distance, illustrated in figure 10, render the algorithm unsatisfactory for many purposes. The minimum ratio weight cycle algorithm described in this subsection works for arbitrary $E \xrightarrow{\tau} \mathbb{Z}^+$,

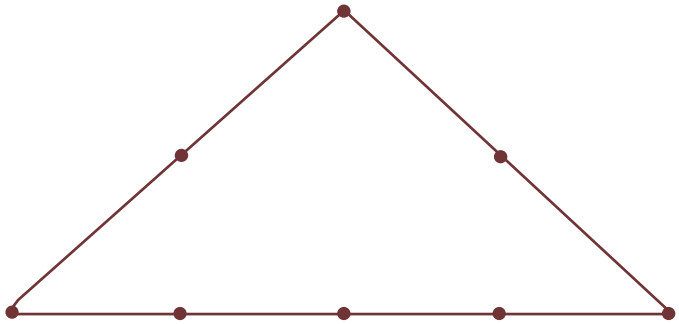


FIGURE 10. The horizontal path and the triangular path have the same number of edges.

and hence deals with arbitrary functions and one-forms g_α in $D[\partial R]$. It also requires considerably less memory resources than the minimum mean weight cycle algorithm. The main consequence of this parsimony with memory is increased speed. Instead of the execution times of a few minutes found for Karp's algorithm, we find execution times on single images of a few seconds (typically less than five seconds on a 256×256 image) for the minimum ratio weight cycle algorithm.The algorithm has the added advantage that degenerate minima of the energy III.3 can be identified simultaneously.

The algorithm was first described by Lawler [Law66], and since then has been much generalized by Nimrod Meggido [Meg79]. It relies on the following, interesting observation. Define a new, parameterized edge weight $E \xrightarrow{w_t} \mathbb{Q} : e \in E \rightsquigarrow w_t(e) = \lambda(e) - t\tau(e)$, where $t \in \mathbb{Q}$. We use the same symbol w for the weight of a set of edges (defined by summation). Then the solution, t^*, of $w_t(C_t^*) = 0$, where C_t^* is the solution to problem B with weights w_t, is equal to the minimum ratio weight W^* in problem A, and the minimizing cycle $C_{t^*}^*$ of problem B is equal to C^* for problem A. The simple proof of this fact is given in appendix B.

The problem is now reduced to finding an efficient search strategy for t^*. Although there are a number of approaches, including binary search [Law66], and a more sophisticated approach using parametric edge weights [Meg79], we find in practice that the fastest process is simply linear search. To see how this works, note that if $t > t^*$, the graph G using the weights w_t will have a negative cycle. We start with a known upper bound t^0 on t^*. Such a t^0 is easy to find. We can just choose the edge with the maximum ratio of its two weights. We then apply a negative cycle detection algorithm with edge weights w_{t^0}. If we do not find a negative cycle, and the algorithm terminates, there will be a zero weight cycle, and we are done: $t^0 = t^* = W^*$, and the

zero weight cycle $C^*_{t^0}$ is a solution C^* to problem A (as are any other zero weight cycles—hence we can detect degenerate minima). If a negative cycle C is detected, t^0 is too large. Since $w_{t^0}(C) < 0$, we have that $t^0 > W(C) \geq t^*$. We therefore replace t^0 by $t^1 = W(C)$. The search continues in this fashion until t^* is found. When t^* is found, we have that $W^* = t^*$ and that $C^* = C^*_{t^*}$. Because the weights λ and τ are integral, a pseudo-polynomial bound can be placed on the search time. Ignoring the precision factors, this bound is $O(mn)$, coming from the $O(mn)$ time bound on the negative cycle detection algorithm. In practice, since we can terminate the negative cycle algorithm as soon as the first negative cycle is found, it never executes to completion until very near the completion of the whole algorithm, rendering the time much shorter than the bound suggests. Further details of the time bound and the implementation are given in appendix B.

4.4. *A Related Extensive Problem*

The algorithm illustrates in the discrete domain the fact that the solution to the optimization problem using equation III.3 is the same as the solution to a related extensive optimization problem $\hat{E}[\partial R] = N[\partial R] - \beta D[\partial R]$, with β chosen in a specific way. The parameter β is *not* a constant, but rather is a function of the problem instance. In fact, it is a very particular function: the value of β is the minimum of equation III.3, making it a functional of the whole image. It is easy to see (as is proved for the discrete case in appendix B), that the minimizing boundary for $\hat{E}$ is the same as the minimizing boundary for equation III.3 if β is chosen in this way. This is the choice of β that gives the minimum energy state of $\hat{E}$ zero energy, and therefore renders it a positive functional on the space $\mathcal{B}$. For each image, β is effectively chosen in this way and then the minimizing cycle of $\hat{E}$ is found.

4.5. *Application*

In order to solve the continuous problem of finding the minimum of equation III.3, we must show how it can be approximated by the discrete problem described in section 4.1.

To do this, we embed a graph G in the image domain $\mathcal{D}$. This has the effect of inducing an injection from the spaces of vertices and edges in G to the zero- and one-chains in $\mathcal{D}$, and in particular the space of cycles in G, $\mathcal{C}$ is injected into the space of one-boundaries in M, $\mathcal{B}$. This means that we can define two weights, $\lambda(e)$ and $\tau(e)$, for each edge e as the integrals of the numerator and denominator in equation III.3 along the image of e under the injection. The discrete energy, equation III.4, will be equal to the continuous energy for those boundaries that are the image of some cycle in

- Embedded graph: vertices are pixels.
- Edges are between 8 nearest neighbours.
- Graph is directed.
- Opposing edges have opposite sign. of λ
- Cycle in graph is discretization of boundary in image plane.

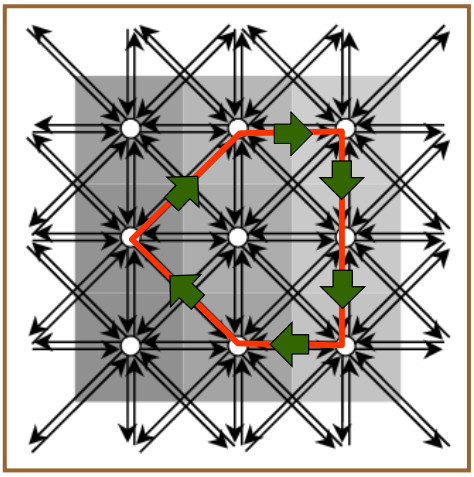


FIGURE 11. A cycle in the embedded graph becomes a continuous piecewise smooth embedding of a circle.

G. We thus have an instance of problem A, whose solution is equal to the solution of the continuous problem over a smaller discrete set of boundaries. The solution to the discrete problem can then be injected back into $\mathcal{D}$ to give a one-boundary in the manifold. This will be a continuous piecewise smooth embedding of S^1 in $\mathcal{D}$ as advertised. Since the injection from the set of cycles in G is not onto, this solution may not be exact. However, by making the graph dense enough in $\mathcal{D}$, the error can be reduced to an arbitrarily small value. Of course, this increases the computational complexity of the discrete problem, but fortunately, in image applications, the pixel scale gives an upper bound on the accuracy we can expect, and hence on the complexity of the problem.

5. DEMONSTRATIONS

Figures 14 and 15 show the results of running the algorithms on some real images. Figure14 uses the minimum ratio weight algorithm, while figure 15 uses the minimum mean weight algorithm. In these demonstrations, the graph that we embed in the image plane is a rectangular lattice with vertices the pixels and a simple bi-directional eight-neighbour structure. This is shown in figure 12, in part (b). We chose the two-form to be integrated over the region to be $\mathbf{F} = \mathbf{d} * \mathbf{d}i + *\beta$, where β is a constant function. The first term is the Laplacian, and gives a boundary term $\mathbf{A} = *\mathbf{d}i$ as discussed above. The second term measures the area, and its purpose is to eliminate very small regions. In figure 14, the denominator was simply the Euclidean length, and

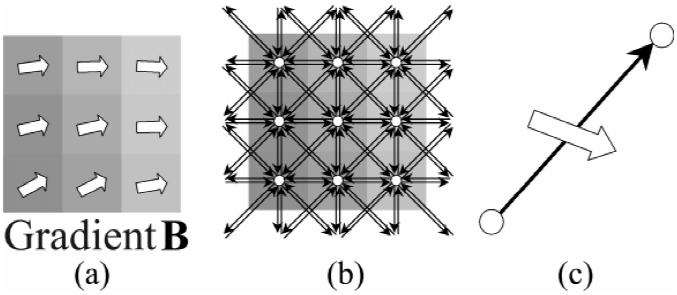


FIGURE 12. (a) For each pixel we compute the gradient vector. (b) The graph has a node for each pixel and eight outgoing edges for each node (except at the boundary.) (c) The edge weight is calculated by taking a cross product of the gradient vector and the edge vector.

$\beta = 10/3$. The energy in equation III.3 for figure 14 then becomes

$$E(\partial R) = \frac{\int_{\partial R} * \mathbf{d}i + \int_R * \beta}{\int_{S^1} *_\gamma \mathbb{I}} \tag{III.7}$$

For figure 15, the denominator was simply an edge count, as required by the algorithm, and $\beta = 0$.

At each vertex/pixel, the discrete version $\mathbf{B}$ of the gradient one-form $\mathbf{d}i$ was computed by taking a wavelet coefficient on a small scale of the order of a pixel:

$$\mathbf{B} = i \diamond \psi_s$$
$$\psi_s(x) = s^{-1}\psi(s^{-1}x),$$

where $\diamond$ denotes convolution, and $\psi = \mathbf{d}G$ is the derivative of a Gaussian. For an edge going from vertex u to vertex v, corresponding to pixels a and b respectively, the integral of the numerator of equation III.7 along the edge was approximated by taking the signed magnitude of the cross product of the vector $b - a$ with the average of the gradients at a and at b, $\frac{1}{2}(\mathbf{B}(a) + \mathbf{B}(b))$. The cross product produces the same effect as the Hodge star in this case, by effectively rotating the gradient by $\pi/2$ and then taking the dot product. The edge weight for the denominator was taken to be its geometric length, corresponding to the choice of $g = \mathbb{I}$. This is illustrated in figure 12.

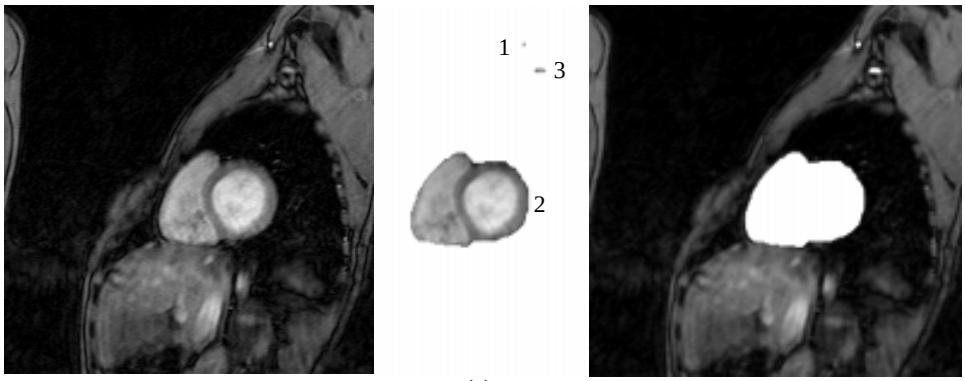
For an image with n pixels, the number of vertices in the graph is simply n, and the number of edges is $O(n)$ also. The time complexity of the minimum ratio weight algorithm $O(n^2)$. For a 256×256 image, this is 10^9. The space required by the algorithm is $O(E)$, which in this case is 10^5. On a dual processor Pentium III 500 MHz machine with 1 GB of memory, the

FIGURE 13. (a) A synthetic contrast-reversing boundary. (b) The result of applying the region two-form $\mathbf{d} * \mathbf{d}i$. (c) The result of applying the region two-form $*|\mathbf{d} * \mathbf{d}i|$. The region found is shown in grey.

algorithm executes in a few seconds on such images. The minimum mean weight algorithm has the same time complexity, but its space requirements are larger: it requires $O(n^2)$ space, as well as time. This is a lot of memory, and it results in slower performance. The algorithm takes some minutes rather than a few seconds on a 256×256 image. In each case, the algorithm was run several times on each image, and after each iteration those vertices through which the previous solution had passed were removed from the graph. In this way a series of regions of increasing energy was extracted. This can be viewed as a series of hypotheses about regions in the image of gradually decreasing probability. The numbers in the figures indicate the order in which the regions were found.

Two main differences between the two sets of demonstrations should be noted. The first is that with $\beta = 0$, more low energy small regions are found. In figure 14, the large regions are found second, first and first respectively, whereas in figure 15 they are found sixth, third and third. The second difference is also expected but more subtle. As was illustrated in figure 10, the use of an edge count as a distance measure fails to penalise 'triangles'. This results in the boundaries of the regions being jagged. This is clear for example for region number 1 in the first image of both figures, which is a diamond in figure 15, but considerably smoother and rounder in figure 14. The same effect can be noted on the boundaries of the large regions in all three images. In addition, this demonstrates the regularizing effect of the denominator in equation III.3.

In order to illustrate that the process can also deal with the case of contrast-reversing boundaries, figure 13 shows the effect of changing from the integral of the Laplacian to the integral of its absolute value $*|\mathbf{d} * \mathbf{d}i|$.



(a)

(b)

(c)

FIGURE 14. The results of running the minimum ratio weight cycle algorithm on some real images, with $\beta = 10/3$. After a boundary was found, the pixels in that boundary were removed and the algorithm run again. The numbers indicate the order in which the regions were found. (a) A 256×256 pixel image. (b) A 200×134 pixel image. (c) A 124×166 pixel image.

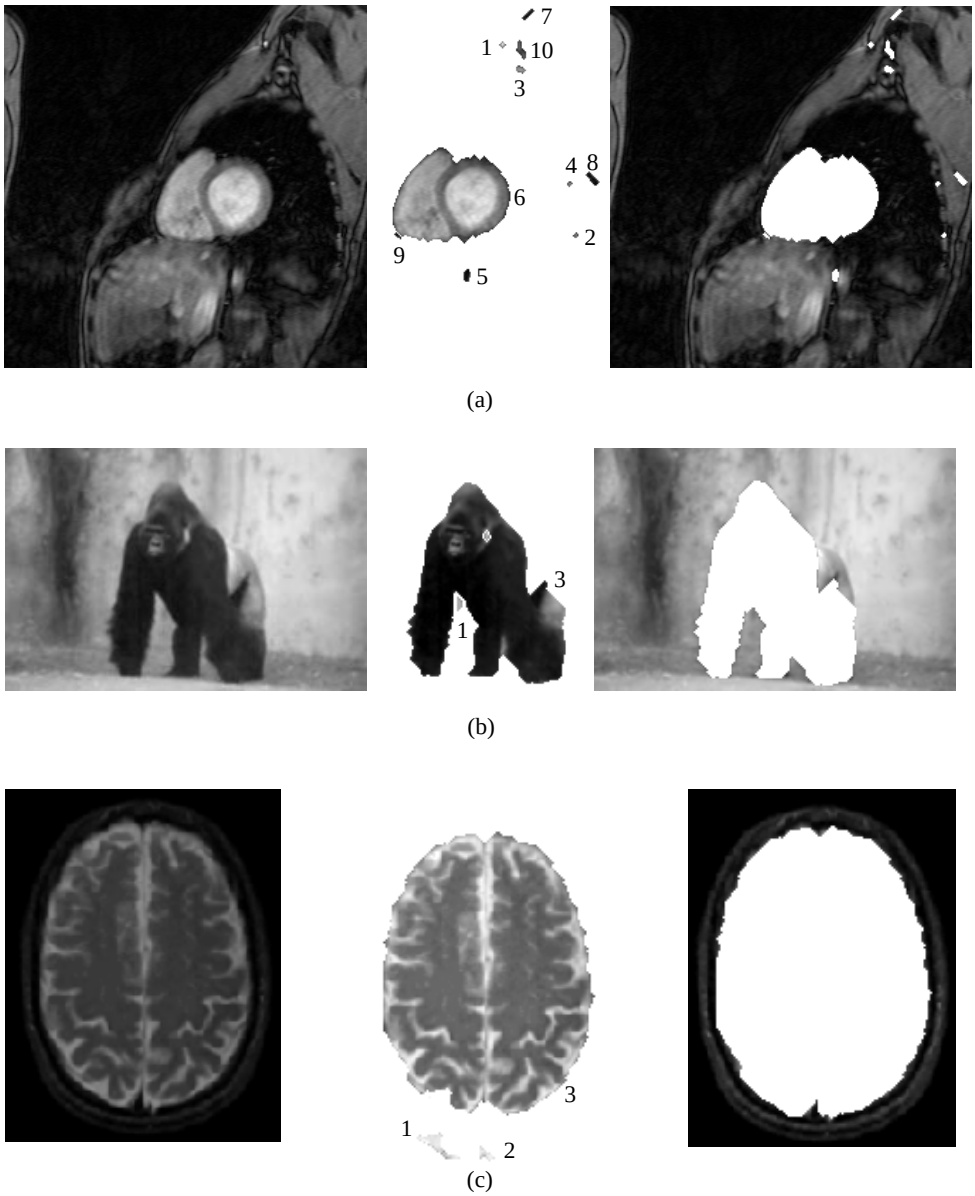


FIGURE 15. Using the same images as in the last example, the demonstrations shown here use the minimum mean weight cycle algorithm with the area term set to zero. Two differences can be noted. The most obvious is that lack of an area term, as expected, results in the finding of smaller areas earlier in the iteration process. The second is that the use of an edge count as a measure of length results in jagged boundaries, since 'triangles' are not penalised.

6. Postscript

The new form of energy functional is not restricted to two dimensions. In the next chapter, we turn to the use of equation III.3 with multiple images, resulting in the identification of boundaries and regions of boundaries in several images simultaneously.

CHAPTER IV

MULTIPLE IMAGES

The techniques of the previous chapter are applied to the extraction of boundaries from several images simultaneously, together with their correspondence. The most important examples of multiple images in vision are stereo pairs and motion sequences, the result being 'attentional' stereo or motion computation. Two types of information are available: that obtained from the images separately, and that obtained by comparing different images. Both can be incorporated into the energy density. No dense flow computation is used or needed, although the results obtained may aid in its computation.

Throughout the discussion so far, we have tried to be as general as possible about what we mean by an image. Where we have restricted the discussion at all it has been to the case of single images. In addition to being the most important class of images, this also enabled us to take advantage of the special relation between regions and boundaries in two dimensions. In some cases however we have more information available than is provided by a single image. Examples include the cases of stereo pairs and motion sequences, for which the input to the visual system consists of several single images. The various image semantics for these new situations contain all the statements from the single image semantics we have considered so far, but they also contain other statements only available to us because of the extra images. These statements are (and can only be) the result of relations *between* the images. There is thus an object recognition problem entirely analogous to the single image case except that we have the extra images both to aid us in the recognition task and to enable us to make more complex statements about the objects that we find. We will apply the techniques of the past chapter to the case of multiple images in this chapter. Our approach to stereo and motion will be different from that most commonly used in the literature, and so we will spend some time outlining the standard approach.

1. STANDARD APPROACH

Stereo and motion differ conceptually, but computationally they have often been treated as equivalent. In both cases the input to the visual system is two or more single images, $i = \{i_a : a \in N\}$. The differences between the two situations concerns the cameras and the times at which the images are generated. For stereo, the cameras are positioned and oriented differently, but the images are generated at the same instant. For motion, we deal with a time-varying image generated by a single camera. In principle this is a function $\mathcal{D} \times \mathbb{R} \to \mathbb{R}$, but we will use a discrete time so that the image is in fact $\mathcal{D} \times \mathbb{Z} \to \mathbb{R}$.[1] The time variation can arise from changes in the position and orientation of the camera, or changes in the remainder of the scene or both.

Since we are dealing with real images, given such an input, the visual system should output some statement from the scene semantics. It is clear that among the statements in the scene semantics for both stereo and motion images is a set that has the following form: "The same point in the scene generated the point x in the first image domain, y in the second image domain, ... ". The idea is that in two pictures of the same scene, whether taken at different times or from different positions, the same points will by and large appear, assuming that the differences of time or space between the viewpoints of the individual images are small, this being a standard assumption. (The notion of 'small' can be quantified in terms of the scene, for example the typical size of depth discontinuities, or the typical velocities of objects, but we do not do that here, treating it instead as a working assumption.) We can thus establish a correspondence between these points in the different image domains. It is necessary to use a relation rather than a function to describe these correspondences because some points occur in only a subset of the image domains. To formalize this for N images: for a subset $A \subset N$, let $\mathcal{D}^A = \times_{a \in A} \mathcal{D}_a$, where $\mathcal{D}_a$ is the domain for image i_a. This will be a subset of $\mathbb{R}^{2\,\mathrm{card}(A)}$. Then the statement output by the visual system is a subset $R \subset \cup_{A \in 2^N} \mathcal{D}^A$ such that each point in each image domain appears in at most one element of this subset. The subset R is required to be continuous except at certain sets of discontinuity points of co-dimension one. Thus a point in one of the image domains may match no others, or a point in another image domain, or two points in two other image domains, and so on. In the case of stereo, or of motion with two images only, we have a subset of

[1] We assume that successive time samples are separated by the same increment of time. This avoids invariance issues that would otherwise complicate matters considerably. Note that we cannot do this in the case of spatial samples, since these will not be uniformly spaced along the boundary.

$\mathcal{D}_1 \cup \mathcal{D}_2 \cup \mathcal{D}_1 \times \mathcal{D}_2$ such that each point appears in only one element, and in this case we can view the relation as a partial map from one image domain to the other. This map can be viewed as a vector field on the first image domain since we are in Euclidean space.[2] In the case of motion, this field is known as the *optical flow*, while in the case of stereo, it is known as the *disparity*. (In the latter case, further constraints to be discussed will in fact force one component of the vector field to zero, rendering it a function.) Given the relation R as above, and certain further information about the cameras, it is then possible to make further statements about the scene; for example, in the stereo case, "The distance from the average focal plane of the cameras to the nearest object in the scene that generated point x in one image and y in the other is d". The function so defined is known as the *depth*.

In the case of stereo, further assumptions are made about the relative positions and orientations of the cameras and the objects in the image. The most important is the *epipolar constraint*. Consider two cameras. Corresponding to each camera is a projection from $\mathbb{R}^3$ to $\mathbb{R}^2$. Call these projections π_1 and π_2, and consider the map from $2^{\mathbb{R}^3}$ to itself given by $\pi_2^{\leftarrow}\pi_{1\rightarrow}$. This map may have fixed points, or in other words there are subsets of $\mathbb{R}^3$ invariant under projection to one image and back projection from the other. In stereo, it is assumed that the projections are such that these subsets foliate the volume of three-space appearing in the images into planes (which may be parallel or converge to the camera depending on the nature of the projection). These planes project into parallel lines in each image, and it is assumed that the correspondence between these lines (but not the points within them) is known. This defines a subset of $\mathbb{R}^4$ within which R must lie. It amounts to giving a map $\mathbb{R} \rightarrowtail \mathbb{R}^2$ that dictates the correspondence between the stacked lines in each image. Figure 16 shows the situation. This map can then be used to reduce the dimension of the space $\mathbb{R}^4$ in which the subset R would naturally sit, by inducing a map $\mathbb{R}^3 \rightarrowtail \mathbb{R}^4$. The situation now takes the form of a number of two-dimensional problems 'stacked' on top of one another, thus reducing the disparity from a vector field to a function.

A second constraint known as the *ordering* constraint arises from assumptions about the scene excepting the cameras. It says that the correspondence within each epipolar line should preserve the ordering of the points in the line, and should be consistent from epipolar line to epipolar line. Viewing R as a partial function from one image domain to the other, this constraint says that the derivative of this function along an epipolar line should take one sign only. Again, figure 16 illustrates the situation.

---

[2]The image domains are related by 'base' isomorphisms that result from the fixation of the cameras. To avoid having to consider these uninteresting isomorphisms directly, we will assume that the domains are identical.

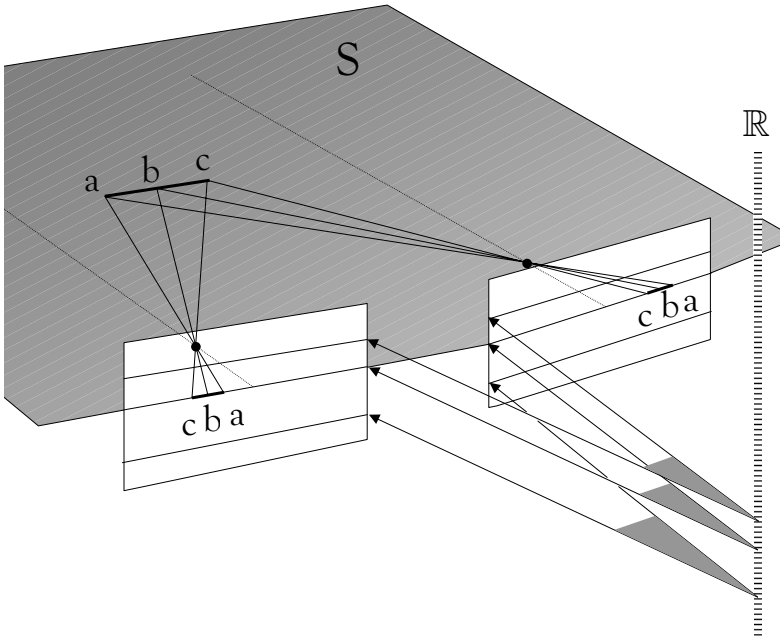


FIGURE 16. Epipolar constraint: the plane S is preserved by projection to one image and back projection from the other. It is a fixed point of the set map created by this process. It defines a line in each image domain that must correspond. Similar sets foliate $\mathbb{R}^3$ and produce corresponding lines. This can be represented by a map from $\mathbb{R} \to \mathbb{R}^2$, as shown in the lower right. Ordering constraint: the three points, a, b and c in the scene must project to the epipolar lines in an ordering preserving way as shown in the figure. Three points in a line directly away from the cameras, if visible from both sides, would violate this constraint. In this way, the ordering constraint is a working assumption about the scene.

The situation for motion is not as constrained. In general, even if there is an epipolar constraint (meaning non-empty invariant subsets) it is not known, and the dimensional reduction that takes place in the case of stereo does not happen. We therefore have to deal with the full $2N$-dimensional space.

1.1. *Previous Work*

Previous work in both stereo and motion has concentrated on exactly the type of statement from the scene semantics that we illustrated above. Almost all the emphasis has been placed on finding a *dense* correspondence in which

R covers the union of the image domains. This is partly because the above statements are often not viewed as ends in themselves, but as a preliminary to image partitioning. The idea then is that unmatched parts of the image (or equivalently, discontinuities in the computed disparity or optical flow) correspond to discontinuities in the depth or the motion of the scene, and hence hopefully to object boundaries [Aya91, BZ87, Fau93]. It does indeed seem likely that if accurate estimates could be made of the positions and directions of velocity discontinuities in particular, then the partitioning task would be made very much easier.

The probability of a particular correspondence is typically assessed using measures of difference between local 'features', examples being the image function or its gradient, or the results of applying a filter to the image function. Often an energy functional (and hence probability distribution for the correspondence given the data) is defined on the space of optical flows or disparities, which is then locally or globally optimized.

Different methods for finding the disparity or optical flow in this way face similar problems. Some regions in each image may be effectively uniform, rendering correspondence ambiguous, and the feature values of points in each image that come from the same source point in the scene may differ due to noise, as some of the early work in optical flow discovered [AB85, FJ90, Hee90, LK81]. Both these problems point to a need for control of the correspondence in a way that does not depend on the data, or in other words a prior probability. Typically such probabilities are chosen to advantage smoother correspondences. Such terms are used in optical flow computations [Ana89, BA91, HS81, NE86], and in stereo [Aya91, BB81, Gri81, Jul71, KO90, MP79, MN85, PMF85, RC98]. Although such prior probabilities work in the sense that they will smooth out noise and propagate the correspondence into regions where the data is indecisive, they also tend to smooth out discontinuities, and hence match regions of the image that should be left unmatched. Since these are often the parts of the correspondence in which we are most interested, this is a pity. Nevertheless, most of this work does not explicitly model discontinuities.

Later work in stereo [Bel96, BM92, CHMR92, GLY95, IB94, JM92, MMP87, Yui89] does model discontinuities, but the difficulties of solving the optimization problems involved means that the solutions can be found along epipolar lines only. This reduces the dimensionality of the problem, and allows the application of dynamic programming techniques. Unfortunately, it does not allow the imposition of constraints connecting different epipolar lines. For example, we might suspect that the disparity would be smoothly varying between epipolar lines. The recent maximum-flow methods [IG98, RC98] allow a fully two-dimensional treatment of the optimization

problem, and the finding of the global minimum, but the smoothing must be convex. Discontinuities are modelled by modulating the smoothing term according to the presence of high image function gradients or corners [IG98]. Another recent approach [BVZ98] finds local optima under large 'moves', an improvement over the standard gradient descent approach. It seems to be necessary to choose between finding the global optimum with convex smoothing, or using non-convex smoothing to find only local optima.

In motion, a more recent approach to the problem, known as "layers" [AS95, DP91, DLR77, HAP94, JB93, WA94, WA96], which can also be applied to stereo, approaches the problem by severely limiting the dimensionality of the space of optical flows. The method does this by modelling the optical flow as a partition of the image domain into regions, with the optical flow in each region lying in a low-dimensional subspace of the space of vector fields. Typically (although not always: [Wei97]) this subspace is an affine or projective subspace. The advantage of the method is that discontinuities are modelled explicitly. To find the optical flow, the EM algorithm is used iteratively to find the regions given the motions and the motions given the regions. The algorithm does not necessarily find the global optimum however, and in addition, it requires the choice of an initial partition and the corresponding models. The same type of ambiguity occurs here as we discussed in the context of image partitioning methods in chapters I and III. The difficulty is not just that it is not known how to make this choice, but that the choice itself is not well-defined. Specific tasks may remove this ambiguity but it is hard to see how it is possible in a task-independent way.

Using a different approach entirely, Shi and Malik [SM98] apply the normalized cut model described in chapter III to the problem of motion sequences by using it to segment the sequence directly into spatio-temporal volumes. The problems are the same as with the use of normalized cut in single images: notions such as continuity and smoothness are hard to incorporate into the model. In addition, no correspondence is set up between the images, since all that is found is a set of vertices.

In another different approach, Cipolla and Blake [CB97] use active contour methods for motion sequences. They initialize a contour in one image and then track it over a few images, using the information to compute average differential invariants of the motion inside the contour. The approach has three disadvantages. The first is that structurally it is a greedy method, and hence does not compute a MAP estimate as such (unless very unrealistic independence assumptions are made). The second is that the algorithm used is a form of gradient descent, which therefore does not compute the global optima even for the separate subproblems of the more limited greedy problem. The

third, as pointed out by Shi and Malik [SM98], is that the algorithm requires initialization through a choice of contour.

2. Stereo and Motion without Dense Correspondences

Although there are tasks for which the kind of dense correspondence generated by the above work is appropriate,[3] it is often the case that a dense correspondence tells us a great deal too much about aspects of the image that do not interest us, and rather too little about aspects that do. For example, in collision avoidance, which applies to many navigation tasks as well as to avoiding an aggressor, it is not really necessary to know the motion of or distance to every visible point in the scene, especially if this does not tell us where the approaching object is or how fast it is moving. It is far more useful to be able to detect the approaching object's location, distance and motion without having to bother with the remainder of the scene.

With this in mind, we will use the form of energy functional of the last chapter to find boundaries in each of a number of images simultaneously with their correspondences, thereby combining object recognition with correspondence computation. This might be called 'object-based' or 'attentional' stereo or motion computation. No dense correspondence is used or computed. The method can be thought of as the extension of the human recognition task that we have been using as our guide to the case of multiple images. Just as in the last chapter we discussed the first stage of our attack on that task in the case of single images, that relating to probability distributions on regions alone, so in this chapter we describe the analogous first stage in the case of multiple images. The extra structure that we are introducing, which in this case is boundary correspondences, serves a dual purpose. First, it encodes extra information about the boundary, either concerning its position or its motion in the scene. Second, it enables us to narrow the probability distribution on regions by constructing energy functionals that use this extra information. We will meet a similar situation in the next chapter, where extra structure (in that case for parts) is introduced and used in a similar way.

We envisage several uses for the models discussed in this chapter. We could go on and try to extend the object recognition aspect of the work, by incorporating part structure and so on in a way analogous to the single image case. This would be the natural choice if we were bearing in mind the collision avoidance example. Alternatively, we could use the information we find to aid other visual processes. In particular, the boundaries and their correspondences can be used as a constraining input to a normal optical

[3] Examples are using a stereo algorithm to create a virtual scene, or to map topography from an aeroplane. Any process that wishes to partition an image by using stereo or motion sequences also clearly needs a dense correspondence.

flow algorithm, both to fix the optical flow on the extracted boundaries, and to weaken any smoothing terms on the boundaries, thus encouraging discontinuities to occur there. As a special case, stereo algorithms could be applied to the interior of the boundary only, thereby amplifying the meaning of 'attentional stereo'. The information could also be used as part of an initialization for the "layers" method, or to estimate differential invariants of the motion as in [CB97]. Just as in chapter III, the method we will describe requires no initialization, thereby answering the objections of Shi and Malik to boundary-based methods.

Although some previous work uses indicators of object boundaries to influence the correspondence computation [BT99, HA89, IG98], all of these indicators are local features. None of the work uses indicators of depth or motion discontinuities with large extension in the image domains, even though object boundaries do have such extension. If available however, this information is very valuable, as it introduces correlations between distant parts of the image domains and, if correspondences are known, between the image domains, that should be preserved by the correspondence computation. In addition, in previous work these indicators are computed for each image independently, a process that does not guarantee that the features correspond, and which throws away a great deal of useful information about the correspondence. In contrast, in the work described here, the boundaries are not found in each image domain independently and used separately, nor are they found and then matched. Instead the boundaries and the correspondence between them are found simultaneously.

If, instead of viewing the boundaries as features to be discarded once a dense correspondence is found, one is interested in both a dense correspondence and the boundaries, then the non-local nature of region and boundary information and the difficulty of modelling discontinuities suggests that it is better first to find corresponding regions and boundaries and then find a dense correspondence rather than the reverse procedure that is normally followed.

3. Representation of Boundaries in Multiple Images

We will represent boundaries in multiple images $i = \{i_a : a \in N\}$ simultaneously as a single boundary in the product space of the domains of the images, or in the stereo case the subset of this space defined by the epipolar constraint. We refer to both these spaces as $\mathcal{D}_S$. We are thus looking at the

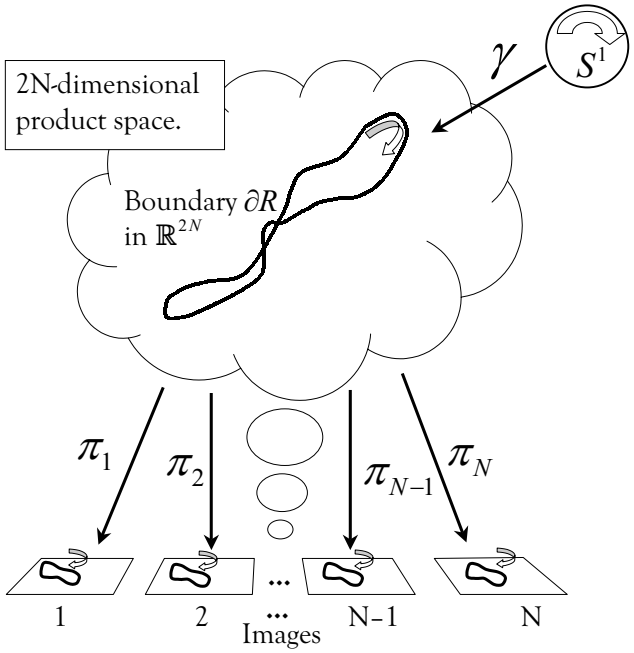


FIGURE 17. The boundary in the $2N$-dimensional space $\mathcal{D}_S$ resulting from the product of the image domains is shown euphemistically, along with the projections to the individual image domains.

following space of maps:

(IV.1)
$$\begin{array}{ccccc} \mathcal{D} & \xleftarrow{\pi_1} & \mathcal{D}_S & \xrightarrow{\pi_2} & \mathcal{D} \\ & {\scriptstyle\gamma_1}\nwarrow & \uparrow{\scriptstyle\Gamma} & \nearrow{\scriptstyle\gamma_2} & \\ & & S^1 & & \end{array} .$$

in the case of two images, with more projections added when there are more than two. We are interested in boundaries ∂R rather than maps, so that we have the familiar invariance under replacement of Γ by $\Gamma\epsilon$. Figure 17 shows this structure graphically for the motion case. Figure 18 shows the analogous structure in the stereo case. Both show a boundary in $\mathcal{D}_S$ (one of the boundaries searched by the model), together with its projections into the two image planes. Figure 18 also shows a pair of epipolar lines that must correspond to each other.

In the stereo case, we must enforce the ordering constraint as well as the epipolar constraint. In order to do this, we impose a condition on the tangent vectors of the boundaries we consider. If we denote the Euclidean coordinates

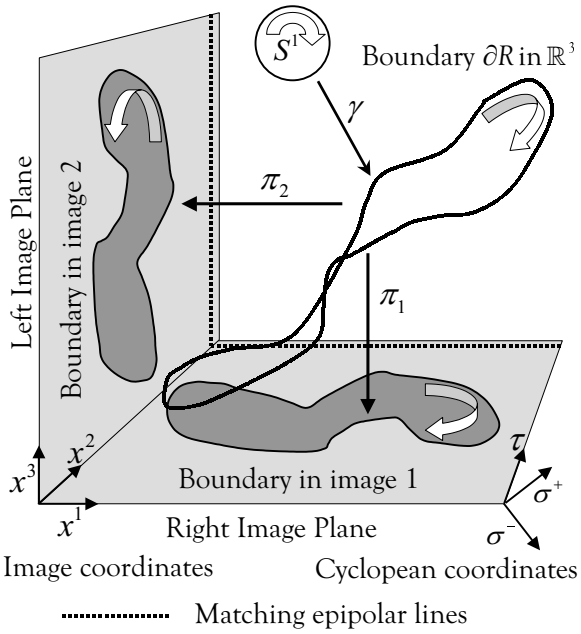


FIGURE 18. A boundary in the three-dimensional space $\mathcal{D}_S \subset \mathbb{R}^3$ resulting when the epipolar constraint is enforced is shown, along with its projections to the two individual image domains. A pair of matching epipolar lines is also shown, as are the coordinate systems used in the text.

in $\mathbb{R}^3$ by x^a, where $a \in \{1, 2, 3\}$, and x^1 and x^3 run along the epipolar lines and x^2 perpendicular to them, we may define a *Cyclopean* coordinate system by $\sigma^- = (x^1 - x^3)$ (the disparity), $\sigma^+ = (x^1 + x^3)$, and $\tau = x^2$ (see figure 18). In the basis derived from the Cyclopean coordinate system, we can express any vector v as $v = v^+ \frac{\partial}{\partial \sigma^+} + v^- \frac{\partial}{\partial \sigma^-} + v^\tau \frac{\partial}{\partial \tau}$. We restrict attention to those boundaries whose tangent vectors $v = \frac{d\gamma}{dt}$ obey

$$(v^+)^2 > (v^-)^2 \tag{IV.2}$$

When we come to discretize the problem, this will mean that edges that violate this constraint will drop out of the graph that we construct.

4. ENERGY FUNCTIONALS

Having defined the representations that we will use, we now move on to the energy functionals. There is a lot of similarity between the cases of stereo and motion and we will try to keep the discussion unified.

4.1. *Numerator*

We will use the numerator of equation III.3 to measure the optimality of the projected boundaries in each image independently. To do this, we define a one-form $\mathbf{A}_a$ on the image domain $\mathcal{D}_a$ of each image. Typically these one-forms will have the same functional form since there is symmetry under interchange of the images. Each of these one-forms may consist of the linear combination of several terms that would normally be referred to as boundary information, and several terms that would normally be referred to as region information, just as discussed in chapter III. We then define a one-form $\mathbf{A}$ on $\mathcal{D}_S$ by pulling back the A_a using the π_a, and then forming a linear combination:

$$\mathbf{A} = \sum_{a \in N} \alpha_a \pi_a^* \mathbf{A}_a \tag{IV.3}$$

where $\alpha_a \in \mathbb{R}$. The α_a will typically all be equal because of symmetry under interchange of the images, and hence they will drop out as an irrelevant multiplicative constant. The integral of this one-form around ∂R will form $N[\partial R]$ in energy III.3. Note that because of the properties of the pullback, we have:

$$\int_{\partial R} \mathbf{A} = \sum_{a \in N} \alpha_a \int_{\partial R_a} \mathbf{A}_a \tag{IV.4}$$

where $\partial R_a = \pi_a \partial R$. Thus this quantity is a measure of how optimal a boundary/region is ∂R when examined in each image independently. If we are using region terms, then the fact that the A_a have the same functional form will tend to lead to the correspondence of regions with similar properties, even though no direct comparison is being made.

4.2. *Denominator*

In chapter III a trivial denominator $D[\partial R]$ was used in equation III.3. Here we exploit the freedom allowed by the model, and construct a $D[\partial R]$ that compares the boundaries in the images to each other. There are many ways to do this, some of which are discussed in this chapter and the next. Here we focus on what seems the simplest possibility for stereo and motion. We will write down a denominator term that does two things. First, it compares boundaries using the image function differences between corresponding points, favouring boundaries in $\mathcal{D}_S$ for which these differences are small. In addition, this preference is weighted in a way that favours boundaries that lie on parts of $\mathcal{D}$ where the velocity or disparity changes sharply.

We will achieve this by defining a single function g on S^1 and then defining the denominator of equation III.3 by $D[\partial R] = \ll g, g \gg$. This

function will itself be a ratio g_N/g_D. The function will compare successive images pairwise. In the stereo case this restriction makes no difference, but it is important in the motion case. It does not make sense to compare images widely separated in time. In effect, we are making a Markov-like assumption about the dependencies of the probability distribution over the space of boundaries when viewed as a time-ordered set. The method does not impose any such restriction however, and we could for example include a term that penalized large accelerations, a second order effect in time.

Our numerator and denominator functions will accordingly take the form $g_N(i) = \sum_{a=1}^{N-1} g_{a,N}(i_a, i_{a+1})$, and $g_D(i) = \sum_{a=1}^{N-1} g_{a,D}(i_a, i_{a+1})$. For every $a \in N$, the $g_{a,N}$ and $g_{a,D}$ will take the same functional form because of symmetry, so it is sufficient for us to define then for the case of two images.

The Numerator g_N. We define the following function on $\mathcal{D}_S$ for a pair of images i_a and i_b:

$$\Delta_{a,b} = i_b\pi_b - i_a\pi_a. \tag{IV.5}$$

Then

$$g_{a,N} = \Gamma^*\Delta_{a,a+1}. \tag{IV.6}$$

The effect of the above term is clear. The function g will be in the denominator of equation III.3, so the optimization favours small values of its integral around the boundary. This in turn means that g_N should be small, while g_D should be large. Now g_N is small at a point $p \in \mathcal{D}_S$ when the image function values at the projected points $\pi_a(p)$ for successive pairs of images are nearly equal. Thus g_N encourages good image function matches between successive boundaries.

The Denominator g_D. The denominator g_D is defined as a function on S^1 using

$$g_{a,D} = *_\Gamma\Gamma^* *_2 \mathbf{d}_2\Delta_{a,a+1}. \tag{IV.7}$$

The definition of $\mathbf{d}_2$ is as follows. For two images i_a and i_b, we can define a projection from $\mathcal{D}_S$ to the tangent space of $\mathcal{D}$, the single image domain, by $\phi = \pi_b - \pi_a$. (Recall that the image domains are identified via isomorphisms, here assumed identical, and that $\mathcal{D}$ is a Euclidean space, so that the subtraction here makes sense.) In the stereo case, the range of this map is a one-dimensional subspace of the tangent space to $\mathcal{D}$, because of the epipolar constraint. In the case of motion, the range is the whole of the tangent space. The fibres of this projection are then parallel planes that foliate $\mathcal{D}_S$. These are the planes of constant disparity or constant velocity. We can define an exterior derivative and a Hodge star within these fibres: these are $\mathbf{d}_2$ and $*_2$.

The derivative measures how fast the image function difference changes if the disparity or velocity are kept the same. By taking the Hodge star and

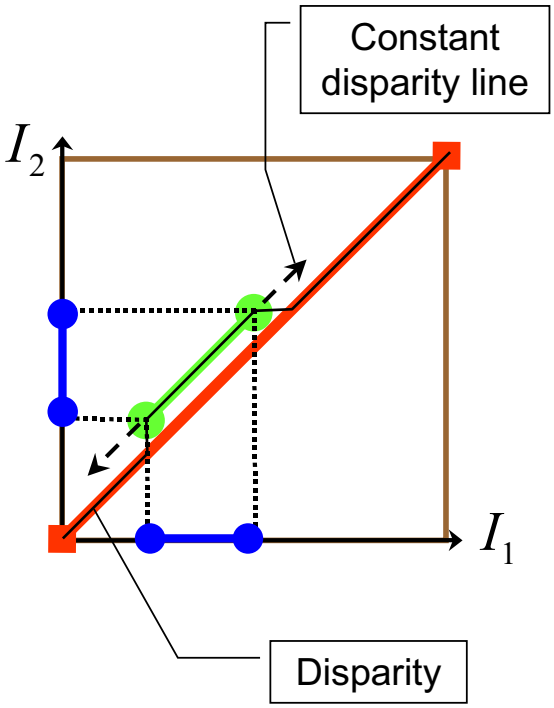


FIGURE 19. The figure shows a slice through the space $\mathcal{D}_S$ in the case of stereo. The constant disparity surface is shown, along which the derivative in equation IV.7 is taken.

pulling back by Γ, we then project this rate of change perpendicular to the boundary (but still at constant disparity or velocity). The final Hodge star $*_\Gamma$ is simply to convert to a function on the circle. The function g_D is thus large if the image function difference changes quickly when we take a small step away from the boundary in a normal direction, but remaining at the same disparity or velocity value. This is the behaviour one expects at a discontinuity. At the boundary in $\mathcal{D}_S$, the image function difference should be small (g_N sees to that). As we step away from the boundary, the disparity or velocity changes rapidly, meaning that if we now evaluate the image function difference away from the boundary at the same disparity or velocity as on the boundary, we are comparing points that do not correspond. Hence we can expect the image function difference to be much larger. A large value of g_D thus suggests that we are at a point of high disparity or velocity gradient. Figure 19 illustrates the geometry involved.

Now $g = \frac{g_N}{g_D}$ as defined above is a function on S^1, and its square, incorporating the Hodge star as in the denominator of equation III.3, transforms correctly and has all the required invariances.

The energy thus tends to find moving objects because stationary objects, while matching well on their boundaries (g_N small), also tend to match well just outside their boundaries, since the background is the same in all images. Thus the derivatives in g_D will be close to zero. Moving objects, on the other hand, have different regions of background just outside their boundaries in the different images in the motion sequence or stereo pair. While there is no guarantee that these will be different, thus rendering the derivative in g_D large (a counter-example would be a uniform background), in general there is no reason to expect the match to be good either. In that case, g_N will be small and g_D will be large, making the denominator in the energy functional small. Optimization will then favour these boundaries.

Many variations on the above theme are possible. We can compare image function gradients on the boundaries. We can favour boundaries having constant disparity (fronto-parallel) or velocity (no deformation): we use $g_a = \Gamma^* \mathbf{d} \|\phi\|^2$. We can favour far away or nearby boundaries using an increasing or decreasing function of ϕ. We have mentioned looking for low acceleration motions. Any combination of these is also possible.

5. ALGORITHMICS

Just as in the case of single images, we will solve the above optimization problems by embedding a graph in the image domain. The images of the cycles in the graph then constitute a subspace of the space of boundaries in the image domain over which we wish to optimize. The algorithm will find the global optimum of the energy functional over this restricted space of boundaries. If the embedded graph is dense enough in the image domain, this will be a good approximation to the global optimum over the full space of boundaries. As usual, the pixel scale provides an upper bound to the useful density of the embedding.

5.1. *Graph Structure*

The graphs that we use in the cases of stereo and motion are derived from the following construction. First, we define a graph G_a in each image domain $\mathcal{D}_a$. The graphs in each image are the same as we used in the last chapter: they have vertex sets the pixels in that image, and directed edge set the eight neighbours of each pixel. Just as we can form products of the image domains, we form the product of these graphs over the set of single images to obtain a $2n$-dimensional rectangular lattice embedded in $\mathbb{R}^{2n}$. From this point, the

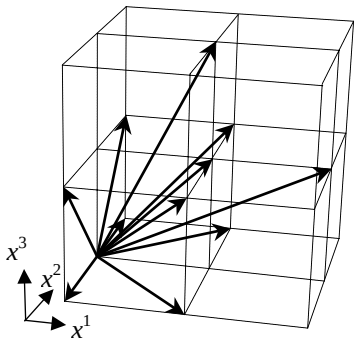


FIGURE 20. Half the neighbourhood structure of the graph used in the experiments. It is embedded in the three-dimensional space illustrated in figure 18. The other half is generated by reversing the signs of these edges (viewed as vectors in $\mathbb{R}^3$.)

cases of stereo and motion diverge slightly, and it is more convenient to treat them separately.

Stereo. For stereo, $n = 2$. However, we also have the embedding of $\mathbb{R}^3$ in $\mathbb{R}^4$ due to the epipolar constraint, and we can pull back the embedding of $G_1 \times G_2$ along this embedding. This is equivalent to embedding a three-dimensional rectangular lattice L^3 in $\mathbb{R}^3$, where L is a linear graph. The projections to the image graphs G_a are given by projection on the first and last two factors of this product. This is entirely analogous to the continuous case. The edges in the lattice are formed in the normal way, with two exceptions. First, we remove edges that violate the constraint in equation IV.2, in order to enforce the ordering constraint. Unfortunately, in the way we have constructed the graph, this eliminates all edges that could change the disparity. In order to avoid this discretization effect, we introduce extra edges to second neighbours, so that the overall out-degree of each vertex is 20. Figure 20 shows one half of the out-edges for a vertex. The rest of the edges are obtained by changing the sign (as vectors in $\mathbb{R}^3$) of those shown. Note the additional second neighbour edges.

For computational efficiency and not for reasons of principle, we restrict the size of the graph in common with many conventional stereo algorithms, by restricting the magnitude of the disparity. This selects a slice of $\mathbb{R}^3$ close to the diagonal and discards the rest. This eliminates many of the vertices in the graph, although the edge structure remains the same in the vertices that are left. Note that this is not a heuristic in the algorithmic sense. The algorithm will still find the global minimum over the space of cycles in this graph. If

the maximum disparity we allow is large enough, the output we obtain will be unaltered.

Motion. In the case of motion we have no epipolar constraint, and so the above reduction in dimensionality does not occur. In principle, we can then go ahead with the $2n$-dimensional rectangular lattice, applying the algorithm to this graph. This however is not practical given today's computational resources. For a 256×256 pixel image, the number of vertices in the graph for three images is approximately 10^{14}, and the number of edges is about 1000 times this. Clearly, to say the least, this is beyond the memory capacities of present computers, and moving storage offline brings the algorithm to a standstill. To reduce this graph to a manageable size, we restrict it in two ways. First, we limit the allowed velocities of points on the boundary. This says that the boundary does not move too far from frame to frame, which is true by assumption. Geometrically this is the same type of restriction that we imposed on the disparity in the stereo case. It means that the boundary lies in a region of the graph close to the diagonal, which of course represents no movement at all. Using such a slice we can dramatically reduce the size of the graph, bringing it to roughly the size of the stereo case with no limits on the disparity. To further reduce the size, we impose another restriction, this time on the the magnitude of the 'time derivative' of the tangent vector to the boundary. Since we are dealing with a discrete time, this means that we impose a restriction on the difference between the tangent vectors to the boundary at corresponding points in successive frames. Thus if γ'_a and γ'_{a+1} are the projections of the tangent vector Γ' to the boundary in two successive frames, we impose that $\frac{|\gamma'_{i+1}-\gamma'_i|}{|\gamma'_i|} \leq 1$ (the division by $|\gamma'_i|$ is necessary to render the condition independent of the choice of representative Γ). This is the statement of the condition in discrete time and continuous space. In the graph, this condition causes us to eliminate some edges. The constraint, which as we have said is imposed for reasons of computational efficiency and is not necessary in principle, does however have a sensible interpretation. It amounts to constraining the amount by which the shape of the boundary can change from frame to frame.

6. Demonstrations

In the demonstrations, we chose the one-forms $A_a = *_a \mathbf{d} i_a$, where $*_a$ is the Hodge star on image domain $\mathcal{D}_a$. The edge weights for the numerator were computed in the same way as for single images, by taking the cross-product of the edge, viewed as a tangent vector, with the image function gradient at the edge as computed using Gaussian derivatives at a scale of a few pixels. The edge weights for the denominator were computed by taking the

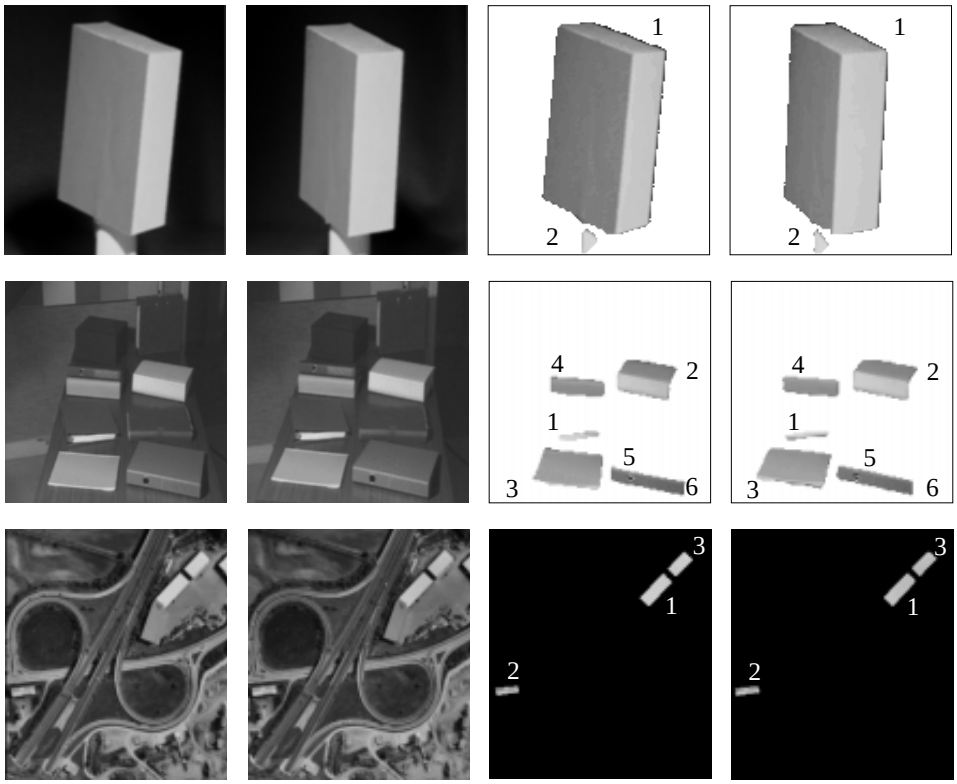


FIGURE 21. The two left-most images in each row are the original stereo pair (from top to bottom: 128×144, 148×148, and 230×260 pixel 8-bit gray-scale images). The two right-most images are the extracted regions in each image of the pair. The numbers indicate the order in which the regions and boundaries were found when following the iteration procedure outlined in the text.

product of the length of the edge with the square of the function value at the edge. Note that there are no free parameters in the model. The algorithm was iterated as in the case of single images, and the boundaries shown in the figures are numbered in the order in which they were found.

In figure 21, the results of running the algorithm on some stereo pairs are shown.

In the stereo case, with the disparity limited to say 10 pixels, the number of vertices in the graph for a 256×256 image is approximately 10^6. The number of edges is, as discussed, an order of magnitude higher than this, bringing the complexity of the algorithm to 10^{13}. The storage needed is of

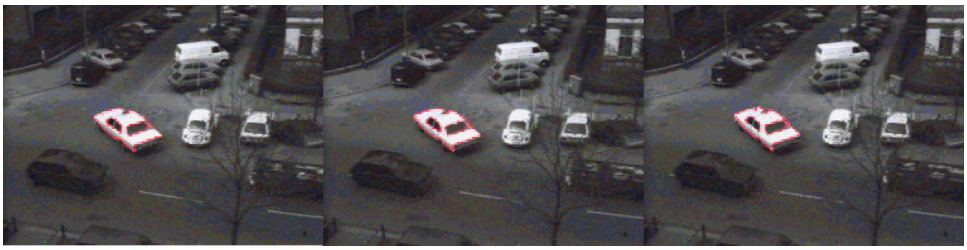


FIGURE 22. The images show three consecutive frames of a motion sequence of images. The boundaries are shown in red.

the order of 10^7. (The constants involved could add an order of magnitude to these figures.)

In figures 22 and 23 the results of running the algorithm on a motion sequences are shown. The algorithm was run on successive triples of images in the motion sequence. In figure 22, one of these triples is found, while in figure 23 a single frame is shown, along with the largest region found. The motion was limited to one pixel per time step.

Figure 24 and 25 show three consecutive frames and one frame respectively of another motion sequence.

Since the algorithm was run on triples of image in the motion case, the complexity is correspondingly higher. Allowing one pixel of motion between frames means that the number of vertices is approximately 10^7, while the number of edges is an order of magnitude or two greater. The time complexity is thus approximately 10^{16}. This may seem enormous, but the asymptotic time complexity is a somewhat meaningless figure in this instance, since the negative cycle algorithm never runs to completion. Of far more importance in practice is the storage requirement, which in this example is approaching one gigabyte. This is more or less the limit that can be handled with the equipment available. The algorithm took of the order of ten minutes to complete in the stereo and motion cases.

As we have mentioned, the results of the boundary extraction can be used to constraint a dense correspondence computation. An example of the full disparity result with and without using the boundary information obtained from the model is shown in figure 26. To compute the full disparity, a maximum-flow type algorithm with a linear smoothing term [IG98] was used. The boundary information from the model was used to fix the disparity and to weaken the smoothing on the boundaries. The figure shows a close-up of part of the last example in figure 21, together with the region found by the model. In the right half of the figure are shown the results of running a maximum flow stereo algorithm on the image, first without any further

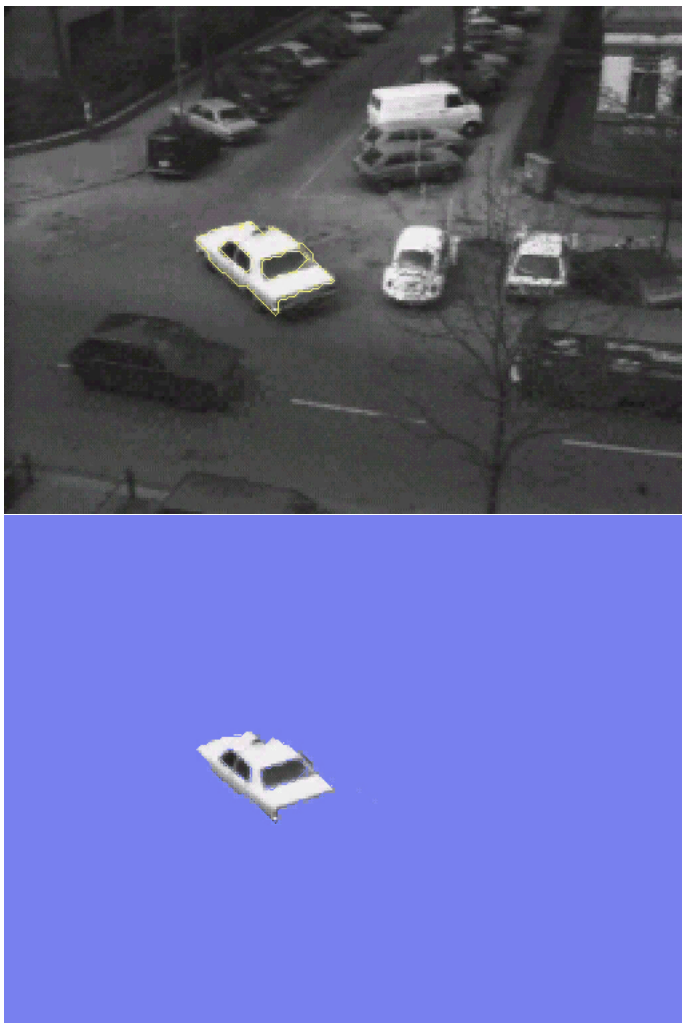


FIGURE 23. One frame of the previous three is shown, along with the largest region extracted.

data, and second with the boundary information used to limit the disparity and weaken the smoothing. The second building, which is smoothed over without the boundary data, is discovered when that data is used.

7. POSTSCRIPT

The application of equation III.3 to multiple images is best thought of by analogy with the object recognition task we have been studying. It is not a substitute for a dense correspondence when one is needed. Instead it is a way of using the extra information inherent in having more than one image to

FIGURE 24. The images show three consecutive frames of a motion sequence of images. The boundaries are shown in red.

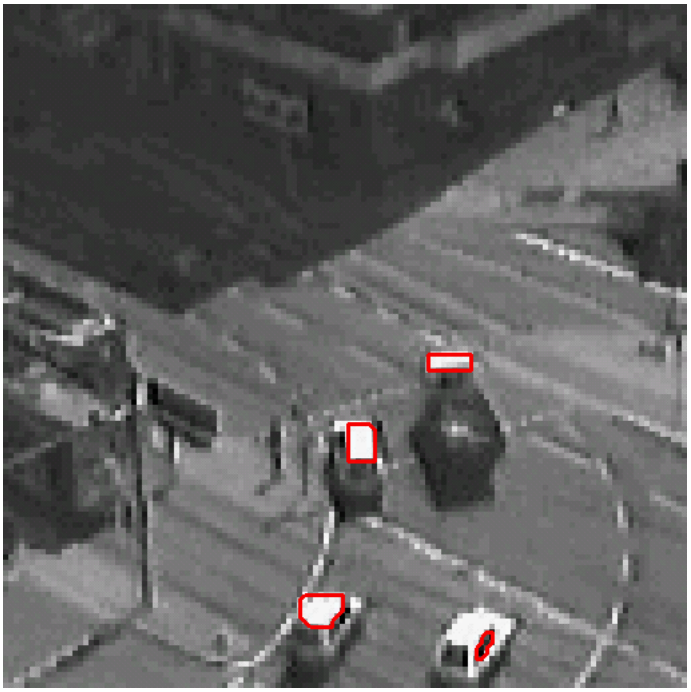


FIGURE 25. One frame of the previous three is shown.

aid in boundary extraction. Although we do not pursue it here, an extension in the multiple image case analogous to the work in the next two chapters is likely to be possible. Instead we now return to the main line of argument, and move to the second stage of the model: part structures.

FIGURE 26. From left to right: A close up of the left image in the last example in figure 21; the region found using the model; the full disparity result found without using the boundary information; and the full disparity result found using the boundary information.

CHAPTER V

ARTICULATED MODELS

We focus on the space $\mathcal{P}$ of part structures and regions. After a review of previous work in shape description, we describe the self-matching model for part structure. Extending the model to real images is straightforward theoretically but not algorithmically. Interpreting self-matchings as boundaries in $\mathcal{D} \times \mathcal{D}$ allows the application of the energy density approach in a limited instance. Interpreting them as hyperpaths in a hypergraph embedded in $\mathcal{D} \times \mathcal{D}$ allows a general treatment for real images, but we must sacrifice intensive energies. A category of hypergraphs is defined that differs slightly from common usage. The necessary hypergraph theory is developed, leading to a general 'shortest hyperpath' algorithm.

As we discussed in chapter II, psychological research suggests that the fundamental classification of objects into categories is based in large part on shape. These are the 'basic' categories, the categories that people recognize most speedily. This suggests strongly that the description of object shape is an important part of object recognition. We have already anticipated this fact by taking as one of the important features of our model the notion of a 'part structure', and a space of such structures, $\mathcal{P}$, and assuming that they are important for the recognition of human figures in images. However, to date we have not clarified what we mean by this exactly, and we have not described any probability distributions on part structures. The present chapter rectifies this situation.

To recap: the probability distributions in which we are interested are of the form $\Pr(P = p|\pi_{\mathcal{P}}(P) = \pi_{\mathcal{P}}(p) \,\&\, I = i)$, which is the appropriate factor of equation II.2. As that equation illustrates, such distributions can be combined with distributions $\Pr(B = b|I = i)$ generated by functionals such as those of chapter III to create distributions $\Pr(P = p|I = i)$. As always, we make the assumption of equation II.4, that $\Pr(P = p|\pi_{\mathcal{P}}(P) = \pi_{\mathcal{P}}(p) \,\&\, I = i)$ depends on the image function only through its restriction to the region $\pi_{\mathcal{P}}(p) \in \mathcal{B}$. Furthermore, for the moment anyway, we assume that $\Pr(P = p|\pi_{\mathcal{P}}(P) = \pi_{\mathcal{P}}(p) \,\&\, I = i) = \Pr(P = p|\pi_{\mathcal{P}}(P) = \pi_{\mathcal{P}}(p))$: the

part structure of a boundary is independent of the image function once we know the geometry of the region. In terms of energy functionals we are thus looking for a functional $E_{\mathcal{P}}(p, b)$, where it is understood that $\pi_{\mathcal{P}}(p) = b$. Before considering such functionals however, we must define their domain $\mathcal{P}$, and before doing that it is useful to review previous work concerned with the description of two-dimensional shape to get an idea of the limitations and advantages of various approaches. In particular, the *symmetry axis* is of great interest to us, since it bears a close resemblance to the part structure description used here.

1. Previous Work

Proposals for the description of two-dimensional shape can be classified along two axes. On the one hand, there is the distinction between processes that use primitives and those that do not . On the other hand, there are those processes that concentrate on the boundary of a shape, and those that utilize the interior, usually called "region-based." The best exemplar of region-based, primitive-less processes is the symmetry axis. Although closely related, we do not discuss in this section the work of Liu, Geiger and Kohn [Liu97, LGK98, LG97], which introduced the notion of self-matchings as a shape descriptor. Section 2 and the first part of section 3 are a review and interpretation of their work.

1.1. *Symmetry Axis*

The medial axis of a curve in the plane is the subset of the plane formed by the centres of circles tangent to the curve at two or more points. In the case of closed curves, the symmetry axis is the subset of the medial axis formed by the circles lying in the interior of the curve. Both axes can also be thought of as appropriate loci of critical points of the distance function from the curve. The latter characterization allows an analysis of the possible topologies of the critical points. It can be shown that the symmetry axis consists of curve segments that join at triple points, in the case of simply connected regions forming a tree structure. Higher order points are possible but these are not transversal, and they split on slight perturbations into a number of triple points.

The first paper to bring the symmetry set/medial axis to the attention of biologists and researchers in computer vision was by Blum [Blu73]. He described the symmetry axis, and its definition via the *grassfire transform*. One imagines the interior of the curve filled with grass, and a fire starting everywhere on the boundary. This fire burns inwards and eventually burns itself out where it reaches grass that has been burnt already. These burn out points are the symmetry axis. Blum emphasized that the symmetry axis

enabled the construction of shape from *morphemes*, shape primitives such as disc, worm, and wedge. Siddiqi and Kimia picked up this suggestion in their paper on the shape triangle, in which they postulated three processes operating on shape: parts, protrusions, and bends [SK95, SKT96, SKTZ99]. Beginning with ad hoc definitions of parts stemming from the shape triangle, the same authors and their collaborators developed a beautiful shape theory based on the singularities created by curve evolution (yet another way to define the symmetry axis) [KTZ95]. The simplest example is when a curve evolves at constant speed in the direction of the inward normal at each point. This is the grassfire transform, and the locus of singularities is therefore the symmetry axis, except that in the treatments referenced above the extra information about when the singularities (*shocks*) are formed becomes extremely important also. When the order of shock formation is taken into account, a shock grammar can be developed and used to help speed up the process of shock extraction by ruling out impossibilities [SK96]. In addition, these papers (in particular [KTZ95]) introduce a term that changes the speed of the evolution in proportion to the curvature at each point of the shape boundary. Such a term smoothes the contour, and does not by itself develop shocks. It can be used to develop a two-dimensional space of shapes, the reaction-diffusion space, which consists of the space of shapes generated over time using different ratios of reaction and diffusion terms. A second space, based on running the reaction and diffusion processes forwards in time and then the reaction processes backwards for the same amount of time, in order to re-construct the parts of the contour not smoothed by the diffusion term, is even more elegant. Subsequent work has used the resulting tree structures to perform shape comparison through clever approximation algorithms [SSDZ99, PSZ99], and current work is focused on extracting symmetry axes from real images [TSK97, ASZ99b].

The problem with this work is paradoxically to be found in the beauty of the geometric objects involved. The definition of the symmetry axis seems so clear and necessary that developing it and adapting it is difficult. A wager must be made that this geometric structure captures the notion of shape in its entirety. In a limited sense this is true, in that knowledge of the nature of the shocks and their times of formation does enable reconstruction of the shape, but it is not completely clear that geometrical information alone is enough to capture the notion of shape and part structure. Functionality may also play a very important role. One example is the 'prickly pear' sequence of images shown in figure 27, taken here from [KTZ95]. The sequence of pear shapes came originally from [HR85]. Here the prickles on the skin of the pear are removed by smoothing using curvature evolution, and then the reaction terms are used to reconstruct the shape without the prickles, thus

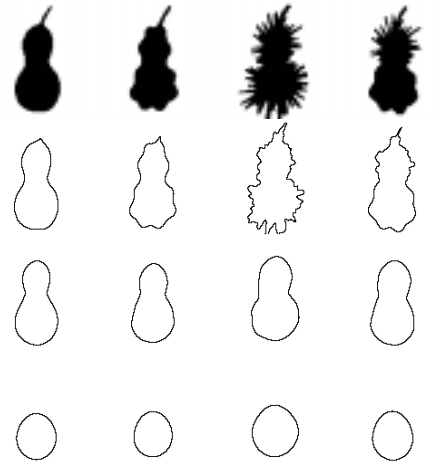


FIGURE 27. The prickles on the skin of the pear are removed by smoothing using curvature evolution, and then the reaction terms are used to reconstruct the shape without the prickles, thus capturing the basic shape at a larger scale. The stalk however is missing.

capturing the basic shape at a larger scale. This is fine as far as it goes, but it results also in the removal of the pear stalk. If one is thinking of the removal of the prickles in terms of the removal of noise of some sort, then the removal of the stalk is a mistake. The stalk is an important part of the pear not because of its shape or size but because of the function it performs. (The same can be said of many parts of other objects.) Capturing this deeper notion of part identity is very hard. It occupies a similar position to image semantics. There may be a way, given enough formalized experience with the world, to define this functional nature of parts in a way that does not depend on human knowledge and usage (or at least depends in a much subtler way on theories of information and so on), but at the present time it is hard to rely on anything other than the fact that we know that pear stalks are important, and that this fact is in itself an important attribute of a pear. This knowledge must be catalogued: it is not deductive, but inductive.

An additional problem is that some shapes do not to lend themselves to description in this way. While it is of course true that the symmetry axis and shock graph can be computed for any shape whatsoever, there is a problem with the stability of the representation. Even with articulated objects, it is well known that the topology of certain parts of the symmetry axis is unstable with respect to small changes in the shape. These areas may be identified with the

help of the elegant concept of *ligature* and *semi-ligature* developed in [ASZ99a]. Unfortunately, for some shapes, these unstable areas constitute almost all of the symmetry axis. This demonstrates what perhaps is intuitively obvious, which is that different types of shapes require different representations.

Even if, as these arguments suggest, the symmetry axis is not enough to deal completely with shape, it may still be extremely useful. The question is then how to alter or develop it. Unfortunately this seems to be another hard question. The self-matching approach described in this chapter on the other hand is considerably more flexible. It will be defined by an energy functional, which can be changed and added to in order to incorporate new and differing criteria. It can also be combined with terms that couple the shape description to other objects, for example images, as we shall see later. Of course the greater flexibility and extensibility of the shape description brings with it a greater arbitrariness; there are parameters to be fixed for example. This can only be viewed as a drawback however if we make the leap of faith that the symmetry axis is *the* shape descriptor par excellence.

1.2. *Primitives*

An example of a boundary-based, primitive-less model is the work of Richards, Dawson and Whittington [RDW86], in which they describe shape using the curvature extrema of the boundary. This is essentially a kind of 'poor man's' symmetry axis. A similar scheme is described in Marr and Nishihara [MN78].

Of the work that uses primitives to describe regions or volumes, the most famous is surely that of Biederman [Bie85]. In this work, an attempt is made to reduce the description of shape to a few (36) components, or *geons*, that are then put together in a grammatical fashion to form objects. According to this point of view, one must first describe parts and only then attempt to decompose objects into them. One can view mathematical morphology as a boundary-based system with primitives also, since it develops boundaries by adding discs and other structuring elements [Ser82]. Another approach, due to Leyton, develops a grammar for shape [Ley88].

The problem with all primitive-based shape theories is they have to assume a dictionary of primitive shapes, the justification for which is usually weak. The geons used by Biederman for example are simple geometric volumes. There seems no particular reason why these should be selected on either mathematical or biological grounds. There may be some computational justification, but this is unclear.

1.3. *Shape Comparison*

It is clear that shape comparison is an important aspect of object recognition. Almost all object recognition schemes have built into them in some way or other the idea of comparing structures to a library of 'known' structures, and often the library is shape-based. Unfortunately, it is not clear how to compare shapes. In this section we argue briefly that part structures are essential for such a comparison, and that comparisons, although perhaps derived from a metric, are not symmetric or transitive.

Mumford has written a much-cited paper on mathematical theories of shape [Mum91], in which he provides an overview of methods for shape description and comparison, including filters (moments, Fourier transform, wavelets), curvature-based descriptions (including the work just described), and metrics on the space of shapes. After describing a few of the infinite number of mathematical possibilities, he points out that similarity is not a metric anyway, at least not in human or pigeon perception. He goes on to describe experiments that demonstrate the asymmetry of perceptual similarity judgments, and their context dependence. For example, in experiments reported in a paper by Tversky [Tve77], people said that the number 99 was very similar to 100, but that 100 was not very similar to 99. Similar results were observed in pigeons. As an example of context dependence, when people were asked whether Austria was most similar to Sweden, Norway, or Hungary, 60% chose Hungary. When asked whether Austria was most similar to Sweden, Poland, or Hungary, 49% chose Sweden. (Other experiments showed the failure of transitivity.) In these cases, the comparisons seem to use different attributes of the compared objects in different contexts. The effect can perhaps be described by using the phrase "when viewed as a . . . ". For example, 99 is similar to 100 when viewed as a 'ordinary number', but 100 is not similar to 99 when viewed as as 'round number'. Austria is more similar to Hungary than Sweden when viewed as a Central European country as against a Scandinavian country, but when viewed as a Western European country as opposed to an Eastern European country it is more similar to Sweden than Hungary or Poland. The categories that are used for the comparison seem to be created by the objects being compared in an asymmetric way. If we are comparing 99 to 100, then 100 is the exemplar, and this is thought of as a 'round number' rather than just an integer. When 99 is the exemplar nothing springs to mind except 'ordinary number'. The same type of effect can be observed in the country comparison. This suggests that the comparison of shapes works as follows. When shape A is compared to shape B, B is the exemplar. B has certain principal identifying characteristics that define a subspace of the space of shapes (all shapes that share the principal

identifying characteristics of B). The similarity of A to B is then measured by the distance of A to this subspace. (If A is already in the subspace, then perhaps secondary characteristics come to the fore.) If A had been picked as the exemplar, then a different subspace would likely have been defined, and hence the 'distance from A to B' is not the same as the 'distance from B to A'. What determines which members of the comparison are used as exemplars or combinations of exemplars is not clear. Sometimes it seems that it is simply the word order used to phrase a question. We will discuss this asymmetry in the shape comparison measure further in chapter VI, where we will effectively define just such an asymmetric measure of similarity using exemplars.

One can compare shapes at the level of their boundaries, by defining a functional on some space of maps between curves, and then minimizing for a given curve pair. This defines a metric on the space of curves. Some processes use special points on the curve and define the energy in terms of these, whereas others define an integral over the whole curve. Of note here is the work by Basri, Costa, Geiger, and Jacobs [BCGJ95]. They make a thoroughgoing analysis of what is desirable and undesirable in a curve-based comparison method, and devise several energy functionals to implement the principles. They try hard to make a curve-based comparison take into account the part structure of the shapes but show that this cannot succeed in all cases. They give several clear examples of why a part-based process would be more successful. The part structure of an object is part of its identity, and ignoring this structure can clearly cause problems. For example, using a contour as a model of a dog and attempting to match it to another such contour can lead to absurdities such as a piece of the head mapping to a piece of the foreleg, or a piece of the tail to a piece of the hind leg. Models at several levels of abstraction are needed to prevent such mistakes being made at all levels, not just at the lowest.

2. Representations for Shape

In this section, we finally define the space $\mathcal{P}$. The model for part structure will be that of a *self-matching*, first introduced in [Liu97]. This representation combines the strengths of the symmetry axis as a shape descriptor with the flexibility and extensibility of an optimization framework. We will consider single images only, so that the domain of the image, $\mathcal{D}$, is a subset of the plane. Although much of what we say can be generalized, this is not as trivial as it was in chapter III. This is because the use of parallel transport to compare tangent vectors at different points plays an important role in the rest of the chapter. In Euclidean space, which is an affine space, this is a simple matter

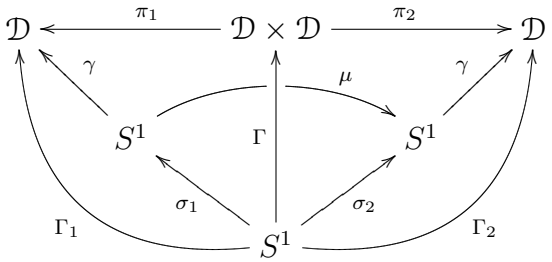
since the tangent bundle collapses to a tangent space. We therefore restrict ourselves to Euclidean space.

2.1. *Symmetry*

We begin by taking a detour into the notion of symmetry, as we will use this as the basis for describing shape and part structure.

A mirror-symmetric region in the image domain defines a diffeomorphism μ from its boundary to itself. This is a representation of the symmetry group $\mathbb{Z}_2$. Every point on the boundary has a partner given by its mirror reflection through some line, the axis of symmetry. We thus have the following diagram:

(V.1)



The diagram bears some explanation. The map μ between the boundary and itself can be represented as two embeddings of the circle Γ_1 and Γ_2 that share a common image γ, and co-images σ_1 and σ_2 respectively. These must be such that $\mu = \sigma_2\sigma_1^{-1}$. The map into the product $\mathcal{D} \times \mathcal{D}$ induced by Γ_1 and Γ_2 is denoted Γ.

The sharing of a common image is not enough to make Γ a representation of a reflection symmetry. To explain the structure a little more and to prepare for the discussion of invariance in section 3.1, we characterize the space $\mathcal{G}$ of Γ in different ways. Let $\tilde{\mathcal{B}}$ be the space $\mathcal{B} \times \mathrm{Diff}(S^1)$ of γ. Let $\mathrm{Diff}(S^1)$ be the space of continuous piecewise-smooth automorphisms of the circle as in chapter III. Let $\mathcal{M}$ be the space of possible μ. Then we have the following three projections:

(V.2)
$$\tilde{\mathcal{B}} \times \mathcal{M} \times \mathrm{Diff}^+(S^1) \longrightarrow\!\!\!\!\!\rightarrow \tilde{\mathcal{B}} \times \mathcal{M} \longrightarrow\!\!\!\!\!\rightarrow \mathcal{G} \longrightarrow\!\!\!\!\!\rightarrow \mathcal{G}/\sim$$

$$\langle \gamma, \mu, \epsilon \rangle \rightsquigarrow\!\!\!\!\!\rightarrow \langle \gamma, \mu \rangle \rightsquigarrow\!\!\!\!\!\rightarrow \Gamma \rightsquigarrow\!\!\!\!\!\rightarrow \Gamma/\sim$$

Here $\langle \gamma, \mu, \varepsilon \rangle$ is the representation in terms of $\sigma_1 = \varepsilon$ and $\sigma_2 = \mu\varepsilon$. The equivalence class of the σ projects down to the space $\mathcal{M}$ of possible μ. Note that this projection means that we are dealing with a boundary in $\mathcal{D} \times \mathcal{D}$, since it absorbs a change from ε to $\varepsilon\epsilon$. Given an embedding γ, and a map μ, there is an equivalence class in the space $\tilde{\mathcal{B}} \times \mathcal{M}$ given by the action of $\mathrm{Diff}^+(S^1)$: the pair $\langle \gamma, \mu \rangle$ is equivalent to the pair $\langle \gamma\epsilon, \epsilon^{-1}\mu\epsilon \rangle$. Dividing

out by this equivalence brings us to the space $\mathcal{G}$ of Γ. Finally, note that as a representation of $\mathbb{Z}_2$, $\mu^2 = \mathrm{id}_{S^1}$. This in turn means that $\sim \Gamma = \Gamma$, where the twist arrow $\sim$ reverses the order of the factors in a product: Γ is a symmetric relation. We can thus quotient by the action of the twist arrow (which is an action of $\mathbb{Z}_2$), and project to the quotient space $\mathcal{G}/\sim$. In section 3.1, these considerations will translate into invariance requirements for our energy functionals, since these will be written as functionals of γ, and the σ.

To pin down the space $\mathcal{G}$ further, we describe the constraints on μ. The space $\mathcal{M}$ is a subspace of $\mathrm{Diff}(S^1)$, and it is proper: each μ must be orientation-reversing, and it must be an isometry of S^1 using the metric induced from $\mathcal{D}$. This means that μ must have at least two fixed points, whose images lie in the diagonal of $\mathcal{D} \times \mathcal{D}$. These are the points in S^1 mapping to the points in $\mathcal{D}$ where the axis of symmetry cuts the boundary at its extremes. In addition, the tangent vectors to the boundary at a point and its image bear a special relation to one another. This relation can be characterized by saying that the tangent vectors are *co-circular*, meaning that there exists a circle passing through the point and its image such that the tangent vectors to boundary and circle are equal at each point.[1]

2.2. *Local Approximate Symmetry*

Clearly the idea of symmetry captures important aspects of shape, and may be of use to us in describing the structure of regions. This suggests that we take the map μ as the starting point, and relax the above constraints in order to describe more general structures. As soon as the hard (logical) constraints are relaxed, we will need to replace them with soft (probabilistic) constraints, and it is from here that the energy functionals arise, as measures of deviation from the above ideals.

The monotonicity requirement seems too fundamental to relax, at least locally: it is what characterizes reflections as opposed to rotations for example. For reflections, μ is self-inverse, but it is not clear what the sense of relaxing this would be. This leaves the fact that the maps are isometries, that they pair co-circular points, and the fact that they are continuous.

[1] We can think the same way about a rotational symmetry, although the nature of the map from the boundary to itself is slightly different. For rotational symmetry, μ is orientation-preserving and has no fixed points. It too must be an isometry of the image of γ using the induced metric. We do not analyse the rotational case further at present since it is not our central interest. However, the interaction between the two forms of symmetry and their possible joint use in describing object shape is an extremely interesting subject. It may enable the construction of a geometric shape descriptor able to cope both with articulated objects and the blob-like shapes ('amorphous') that cause the symmetry axis to fail.

The relaxation of the isometry and co-circularity requirements is very natural, since it allows for a little bit of 'give' in the symmetry maps. Slight distortions of the region away from perfectly symmetric should be penalized, but not ruled out. This is especially true when we are dealing with real data rather than mathematical abstraction.

It is harder to see how to relax the continuity requirement. The idea of doing so will be to describe 'local' symmetry, with different parts of the boundary being symmetric with respect to different axes separated by isolated discontinuities. (This will force us to relax the monotonicity constraint at isolated points also.)

Luckily our application gives us some information. We are interested primarily in describing articulated objects. These types of objects typically consist of elongated parts with an approximate mirror symmetry, but the different parts do not share a mirror. As we discussed in section 1, there is already a representation of such shapes, the symmetry axis, that captures this insight. The symmetry axis results in a tree structure that describes the shape of the object. This tree structure has vertices of degree at most three, unless a very high degree of symmetry is present. This suggests that we permit the map μ to have discontinuities that would allow such behaviour. This is possible as follows.

The key point is that discontinuities do not occur by themselves, but in groups of three. Suppose, for example, that there is a discontinuity at point p on the boundary. Let $\mu(p) = q$ and let the limit of μ from the other side of the discontinuity be point r. Then we insist that $\mu(q) = r$ and $\mu(r) = p$, and that the limits from the other directions to these points are p and q respectively. There are thus discontinuities at q and r also. By taking the closure of the graph of μ, we end up with three points all matched to each other, as opposed to two in the normal case. (Thus what seemed like a very particular relaxation of the self-inverse requirement becomes instead a relaxation of the nature of the relation.) An example of a map μ obeying these restrictions is shown in figure 28, while figure 29 shows how the map describes a part structure.[2]. We call such maps *self-matchings*. The discussion of equivalence classes summarized in diagram V.2 remains the same, except that the space $\mathcal{M}$ has been enlarged considerably to allow local approximate symmetries. This in turn enlarges both $\mathcal{G}$ and the space of the σ. The space $\tilde{\mathcal{B}}$ of course is unaltered, since that represents the space of boundaries, $\mathcal{B}$. This in turn means that we are still dealing with the action of Diff(S^1) and not a larger group.

[2]We will continue to speak of maps, with the completion to the ternary relation understood where it is necessary

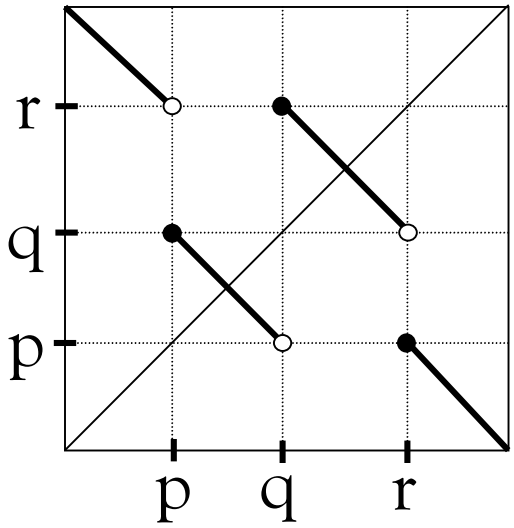


FIGURE 28. The figure shows the graph in $S^1 \times S^1$ of the map from S^1 to itself that constitutes the self-matching. The top and bottom, and left and right sides of the square are identified to form the torus. Open circles signify limit points not in the map (but in its closure), while closed circles represent limit points in the map. A single triple of discontinuities is shown.

Thus, for a given boundary $\partial R \in \mathcal{B}$, the space of part structures is $\mathcal{M}/\mathrm{Diff}(S^1)$. This is the fibre of $\mathcal{P}$ over $\pi_{\mathcal{P}}(p) = \partial R$. The space $\mathcal{P}$ is thus the union of these over $\mathcal{B}$, and is equal to $\mathcal{G}/\sim$.

This completes the definition of $\mathcal{P}$. We now move on to discuss energy functionals on this space.

3. FUNCTIONALS FOR PARTS

We have defined $\mathcal{P}$ as the space $\mathcal{G}/\sim$ of embeddings $S^1 \overset{\Gamma}{\rightarrowtail} \mathcal{D} \times \mathcal{D}$ whose projections $\pi_1\Gamma$ and $\pi_2\Gamma$ share a common image γ, and that satisfy such additional constraints as are necessary to produce self-matchings μ of the form described in section 2. As described at the beginning of this chapter, we will interpret a functional on this space, $E(\Gamma)$, as the negative logarithm of a probability distribution $\Pr(\Gamma|\gamma)$.

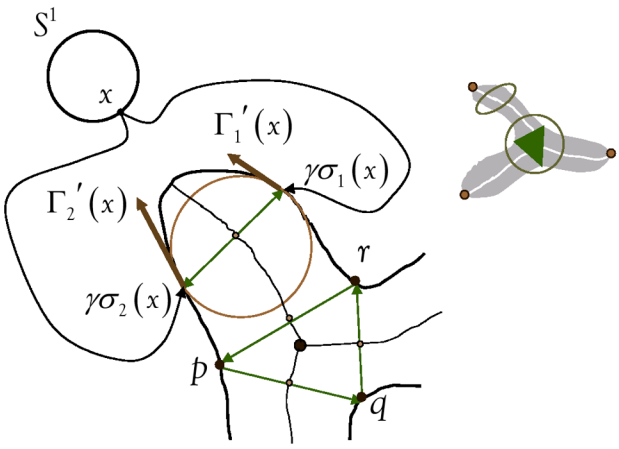


FIGURE 29. A point x is mapped to two points on the boundary. The points and tangent vectors to the boundary are shown along with an inscribed circle. For perfect symmetry these points should be co-circular: the sum of the tangent vectors should be perpendicular to the line joining them, and their difference should be parallel to it. Also shown is a triple of discontinuities. The right of the figure shows a representation in which the paths in $\mathcal{D} \times \mathcal{D}$ are mapped into $\mathcal{D}$ by taking the median point of each pair. The role of the discontinuities in the description of parts is clear.

3.1. *Invariance*

From the discussion summarized in diagram V.2 it is clear that if we write $E_{\mathcal{P}}$ as a functional on any of the spaces in that diagram other than $\mathcal{G}/\sim$, we will have some invariance requirements to impose. We will require invariance under the replacement of the σ by $\sigma\epsilon$, corresponding to the first projection. We will require invariance under the replacement of $\langle\gamma, \sigma\rangle$ by $\langle\gamma\epsilon, \epsilon^{-1}\sigma\rangle$ corresponding to the second projection. Finally, we will require invariance under the exchange of σ_1 and σ_2, corresponding to the third projection.

As we mentioned, these invariances already include the replacement of Γ by $\Gamma\epsilon$. In addition, we will require invariance under the action of isometries, and simple scaling behaviour under global changes of scale, just as in section 1.1. These actions take the same form on both factors in $\mathcal{D} \times \mathcal{D}$. In other words, we are not expecting invariance under the action of different isometries on each factor.

3.2. *Incorporating Symmetry*

We have argued that a relaxed form (approximate, local) of reflection symmetry would be ideal for describing part structure using the space of maps $\mathcal{M}$ and that the energy functionals that we write down should be measures of deviation from ideal reflection symmetry. We argued also that the natural candidates for relaxation were the isometry requirement, the co-circularity requirement, and the continuity requirement.

We thus want functionals that measure the deviation of the maps $\mu \in \mathcal{M}$ from isometric, continuous maps taking points to co-circular points. We will do this by making the functional an integral over S^1 of three terms, each of which measures deviation in one of the above quantities. We want the measure to be one of total deviation, so that for each orientation we would like the measure to have a consistent sign. In fact, for algorithmic reasons to be clarified later we will construct orientation-independent positive energies. They will thus take the form illustrated in the last of equations III.2:

$$E(\Gamma) = \int_{S^1} \sum_{i \in \mathcal{I}} g_i *_\Gamma g_i \tag{V.3}$$

where the g_i are functions or one-forms on S^1. They might be pullbacks $\Gamma^* \phi_i$ by Γ of functions or one-forms ϕ_i on $\mathcal{D} \times \mathcal{D}$. [3]

We deal with the three terms in turn.

Co-circularity. Given a point $x \in S^1$, we wish to characterize the co-circularity of the two points $\gamma\sigma_1(x)$ and $\gamma\sigma_2(x)$. The condition is that the components of the tangent vectors along the line joining the two matched points should be equal and opposite, and that the components perpendicular to this line should be equal (recall that μ is orientation-reversing, so that the tangent vectors will be oriented as shown in the figure).

It is easy to devise a term that measures deviation from this situation. The line joining the two points is given by the T-valued function on $\mathcal{D} \times \mathcal{D}$ $\phi = \pi_2 - \pi_1$. Pulled back by Γ this becomes $g = \Gamma^* \phi = \pi_1 \Gamma - \pi_2 \Gamma = \gamma\sigma_1 - \gamma\sigma_2$. The magnitude of this vector measures the distance between the points, and because of this we will also be penalizing larger distances. This penalty is very intuitive. Consider a region with more than one reflection symmetry. Which is the more important? Figure 30 shows an example. It seems right that the symmetry that divides the region along its 'long' direction, and hence which minimizes the distances between the points, is the 'stronger'.

[3] Note that the Hodge star here is defined by the pullback by Γ of the product metric on $\mathcal{D} \times \mathcal{D}$ given by the inner product on T. In coordinates such as those used in section 1.2, this metric is $\sigma_1'(t)^2 \|\gamma'(\sigma_1 t)\|^2 + \sigma_2'(t)^2 \|\gamma'(\sigma_2 t)\|^2$

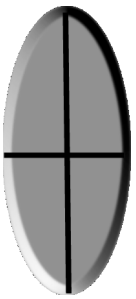


FIGURE 30. Of the two reflections, the one in the 'long' direction seems the 'strongest'.

The map Γ defines its own tangent map, Γ', which maps the tangent bundle of S^1 into T and gives us the tangent vectors at the two points. This is a linear map and is therefore a T-valued one-form on S^1. In fact it is the exterior derivative of Γ, $\mathbf{d}\Gamma$. We can therefore define the two T-valued one-forms $A_\pm = \mathbf{d}\Gamma_1 \pm \mathbf{d}\Gamma_2$. From this data we form the real-valued one-forms, $B_\parallel = (A_+, g)$ and $B_\perp = (A_-, g)_\times$ which are the sum of the components along the vector g and the difference of the components perpendicular to it respectively. Both these should be zero if the points are co-circular. The co-circularity energy $E_C(\Gamma)$ now becomes

$$E_C(\Gamma) = \ll B_\perp, B_\perp \gg + \ll B_\parallel, B_\parallel \gg \tag{V.4}$$

Note that the form of the co-circularity energy does not correspond exactly to the geometric constraint. We could adjust the energy so that we reproduce exactly this constraint by normalizing the tangent vectors to the curve. This is inelegant however, particularly since the form of equation V.4 acts also as a measure of deviation from isometry. We may choose γ so that its tangent map has magnitude unity everywhere. In that case, the tangent maps of the σ will be forced to have the same magnitude in order to reduce $E_C(\Gamma)$. This is precisely the isometry condition. This effectively obviates the need for a separate isometry term, although we now proceed to describe one.

Isometry. Let the real-valued one-forms $\{(\mathbf{d}\Gamma_i, *_\Gamma \mathbf{d}\Gamma_i) : i \in \{1, 2\}\}$ be denoted $\mathbf{B}_i$. Then the simplest choice for an isometry term that scales like the co-circularity term is:

$$E_I(\Gamma) = \ll B_2 - B_1, B_2 - B_1 \gg . \tag{V.5}$$

If the map μ were an isometry, B_1 and B_2 would be equal. By construction this term is invariant in all the right ways.

Continuity. The discontinuities occur at isolated points of ∂R, as described in section 2.2. To characterize them, we define a family of distributions on S^1. Denote the sum of delta functions at the points of discontinuity

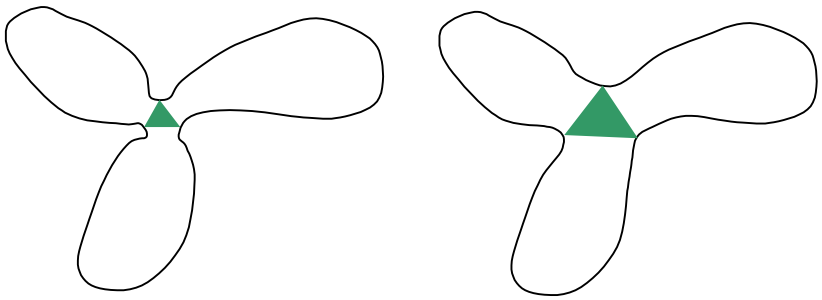


FIGURE 31. Part junctions with shorter discontinuities separate parts better and should be preferred.

by Δ. Define $S^1 \xrightarrow{J} T^2$ as $J(t) = \lim_{\tau \to t^+} \Gamma(\tau) - \lim_{\tau \to t^-} \Gamma(\tau)$, the $T \times T$-valued *jump* that takes place at each point, which is almost everywhere zero (the exceptions being the discontinuity points). Then the family of distributions is $\Delta_\alpha = \|J\|^\alpha \Delta$, where $\alpha \in \mathbb{R}^+$ and $\|\cdot\|$ is formed using the product metric on T^2.

We then penalize discontinuities in the following way:

$$E_{D,\alpha}(\Gamma) = \int_{S^1} *_\Gamma \Delta_\alpha \tag{V.6}$$

If $\alpha = 0$, the integral simply counts the number of discontinuities. As α becomes positive, we have a measure of the magnitude of the discontinuities. When $\alpha = 1$, the integral measures the total geometric distance jumped in $\mathcal{D}$, whereas for $\alpha = 2$ it measures the sum of the squares of the distances. The effect of this can be seen in figure 31. Penalizing larger jumps tends to make discontinuities occur at 'pinched off' parts of the boundary, which is what one would expect intuitively.

If $\alpha = 4$, then the scaling of $E_{D,4}$ under a scaling of γ (we are talking about the third type of scale invariance illustrated in figure 7) is the same as the other terms. This means that the energy as a whole scales by a constant factor when the boundary is scaled. If $\alpha \neq 4$, the situation is interesting. Now a change in λ_D in equation V.10 is equivalent to changing the scale of the boundary, since E_D does not scale in the same way as the other terms, and so we can investigate the scaling behaviour of these functionals in an elegant way. This choice of α also means that we have introduced a fixed length scale, thus violating the invariance we have demanded of previous functionals. This is only excusable if we know something about the relation between the image scale and the scene scale. Changing the discontinuity cost

under these conditions, we would expect different stable regimes in which the part structure is constant, separated by phase transitions in which the topology and detailed structure changes. This is exactly what happens. It may be possible to investigate these transitions analytically, perhaps find some universal behaviours, and study the relationship with the smoothing that takes place in reaction-diffusion space.

Area. The terms that we have discussed so far suffer from a trivial solution. If γ collapses to a line, then the resulting minimum energy will be zero. In order to disallow this trivial solution, we need a term to force the two components of Γ away from the diagonal. One way to do this is to use a measure of the area of the region enclosed by the boundary. Using orientation-dependent terms, the construction of such a term is a simple matter. As in chapter III, we can use Stokes' theorem to convert the area integral to a boundary integral. Of course we must express this as a functional on Γ, but this is also simple (and general) as follows.

Given a functional E_B on boundaries in $\mathcal{D}$, we can define a functional $\tilde{E}_{\mathcal{B}}$ on $\mathcal{G}$ by

$$\begin{aligned} \tilde{E}_{\mathcal{B}}(\Gamma) &= E_{\mathcal{B}}(\pi_1\Gamma) \pm E_{\mathcal{B}}(\pi_2\Gamma) \\ &= E_{\mathcal{B}}(\gamma\sigma_1) \pm E_{\mathcal{B}}(\gamma\sigma_2) \\ &= 2E_{\mathcal{B}}(\gamma) \end{aligned} \tag{V.7}$$

where the last step follows from invariance arguments modulo some subtleties concerning discontinuities. The necessity of using a plus or a minus sign depends on whether the energy $E_{\mathcal{B}}$ is orientation-dependent or not, since the σ have opposing orientations.

Unfortunately, if we restrict ourselves to orientation-independent energies we can no longer use this method. Fortunately there is another way, given the extra information provided by Γ. The map between the boundary and itself, due to its piecewise monotonicity, 'slices' the enclosed region into disjoint and exhaustive pieces. At a point p on the circle, it can be seen from elementary geometry that the area of the quadrilateral created by the images of p and $p + dp$ under Γ_1 and Γ_2 is given by

$$E_A(\Gamma) = \int_{S^1} *_\Gamma |\frac{1}{2}(A_+, g)_\times| \tag{V.8}$$

This is positive and orientation-independent as required. It is not the most convenient form for our purposes nevertheless. In order to penalise small areas this term must enter the energy negatively, which then raises the question of how we shall keep the overall energy positive. It is easier instead to introduce a repulsive potential on the space $\mathcal{D} \times \mathcal{D}$. Such a function can be integrated over Γ easily. In the demonstrations shown later, we use a potential of the

form

$$E_A(\Gamma) = \ll \Gamma^* \|\phi\|^{-1}, \Gamma^* \|\phi\|^{-1} \gg \tag{V.9}$$

where $\phi = \pi_2 - \pi_1$. This tends to push Γ away from the diagonal. The use of such a potential is not ideal. It is akin to adding a term to favour greater lengths in a bid to offset the extensive nature of the energy. It may be that the only principled way to go beyond this is in the use of targets.

3.3. *Incorporating the Image*

So far we have assumed the independence of the part structure from the image given the boundary, as we stated at the beginning of the chapter. It is interesting to consider the possibilities that arise however when we violate this assumption. Let $\langle x, y \rangle$ be a point in $\mathcal{D} \times \mathcal{D}$. One possibility is to take the pullback of a difference between two functions or one-forms involving the image function, and use this as one of the g_i in equation V.3. An example would be to take $g = \Gamma^*(i\pi_2 - i\pi_1)$, as we did in the multiple image case. The same could be done with the gradient $\mathbf{d}i$ of the image function. In this case we are comparing not just the co-circularity at the two points, but image data also.

Another very interesting possibility is to use some function of the line segment between the matched points. For example, the integral of the modulus of the image gradient would provide a measure of the homogeneity of the image between the points.

More interesting still would be to look for particular attributes of the region corresponding to particular parts, for example colour. We do not yet have the machinery in place for this however, since it involves a labelling of the part structure. As the reader may recall from the second chapter, this is the eventual goal. We will return to this point in chapter VI.

3.4. *Incorporating the Boundary*

We now have a functional $E_{\mathcal{P}}(\Gamma)$, given by

$$E_{\mathcal{P}}(\Gamma) = \lambda_C E_C(\Gamma) + \lambda_I E_I(\Gamma) + \lambda_D E_D(\Gamma) \, (+\lambda_A E_A(\Gamma)) \tag{V.10}$$

where the $\lambda \geq 0$. We interpret the normalized negative exponential of this energy as the probability $\Pr(P = p | \pi_{\mathcal{P}}(P) = \pi_{\mathcal{P}}(p))$ that the part structure of and region occupied by the projection of a human being in the image domain is $p = \Gamma$, given that the region itself is $\pi_{\mathcal{P}}(p) = \gamma$. We ignore for the moment the image-dependent terms discussed in section 3.3. This energy is a measure of the deviation of the map Γ from perfect reflection symmetry, either because it gives a little in matters of isometry or co-circularity, or because it is

symmetric but around several different axes. The area term is parenthesized because it will be included in this way for some algorithms only.

In order to take the next step, we must combine this probability with the probability of a particular region given the image function, $\Pr(\gamma|I = i)$ to give the conditional probability of a region and part structure Γ given the image, $\Pr(\Gamma|I = i)$. There are two ways to do this.

One is to add to equation V.10 a term such as the positive orientation-independent one suggested in the discussion after equation III.2. This energy can then be written in terms of Γ itself using equation V.7. Let $\phi = |\mathbf{d}i|^{-1}$. Then the term to add is

$$E_{\mathcal{B}} = \sum_{a=1}^{2} \ll \Gamma^*(\phi\pi_a), \Gamma^*(\phi\pi_a) \gg \tag{V.11}$$

The second way is more in keeping with the spirit of the last two chapters. This is to use $E_{\mathcal{P}}$ (minus the area term) as the denominator $D[\partial R]$ in equation III.3. This makes sense since it is always positive and we wish to favour small values. As the numerator $N[\partial R]$ we can then use exactly the same form as in equations IV.3 and IV.4. Pulling back one-forms from each image domain and forming a linear combination gives a one-form on $\mathcal{D} \times \mathcal{D}$ that can be integrated along Γ. (This is the same procedure as that shown in equation V.7.) The difference in this case is that the images of the two projected boundaries are the same, so that the contributions to the numerator from the projected boundaries are equal, leading to the factor of two in equation V.7. Since we are dealing with orientation-dependent functionals in the numerator, we can easily incorporate an area term in the same way that we did in chapter III.

4. Algorithmics

We thus have two possible forms of energy functional. One is extensive, orientation-independent and positive, formed by adding the boundary term to equation V.10. The second is intensive, orientation-dependent and of varying sign, formed by dividing an orientation-dependent boundary and area term by equation V.10. Both these approaches are important. They correspond to different ways of thinking of the map Γ, and consequently different algorithms are necessary to optimize them. We discuss their applicability in the next two subsections.

4.1. *Treatment as a Boundary*

Continuous Case. In this case we are on firm ground. We can use the second, intensive form of energy functional, expressing the energy as a ratio

of conventional boundary and area terms to the energy $E_{\mathcal{P}}$ defined in equation V.10. We can discretize the space $\mathcal{D} \times \mathcal{D}$ in the same way as we did in the last chapter. The minimum ratio weight cycle algorithm applies unaltered, and so no initialization is necessary.

An apparent problem concerns the nature of the constraints that we must place on the boundaries we consider: the two projections of Γ must share a common image. This is a non-local constraint, and it is not clear how to implement it. It seems though that direct enforcement of this constraint is not be necessary, because the symmetry of the energy functionals and the coupling of the two projected boundaries through μ means that the same boundary will be found in each projection.

Since there are no discontinuities, the part structures found using this model have only one part. This does not render then trivial however. The energy functional favours mirror-symmetric regions over those without such symmetry. It is therefore a concrete example of boundary shape being incorporated into an energy functional in addition to generic models of boundary and region properties such as image function gradients and textures. This is a novel ability.

Discontinuous Case. The problem with this approach occurs when we wish to expand the space of Γ that we consider to include those with the types of discontinuities discussed in section 2.2. Consider the discretization of the problem. The graphs we will use are rectangular lattices embedded in $\mathcal{D} \times \mathcal{D}$. Edges connect pairs of pairs of points in $\mathcal{D}$. As long as any relations we are thinking of can be described in these terms the graphical structure is appropriate and the minimum ratio weight cycle algorithm will apply. Thus, for example, discontinuities in themselves are not a problem. Permitting them corresponds to adding edges to the graph connecting more widely separated points. The problem we have in mind however is a little more complicated. The discontinuities we are considering come in threes, meaning that triples of pairs of points in $\mathcal{D}$ are involved. As was mentioned in section 2.2, this takes us beyond the world of graphical structures and binary relations. Instead we must look at more general relations, or in other words *hypergraphs*.

What is needed then is a generalization of the idea of a cycle to the case of hypergraphs, and the construction of an algorithm that corresponds to the minimum ratio weight cycle algorithm. We do not describe such a generalization in this thesis. What then can be done? Fortunately there is another way to view the map Γ, not as a boundary but as a hyperpath in a hypergraph that loses none of the information in the map. This then suggests not a generalization of the minimum ratio weight cycle algorithm, but a generalization of a shortest path algorithm. In the previous work on boundary extraction that we discussed in section 2.1, the only methods that

could find global solutions were those based on dynamic programming. We pointed out that those models could be solved more efficiently using Dijkstra's algorithm. We thus concentrate on generalizing Dijkstra's algorithm to the case of 'hyperpaths' in hypergraphs. This generalization will suffer from the same problems as the original. It will require positive energies (which in our case implies orientation-independence), and hence initialization to avoid trivial solutions. These are unavoidable problems however in the absence of a hypercycle algorithm. Because of the demands of the algorithm, we will be forced to use the first, extensive form of energy, equation V.10 with the area term, added to equation V.11.

This does not mean abandoning completely the idea of using intensive energies. The arguments in favour of them over extensive energies still hold of course. Even if we were dealing with intensive energies though, the analysis of section 2.2 tells us that in the case of open paths we will still need to constrain our solution space to avoid trivial solutions, just as we would in the extensive case. Coupled with the fact that the conversion from a solution for the extensive case to one for the intensive depends heavily on an algorithm for the extensive case, this leads us to concentrate on the extensive case for the remainder of the chapter, leaving discussion of possible intensive algorithms to future work. We now show how Γ can be regarded as a path in the continuous case, and as a hyperpath when discontinuities are permitted.

4.2. *Treatment as a Path*

The fact that Γ_1 and Γ_2 must share a common image is a constraint. Another way of stating it is to say that the map $S^1 \stackrel{\Gamma/\sim}{\rightarrow} \mathcal{D} \times \mathcal{D}/\sim$ is not injective: it is not a subobject of $\mathcal{D} \times \mathcal{D}/\sim$. The image of this arrow however is a subobject by definition, and hence is constraint free. We discuss this first in the continuous case, and then generalize to the case in which discontinuities are allowed.

Continuous Case. Recall that the map μ has at least two fixed points, or alternatively Γ touches the diagonal in $\mathcal{D} \times \mathcal{D}$ in at least two places. These fixed points divide the circle into two segments. It is clear that μ maps one of these segments to the other. Since μ is self-inverse (Γ is a symmetric relation), this means that given the value of μ on one of these segments, we can deduce it for the whole circle. It can also be seen that the map from one of the segments into $\mathcal{D} \times \mathcal{D}/\sim$ will be an injection, with the two ends of the segment mapping to the images of the fixed points in the diagonal of $\mathcal{D} \times \mathcal{D}$, which is the boundary of $\mathcal{D} \times \mathcal{D}/\sim$. The upshot of this is that in the continuous case we can view Γ as a map of the interval Υ to $\mathcal{D} \times \mathcal{D}/\sim$, or in other words as a path. All the energy functionals we have defined can

be integrated along Υ as well as they can along the circle, and symmetry means that their value on the circle will simply be twice their value on the interval. Rather than search for a minimizing boundary then, we can search for a minimizing path. It is important to be clear about the meaning of this. The space over which our functionals are defined still represents a space of boundaries in $\mathcal{D}$ and their part structures. All that has happened is that the symmetry of our representation creates a redundancy; eliminating the redundancy allows us to represent boundaries in $\mathcal{D}$ and their part structures as paths rather than boundaries in the space $\mathcal{D} \times \mathcal{D}/\sim$. The two projections of such a path will be the two 'sides' of the boundary in $\mathcal{D}$, matched to each other by μ, and separated by the fixed points.

Since we are now dealing with paths, we must constrain our solution space in order to avoid trivial solutions. If we look at the paths between all pairs of points on the diagonal in $\mathcal{D} \times \mathcal{D}$ using our extensive orientation-independent energy, we will find a trivial solution, as predicted by section 2.2. Instead, in the absence of any other constraint, we must pick the boundary points on the path. Clearly this is not desirable. The true solution to this problem lies either in the development of a better approach to finding Γ as a boundary or in the use of *targets*. We return to this theme in chapter VI.

Discontinuous Case. If we permit the map μ to have discontinuities of the type we described in section 2.2, then the situation is similar to that in the continuous case, except that now the circle is first divided into a number of parts by the discontinuities. (This is of course the whole point.) Each of these parts will have at least one fixed point because μ is self-inverse and continuous on each part. Γ restricted to each of these parts can be viewed as a path in $\mathcal{D} \times \mathcal{D}$ from the image of the fixed point on the diagonal to the point of discontinuity. These paths are grouped together at three way junctions corresponding to the three discontinuities in each triple of discontinuities.

The structure we have described is rather like a tree, but this is a little misleading. The discontinuities cannot occur singly: they must occur in the manner we have described to guarantee the type of structures in which we are interested. The correct structure for describing this situation is that of a *hyperpath*, to be defined in the next section. Hyperpaths are to hypergraphs what paths are to ordinary graphs and they possess all the properties that we require. Just as in the last section, the use of hyperpaths means that we must constrain our solution space. We will do this by picking the fixed points. The only difference is that now there are more of them. Again the solution to this problem lies either in the development of a 'minimum ratio weight hypercycle' algorithm, or in the use of targets, to be discussed in the next chapter.

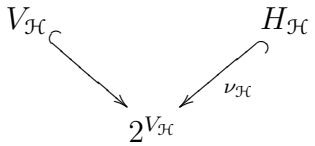
Following the same organizing principle used in chapter III, we now describe the discrete structures and algorithm that we will use, before describing their relation to the continuous formulation. In this case we begin by describing a category of hypergraphs to fix notation and define useful concepts.

5. HYPERGRAPHS

The category of undirected hypergraphs, **Hypergraph**, is defined in a manner analogous to the category of undirected graphs, **Graph**.

DEFINITION 5.1. An object $\mathcal{H}$ of the category is a set of *vertices*, denoted $V_{\mathcal{H}}$, a set of *hyperedges*, denoted $H_{\mathcal{H}}$, and an injection (the *edge map*) $H_{\mathcal{H}} \overset{\nu_{\mathcal{H}}}{\hookrightarrow} 2^{V_{\mathcal{H}}}$.[4]

(V.12)



REMARK 5.2. An element $h \in H_{\mathcal{H}}$ will be described as containing the vertices $v \in V_{\mathcal{H}}$ that are in $\nu_{\mathcal{H}}(h)$. This will often be notated $v \in h$ rather than the more exact $v \in \nu_{\mathcal{H}}(h)$.

REMARK 5.3. One can force every vertex to be contained in some edge, but this seems an unnecessary restriction that, furthermore, does not agree with the graph case, where vertices not in an edge are indeed possible.

DEFINITION 5.4. Composing $\nu_{\mathcal{H}}$ with the cardinality map from $2^{V_{\mathcal{H}}} \to \mathbb{N}$ gives the *valency* of $h \in H_{\mathcal{H}}$.

[4]Other possibilities suggest themselves. One is to define the vertex set $V_{\mathcal{H}}$ to be the set of singleton edges. It is then helpful to think of the vertices as a distinguished subset of $H_{\mathcal{H}}$ such that its image in $2^{V_{\mathcal{H}}}$ under $\nu_{\mathcal{H}}$ is the same as its canonical injection into $2^{V_{\mathcal{H}}}$. This ensures that each hyperedge is a subset of $V_{\mathcal{H}}$. This seems a more natural and attractive way to proceed, but it is not standard for some reason. Even more attractive is to view $H_{\mathcal{H}}$ itself as having certain structures (for example it is a poset), and then to use these structures to define the vertex set (the minimal elements of $V_{\mathcal{H}}$ for example). This of course converts $H_{\mathcal{H}}$ itself into a category.

DEFINITION 5.5. A morphism between hypergraphs $\mathcal{H} \xrightarrow{\mu} \mathcal{H}'$ consists of a pair of maps $V_{\mathcal{H}} \xrightarrow{\hat{\mu}} V_{\mathcal{H}'}$ and $H_{\mathcal{H}} \xrightarrow{\bar{\mu}} H_{\mathcal{H}'}$ such that:[5]

(V.13)
$$\begin{array}{ccc} H_{\mathcal{H}} & \xrightarrow{\bar{\mu}} & H' \\ {\scriptstyle \nu_{\mathcal{H}}}\downarrow & & \downarrow{\scriptstyle \nu_{\mathcal{H}'}} \\ 2^{V_{\mathcal{H}}} & \xrightarrow[\hat{\mu}_{\rightarrow}]{} & 2^{V_{\mathcal{H}'}} \end{array}$$

Note that given $V_{\mathcal{H}} \xrightarrow{\hat{\mu}} V_{\mathcal{H}'}$, if a suitable $\bar{\mu}$ exists it is unique, because $\nu_{\mathcal{H}'}$ is injective.

DEFINITION 5.6. There is an initial object, $\varnothing$, the hypergraph with empty vertex and hyperedge sets. There are terminal objects, 1, with $V_1 = \{\cdot\}$ and $H_1 = \{\{\cdot\}\}$.

DEFINITION 5.7. An *embedding* of one hypergraph in another is a monic arrow in **Hypergraph**. This is a morphism as above except that $\hat{\mu}$ is an injection. This means that $\hat{\mu}_{\rightarrow}$ is an injection and hence that $\bar{\mu}$ is an injection. The embeddings in a given hypergraph form a poset where, for embeddings $\mathcal{S} \xrightarrow{i} \mathcal{H}$ and $\mathcal{S} \xrightarrow{j} \mathcal{H}$, $i \preceq j$ if i factors through j, that is $\exists k : i = jk$. Two embeddings are equivalent if $i \preceq j$ and $j \preceq i$. A *subhypergraph* is an equivalence class of embeddings, although we will not labour the distinction between classes and representatives. We will sometimes refer to a subhypergraph by its morphism, and sometimes by the domain of the morphism. For $\mathcal{S} \xrightarrow{\pi} \mathcal{H}$, we will write V_π for the image of $V_{\mathcal{S}}$.

DEFINITION 5.8. Given $U \hookrightarrow V_{\mathcal{H}}$ for some hypergraph $\mathcal{H}$, we can define the subhypergraph induced by U as the pullback of $\nu_{\mathcal{H}}$ along $2^U \hookrightarrow 2^{V_{\mathcal{H}}}$. The hyperedge set of this hypergraph consists of those hyperedges in $H_{\mathcal{H}}$ all of whose vertices lie in U.

We also define a hypergraph $\mathcal{H} - U$. There is a projection from $2^{V_{\mathcal{H}}}$ to $2^{U'}$, where $U' = V_{\mathcal{H}} \setminus U$, that sends each $E \subset V_{\mathcal{H}}$ to $E \setminus U \subset U'$. This is in fact an equivalence relation on $2^{V_{\mathcal{H}}}$, two subsets being equivalent if they differ by elements in U. This equivalence relation induces a similar relation on $H_{\mathcal{H}}$, creating a set $H_{\mathcal{H}'}$ and an injection into $2^{U'}$. The pair U',

[5]This definition corresponds to order-preserving maps between the hyperedge sets considered as posets. The definition differs from another in common use. This is that a morphism is a pair of maps $V_{\mathcal{H}} \xrightarrow{\hat{\mu}} V_{\mathcal{H}'}$ and $H_{\mathcal{H}} \xrightarrow{\bar{\mu}} H_{\mathcal{H}'}$, such that $\hat{\mu}_{\rightarrow}\nu_{\mathcal{H}}(h)$ factors through $\nu_{\mathcal{H}'}\bar{\mu}$, or in terms of elements, $\forall h \in H_{\mathcal{H}} : \hat{\mu}_{\rightarrow}\nu_{\mathcal{H}}(h) \subset \nu_{\mathcal{H}'}\bar{\mu}(h)$. This is not a very natural definition. It is not order-preserving for a start, and it means that **Graph** is no longer a full subcategory of **Hypergraph**. It seems to be used to allow a definition of 'path' in the context of a hypergraph. This also is somewhat forced, since hypergraphs are not linear structures. Paths are replaced here by the more natural concept of hyperpath.

$H_{\mathcal{H}'}$, and the induced map $H_{\mathcal{H}'} \overset{\nu_{\mathcal{H}'}}{\hookrightarrow} 2^{U'}$ define a hypergraph $\mathcal{H}'$. We define $\mathcal{H} - U = \mathcal{H}'$. This is the co-equalizer of two hypergraph morphisms.

DEFINITION 5.9. Given a hypergraph $\mathcal{H}$ and a hyperedge $h \in H_{\mathcal{H}}$, we denote the hypergraph produced by the removal of h by $\mathcal{H} - h$. Note that this does not change the vertex set.

DEFINITION 5.10. Given two subhypergraphs, $\mathcal{S} \overset{\pi}{\rightarrowtail} \mathcal{H}$ and $\mathcal{T} \overset{\pi'}{\rightarrowtail} \mathcal{H}$ of a hypergraph $\mathcal{H}$, we can canonically define two new subhypergraphs of $\mathcal{H}$, the *intersection* $\mathcal{S} \sqcap \mathcal{T} \overset{\pi \sqcap \pi'}{\rightarrowtail} \mathcal{H}$ and *union* $\mathcal{S} \sqcup \mathcal{T} \overset{\pi \sqcup \pi'}{\rightarrowtail} \mathcal{H}$ of $\mathcal{S}$ and $\mathcal{T}$. These are formed by taking the intersection and union respectively of the vertex and hyperedge sets of $\mathcal{S}$ and $\mathcal{T}$.

DEFINITION 5.11. A *connected* hypergraph $\mathcal{H}$ is one for which there do *not* exist subhypergraphs $\mathcal{S} \rightarrowtail \mathcal{H}$ and $\mathcal{T} \rightarrowtail \mathcal{H}$ such that $\mathcal{S} \sqcap \mathcal{T} = \varnothing$ and $\mathcal{S} \sqcup \mathcal{T} = \mathcal{H}$. If it does exist, such a pair of subhypergraphs will be called a *disconnection* of $\mathcal{H}$.

DEFINITION 5.12. An *n-connected* hypergraph is one that is disconnected by the removal of any n hyperedges. An unconnected graph is therefore zero-connected.

DEFINITION 5.13. A vertex v in a connected hypergraph $\mathcal{H}$ will be called a *pin* if $\mathcal{H} - \{v\}$ is not connected.

DEFINITION 5.14. The *degree* of a vertex $v \in V_{\mathcal{H}}$ is the cardinality of $\{h \in H_{\mathcal{H}} | v \in \nu_{\mathcal{H}}(h)\}$. Note that if the hyperedge set contains singletons, these contribute to the degree. Vertices of degree one are called *terminal*.

DEFINITION 5.15. A connected, one-connected hypergraph in which all the vertices except the terminals are pins, and in which each hyperedge contains at most one terminal vertex will be called a *hypertree*.

5.1. *Hyperpaths*

DEFINITION 5.16. A *hyperpath* is a hypertree in which all vertices have degree at most two.

REMARK 5.17. It is possible to remove all talk of pins, and the restriction on the number of terminal vertices in each hyperedge, by moving to *directed* hypergraphs. The situations we will discuss however are akin to the case of directed graphs that are symmetric. Such graphs can be viewed as undirected graphs; we choose to view the symmetric hypergraphs with which we deal in an analogous way. Because of the more complicated symmetry properties when edges with valency greater than two are present, this viewpoint requires the extra constraints mentioned above. Nevertheless, it is less complicated to

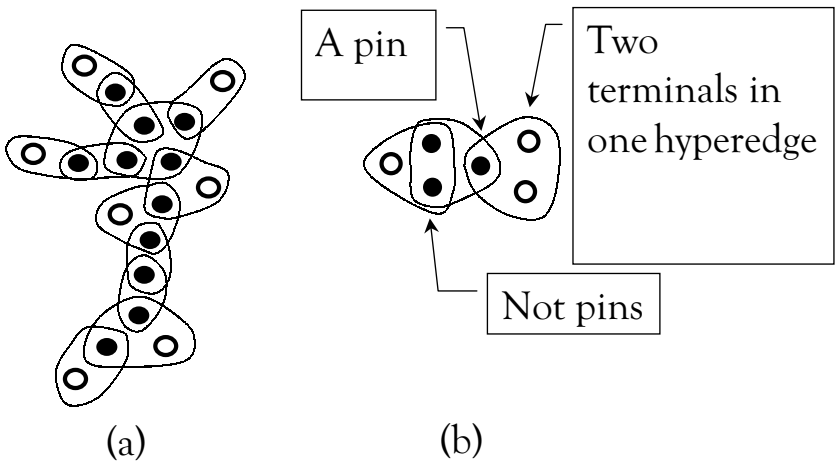


FIGURE 32. In part (a) a hyperpath is shown. The terminal vertices are not solid. In part (b), a hypergraph is shown. It is not a hyperpath because 1) the indicated vertices are not pins, and 2) one hyperedge contains two terminal vertices.

force hyperpaths to have the correct properties at this stage, even at the cost of a little inelegance, than deal with the consequences of introducing direction.

REMARK 5.18. Hyperpaths are the natural generalization of linear graphs. They are 'one-dimensional', this being the import of conditions on pin and terminal vertices. They are topologically trivial, this being the import of one-connectedness. Nevertheless they may still have a tree-like structure courtesy of the hyperedges with valency greater than two. Hypergraphs allow the 'local' interaction of more than two vertices at once, and this should be expressed in the linear structures. The normal concept of a 'chain' in a hypergraph is really just a graphical concept. Indeed there is a functor from **Hypergraph** to **Graph** that creates a bijection between chains in a hypergraph and paths in the corresponding graph (excepting certain trivial paths). This is not the case with hyperpaths, which are not so easily reduced to graphical notions. This functor F works as follows. For a hypergraph $\mathcal{H}$, the vertex set of $F(\mathcal{H})$ is $V_{\mathcal{H}} \cup H_{\mathcal{H}}$. There are no edges between vertices in $V_{\mathcal{H}}$ and none between vertices in $H_{\mathcal{H}}$. There is an edge between vertices $v \in V_{\mathcal{H}}$ and $h \in H_{\mathcal{H}}$ iff $v \in \nu_{\mathcal{H}}(h)$. The image $F(\mathcal{H}) \stackrel{F(\mu)}{\to} F(\mathcal{H}')$ of a morphism of hypergraphs $\mathcal{H} \stackrel{\mu}{\to} \mathcal{H}'$ has as vertex map the union of $\hat{\mu}$ and $\bar{\mu}$, and as edge map the derived map between the power sets. It can be seen that these maps obey equation V.13. Under this functor, hypertrees will be mapped to trees.

REMARK 5.19. Note that there are no vertices of degree zero in a hyperpath, since then the hypergraph would not be connected. Note that singleton edges do not occur in hyperpaths. If one occurred, either the vertex in it would have degree one, in which case it is easy to see that the hypergraph would not be connected (unless it were a terminal object), or it would have degree two, in which case removal of the singleton edge would not disconnect the hypergraph, meaning that it would not be one-connected.

LEMMA 5.20. *A connected subhypergraph $\mathcal{L}' \overset{\pi}{\rightarrowtail} \mathcal{L}$ of a hyperpath $\mathcal{L}$ is a hyperpath.*

PROOF. We must show that $\mathcal{L}'$ is connected, one-connected, and that all its vertices have degree less than or equal to two.

The first is given to us. The last is clear because if a vertex $v \in V_{\mathcal{L}'}$ was contained by more than two hyperedges, then the images of these hyperedges in $H_{\mathcal{L}}$ would contain the image of $v \in V_{\mathcal{L}}$. Since $\mathcal{L}'$ is a subhypergraph, the map between hyperedges is an injection, and thus the image of v would have degree greater than two, which is a contradiction.

For the second, we wish to show that $\mathcal{L}$ being one-connected implies that $\mathcal{L}'$ is one-connected. Assume the negation of this implication, that $\mathcal{L}$ is one-connected and that $\mathcal{L}'$ is not. Let 'conn' be the predicate that is true iff its argument is a connected hypergraph. Then $\neg\exists h : \text{conn}(\mathcal{L} - h) \wedge \exists g : \text{conn}(\mathcal{L}' - g)$. This entails $\exists g : \forall h : (\neg\text{conn}(\mathcal{L} - h) \wedge \text{conn}(\mathcal{L}' - g))$. In a moment we will show that $\neg\text{conn}(\mathcal{L} - \bar{\mu}(g)) \Rightarrow \neg\text{conn}(\mathcal{L}' - g)$, and hence that $\exists g : \bot$. This is a contradiction, proving that $\mathcal{L}'$ is one-connected.

To show that $\neg\text{conn}(\mathcal{L} - \bar{\mu}(g)) \Rightarrow \neg\text{conn}(\mathcal{L}' - g)$, note that $\neg\text{conn}(\mathcal{L} - \bar{\mu}(g))$ means that $\exists \mathcal{S}, \mathcal{T} : (\mathcal{S} \sqcup \mathcal{T} = \mathcal{L} - \bar{\mu}(g) \wedge \mathcal{S} \sqcap \mathcal{T} = \varnothing)$. The subhypergraphs $\mathcal{S}$ and $\mathcal{T}$ pullback along π, to give two subhypergraphs of $\mathcal{L}'$, $\mathcal{S}'$ and $\mathcal{T}'$. From the properties of the pullback, it follows that $\mathcal{S}' \sqcup \mathcal{T}' = \mathcal{L}' - g \wedge \mathcal{S}' \sqcap \mathcal{T}' = \varnothing$. □

REMARK 5.21. We will often deal with hypergraphs with distinguished vertices, and in particular with hyperpaths with distinguished terminal vertices. The former will be called *pointed*, and the latter *terminally pointed*. We will abbreviate 'terminally pointed' by 't.p.'. Morphisms between pointed hypergraphs must preserve the distinguished vertex.

DEFINITION 5.22. Given a hyperpath $\mathcal{L}$ and a vertex $v \in V_{\mathcal{L}}$, removal of a hyperedge h that contains v defines a t.p. subhyperpath $\mathcal{L}_{v,h} \overset{\iota_{v,h}}{\rightarrowtail} \mathcal{L}$ of $\mathcal{L}$ with distinguished vertex v as follows. $\iota_{v,h}$ is the maximal connected subhypergraph of $\mathcal{L} - h$ whose vertex set includes v. This is unique, because were there another such hypergraph, call it $\mathcal{S}'$, then $\mathcal{S}' \sqcup \mathcal{L}_{v,h}$ would be connected because any disconnection would either define a disconnection

for $\mathcal{S}'$ or $\mathcal{L}_{v,h}$, which are assumed connected, or would *be* $\mathcal{S}'$ and $\mathcal{L}_{v,h}$. The latter is not a disconnection however, because $\mathcal{S}' \sqcap \mathcal{L}_{v,h}$ has vertex set $\{v\}$. Since $\mathcal{S}' \sqcup \mathcal{L}_{v,h}$ contains $\mathcal{L}_{v,h}$, this contradicts the maximality assumption. The subhyperpath(s) produced in this way are called the subhyperpaths *induced* by v. For a terminal vertex, there is only one such subhyperpath, the hypergraph with v as a vertex and no hyperedges. It is convenient to introduce a second subhyperpath induced by a terminal vertex. This is defined to be $\mathcal{L}$ itself.

REMARK 5.23. For a hyperpath $\mathcal{L}$ and $v \in V_{\mathcal{L}}$, there are two subhyperpaths induced by v, $\mathcal{L}_1$ and $\mathcal{L}_2$. Then $\mathcal{L}_1 \sqcup \mathcal{L}_2 = \mathcal{L}$, and $\mathcal{L}_1 \sqcap \mathcal{L}_2$ is the hypergraph with one vertex (v), and no hyperedges.

DEFINITION 5.24. Given a t.p. hyperpath $\mathcal{L}$ with distinguished vertex t, we denote by $\mathcal{L}_v \overset{\iota_v}{\rightarrowtail} \mathcal{L}$ the unique t.p. subhyperpath induced by $v \in V_{\mathcal{L}}$ that does not contain t. Its distinguished vertex is v. It may be described as the subhyperpath *less than v relative to t*. If v is the distinguished vertex t, we define this subhyperpath to be $\mathcal{L}$ itself.

Since subhypergraphs of a hypergraph have a natural partial ordering, the map from $V_{\mathcal{L}}$ to the subhyperpaths less than a given vertex relative to the distinguished vertex induce a partial ordering on $V_{\mathcal{L}}$ itself. We denote this by $\lessdot$.

If T is the set of terminal vertices of $\mathcal{L}$ excluding the distinguished vertex, then the *colour*, $\kappa_{\mathcal{L}}(v) \subset T$, of a vertex v, is defined to be the subset of T contained in V_{ι_v}.

DEFINITION 5.25. We define the *predecessors* of a vertex v in a t.p. hyperpath $\mathcal{L}$, to be the set $\nu_{\mathcal{L}}(h_v) \setminus \{v\}$, where h_v is the unique hyperedge in V_{ι_v} containing v. These vertices are also the greatest lower bounds of v under the partial ordering $\lessdot$. This means that the subhyperpaths induced by the predecessors are disjoint (or one-connectedness would be violated) and exhaustive.

5.2. *Hyperpaths in a Hypergraph*

DEFINITION 5.26. The set of hyperpaths *in* a hypergraph $\mathcal{H}$, denoted $\Pi(\mathcal{H})$, is the set of subhypergraphs of $\mathcal{H}$ that are hyperpaths. We denote the space of subhyperpaths in a hypergraph whose terminal nodes map onto a given $T \subset V_{\mathcal{H}}$ by $\Pi_T(\mathcal{H})$, and those whose terminal nodes map to a subset of T by $\Pi_{\subseteq T}(\mathcal{H})$.

The set of t.p. hyperpaths in a pointed hypergraph $\mathcal{H}$ with distinguished vertex $t \in V_{\mathcal{H}}$ whose terminal nodes map onto $T \cup \{t\}$ for given $T \subset V_{\mathcal{H}}$ will be denoted by $\Pi_{T,t}(\mathcal{H})$. Those with terminal vertices a subset of $T \cup \{t\}$ will be denoted $\Pi_{\subseteq T,t}(\mathcal{H})$. Note that t must always be in the image of the

terminal vertices because of the definition of morphisms between pointed hypergraphs.

We will omit the hypergraph argument when the context is clear.

REMARK 5.27. Most of the definitions concerning hyperpaths in the previous section are easily extended to the case of hyperpaths in a hypergraph. The vertex set $V_{\mathcal{L}}$ is replaced by its image V_π.

DEFINITION 5.28. We are given a t.p. subhyperpath $\mathcal{L} \overset{\pi}{\rightarrowtail} \mathcal{G}$ in $\Pi_T(\mathcal{G})$ for a hypergraph $\mathcal{G}$. If necessary $\mathcal{G}$ can be regarded as pointed with distinguished vertex t the image of the distinguished vertex of $\mathcal{L}$. Given a vertex $v \in V_\pi$, with pre-image $u \in V_{\mathcal{L}}$, we define the subhyperpath of $\mathcal{G}$ less than v relative to π as $\pi\iota_u$, and denote it π_v.

The partial ordering on $V_{\mathcal{L}}$ induces a partial ordering on V_π. A vertex $v \in V_\pi$ is less than $u \in V_\pi$, $v \lessdot u$ iff π_v factors through π_u. We can define the predecessor relation on V_π by projection from that on $V_{\mathcal{L}}$ also. We denote the image of the unique hyperedge in $H_{\mathcal{L}}$ that contains the distinguished vertex by $h_\pi \in H_{\mathcal{G}}$. It contains t. We denote the set $\nu_{\mathcal{G}}(h_\pi) \setminus \{t\}$ by $P_{\pi,t}$. These are the *predecessors* of t relative to π.

We define the *colour*, $V_\pi \overset{\kappa_\pi}{\rightarrow} 2^T$ of $v \in V_\pi$ as the unique arrow that makes the following commute:

(V.14)
$$\begin{array}{ccc} V_{\mathcal{L}} & & \\ {\scriptstyle\hat{\pi}}\downarrow & \searrow^{\hat{\pi}\rightarrow\kappa_{\mathcal{L}}} & \\ V_\pi & \xrightarrow[\kappa_\pi]{} & 2^T \end{array}$$

DEFINITION 5.29. Using the above data, we define the *coloured* set of vertices in a t.p. hyperpath $\mathcal{L} \overset{\pi}{\rightarrow} \mathcal{G}$ in $\Pi_T(\mathcal{G})$, denoted by $K_{\pi,t} \subset V_{\mathcal{G}} \times 2^T$, as $K_{\pi,t} = \langle\hat{\pi}, \kappa_\pi\rangle_{\rightarrow}(V_{\mathcal{L}})$.

DEFINITION 5.30. A *weighted* hypergraph is a hypergraph $\mathcal{H}$ with a function $H_{\mathcal{H}} \overset{W}{\rightarrow} \mathbb{R}$. The weight of a subhypergraph $\mathcal{H}' \overset{\mu}{\rightarrowtail} \mathcal{H}$ is then defined as $\sum_{h\in H_{\mathcal{H}'}} W\bar{\mu}(h)$. The weight function on subhypergraphs is denoted by the same symbol as the weight of a hyperedge.

6. OPTIMAL HYPERPATH ALGORITHM

We look first at an algorithm, called algorithm A, for the following problem, surprisingly called problem A.

Input: A weighted hypergraph $\mathcal{G}$, hyperedge weight function W, and a set of vertices $T \subset V_{\mathcal{G}}$.

Output: $W^* = \min_{p\in\Pi_{\subseteq T}(\mathcal{G})} W(p)$.

There are other problems that we could consider, but they seem to be hard to solve. We will discuss other possibilities at the end of this section.

The algorithm we will describe is a generalization of Dijkstra's shortest path algorithm to the hypergraph setting. If the hypergraph has only edges of valency two, and if the set T is of cardinality two, then the algorithm will find the shortest path between the vertices in T. Indeed, in this case the algorithm reduces to Dijkstra's shortest path algorithm in its bi-directional form. this provides an intuition as to the working of the algorithm. In the bi-directional shortest path algorithm, paths are explored from each of two source vertices in weight order. When two paths meet at a vertex, the exploration ceases and a search over the already explored paths finds the optimal path between the two points. In the hyperpath case the situation is similar. Paths are explored from each of the vertices in T in weight order. If a path ends on a vertex that belongs to a hyperedge, all of whose other vertices have been reached except for one, the explored paths can join together using the hyperedge, and then continue to explore from the unexplored vertex. Eventually two hyperpaths will meet at a vertex, and exploration ceases. A search procedure now reveals the minimum weight hyperpath. This is illustrated schematically in figure 33.

We now proceed with the formal development of the algorithm.

DEFINITION 6.1. The space of *coloured vertices* is $\mathcal{C} = V_{\mathcal{G}} \times 2^T$.

DEFINITION 6.2. The space of *seen* vertices is $\mathcal{S} \subset \mathcal{C}$. Initially, $\mathcal{S} = \emptyset$.

DEFINITION 6.3. The *heap* is denoted $\Omega \subset \mathcal{C}$. Initially, it is equal to $\mathcal{C}$.

DEFINITION 6.4. The projections from $\mathcal{C}$ are denoted by $p_i, i \in \{1, 2\}$. We will denote $p_i(v)$ by v_i for $v \in \mathcal{C}$.

The central object of the algorithm will be a partial injective map π from $\mathcal{C}$ to the space of t.p. subhyperpaths of $\mathcal{G}$. To be more exact, the co-domain of the map will be $\cup_{v \in V_{\mathcal{G}}} \Pi_{\subseteq T, v}$. The initial value of π is as follows:

$$\pi(v) = \begin{cases} \langle v_1, \emptyset \rangle & v_1 \in T \,\&\, v_2 = \{v_1\}, \\ \uparrow & \text{otherwise.} \end{cases} \tag{V.15}$$

where $\uparrow$ means undefined ($\downarrow$ will mean defined).

The algorithm will iterate. The first action of each iteration will be to remove from Ω that v with the smallest value of $W\pi$. This vertex will be added to $\mathcal{S}$. This will be the only vertex moved, so that $\mathcal{S} \cup \Omega = \mathcal{C}$ always. The algorithm will iterate until a vertex v is removed from Ω for which there exists a vertex $u \in \mathcal{S}$ with $v_1 = u_1$ and $v_2 \cap u_2 = \emptyset$, or until $\forall v \in \Omega : \pi(v) \uparrow$.

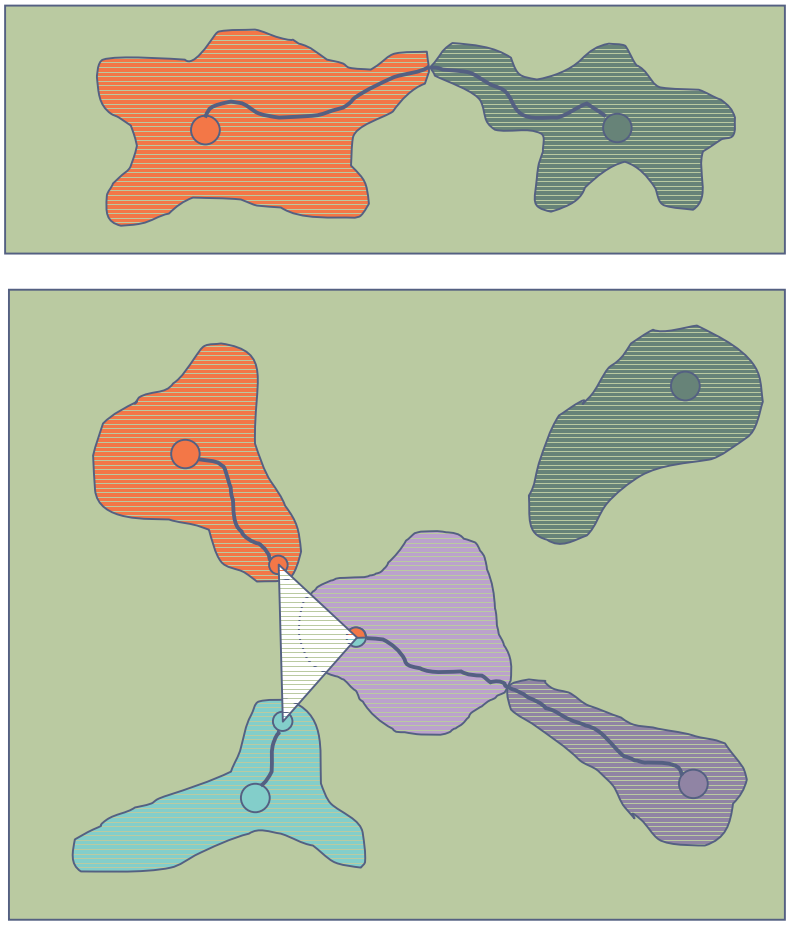


FIGURE 33. The top rectangle shows the bi-directional version of the shortest path algorithm in action. Paths have been expanded from each vertex in weight order. When two such paths meet at a vertex, as shown, the iteration ceases and a search procedure reveals the optimal path. Similarly, in the lower part of the figure, paths are shown expanding from the terminal vertices in the optimal hyperpath algorithm. The difference is that when two or more (depending on the hyperedge valencies) paths converge on a hyperedge, as shown by the pale triangle, they can join and expansion can continue from the unexplored vertex in the hyperedge. Again, when two paths meet at a vertex, expansion ceases and a search procedure finds the optimal hyperpath.

The algorithm will be such that the following statements are true at the beginning of each iteration:

$$\forall v \in \mathcal{S} : \pi(v) \downarrow \tag{V.16}$$

$$\forall v \in \mathcal{S} : W\pi(v) = \min_{p \in \Pi_v} W(p) \tag{V.17}$$

$$\forall v \in \Omega : \pi(v) \downarrow \Rightarrow \pi(v) \in \check{\Pi}_{\mathcal{S},v} \tag{V.18}$$

$$\forall v \in \Omega : \pi(v) \downarrow \Rightarrow W\pi(v) = \min_{p \in \check{\Pi}_{\mathcal{S},v}} W(p) \tag{V.19}$$

where $\Pi_v = \Pi_{v_2,v_1}(\mathcal{G})$, and $\check{\Pi}_{\mathcal{S},v} = \{p \in \Pi_v | K_{p,v_1} \setminus \{v\} \subset \mathcal{S}\}$. We now analyse these invariants and show how they lead to an algorithm.

PROPOSITION 6.5. *If equations V.17 and V.19 hold at the beginning of one iteration, then equation V.17 holds at the beginning of the next.*

PROOF. At the beginning of the first iteration, equations V.17 and V.19 hold by the definition of π.

At the beginning of an iteration, consider the vertex $k \in \Omega$ with the least value of $W\pi$. (If there is no such vertex because π is undefined for every vertex in Ω, then the iteration ceases.) We know from equation V.19 that

$$W\pi(k) = \min_{p \in \check{\Pi}_{\mathcal{S},k}} W(p).$$

If we can show that

$$W\pi(k) = \min_{p \in \Pi_k \setminus \check{\Pi}_{\mathcal{S},k}} W(p),$$

then by conjunction we will have shown that

$$W\pi(k) = \min_{p \in \Pi_k} W(p).$$

Since k is added to $\mathcal{S}$, and equation V.17 already holds for all $v \in \mathcal{S}$, it will be true for all $v \in \mathcal{S} \cup \{k\}$ if we do not alter the value of π for vertices in $\mathcal{S}$ after this point, which we do not.

To show that $W\pi(k) = \min_{p \in \Pi_k \setminus \check{\Pi}_{\mathcal{S},k}} W(p)$, consider an arbitrary hyperpath $p \in \Pi_k \setminus \check{\Pi}_{\mathcal{S},k}$. We claim that $\exists u \in K_{p,k_1} \setminus \{k\}$ such that the following hold:

$$u \notin \mathcal{S} \tag{V.20}$$

$$p_{u_1} \in \check{\Pi}_{\mathcal{S},u} \tag{V.21}$$

If this is true, then we have that $W(p) \geq W(p_{u_1}) \geq W\pi(u) \geq W\pi(k)$. The first inequality is because $p_{u_1} \preceq p$. The second is because of equation V.19. The third is true by assumption about the minimality of $W\pi(k)$.

To show that such a u exists, consider a minimal vertex $w \in V_p$ such that $\langle w, \kappa_p(w) \rangle \notin \mathcal{S}$. (Minimal here means with respect to the ordering $\lessdot$.) We

claim that $p_w \in \check{\Pi}_{\mathcal{S},\langle w,\kappa_p(w)\rangle}$, and that therefore we can take $u = \langle w, \kappa_p(w)\rangle$. If this were not the case, then there would exist a vertex $x \in K_{p_w,w} \setminus (\mathcal{S} \cup \{w\})$. It follows that $x_1 \lessdot w$ as vertices in V_p, and this contradicts the minimality of w. □

PROPOSITION 6.6. *If equations V.17 and V.19 hold at the beginning of one iteration, then equation V.19 holds at the beginning of the next.*

PROOF. We know that after the removal of k from Ω and its placing in $\mathcal{S}$, we have that equation V.17 holds for all $v \in \mathcal{S} \cup \{k\}$. (We will denote the latter set by $\mathcal{S}'$.)

However at this point, as it stands, equation V.19 does not necessarily hold. For each $u \in \Omega$, the invariant must now hold over the space $\check{\Pi}_{\mathcal{S}',u}$, where clearly $\check{\Pi}_{\mathcal{S},u} \subset \check{\Pi}_{\mathcal{S}',u}$. Therefore the values of π for the vertices in Ω may have to be updated to maintain equation V.19. In this proof we show which vertices need to be updated, and how this should be done. We will denote the space $\check{\Pi}_{\mathcal{S}',u} \setminus \check{\Pi}_{\mathcal{S},u}$ by $\check{\Pi}_{\mathcal{S}',\mathcal{S},u}$.

Consider $u \in \Omega$. Let

$$\phi_{\mathcal{S},k}(u) = \arg \min_{p \in \check{\Pi}_{\mathcal{S}',\mathcal{S},u}} W(p).$$

Then if we change the value of $\pi(u)$ to $\arg\min_{p \in \{\pi(u),\phi_{\mathcal{S},k}(u)\}} W(p)$, we have by conjunction that $W\pi(u) = \min_{p \in \check{\Pi}_{\mathcal{S}',u}} W(p)$, and thus that equation V.19 remains valid.

We thus need to find $\phi_{S,k}(u)$. We partition the hyperpaths in $\check{\Pi}_{\mathcal{S}',\mathcal{S},u}$ in the following way. As described above, each $p \in \check{\Pi}_{\mathcal{S}',\mathcal{S},u}$ defines a hyperedge $h_p \in H_{\mathcal{G}}$, the unique hyperedge in p containing u_1. This defines an equivalence relation on Π_u: p is equivalent to p' iff $h_p = h_{p'}$. Also as above, we can define the predecessors of u_1 relative to p, P_{p,u_1}. First we show that those hyperpaths p for which $k_1 \notin P_{p,u_1}$ can be eliminated.

Take an arbitrary hyperpath $p \in \check{\Pi}_{\mathcal{S}',\mathcal{S},u}$ in the latter class. If we can show that there is a hyperpath $p' \in \check{\Pi}_{\mathcal{S},u}$ such that $W(p) \geq W(p')$, then because $W(p') \geq W\pi(u)$ by equation V.19, we will have shown that $\forall p \in \check{\Pi}_{\mathcal{S}',\mathcal{S},u}$: $W(p) \geq W\pi(u)$, and thus that these hyperpaths need not be considered.

To construct such a hyperpath consider the set P_{p,u_1}. There is a unique $v \in P_{p,u_1}$ such that $k_1 \in V_v$. This follows from the exhaustiveness and exclusiveness of the subhyperpaths induced by the elements of P_{p,u_1}. This vertex induces a subhyperpath $p_v \in \Pi_{\kappa_p(v),v}$. From v we can construct a vertex $\langle v, \kappa_p(v)\rangle \in \mathcal{S}$ (because $p \in \check{\Pi}_{\mathcal{S}',\mathcal{S},u}$), and hence a hyperpath $\pi(\langle v, \kappa_p(v)\rangle) \in \Pi_{\kappa_p(v),v}$. The latter path is distinct from p_v because its vertex set does not contain k, since it was established in an iteration prior to k being added to $\mathcal{S}$. In addition, by equation V.17, the path $\pi(\langle v, \kappa_p(v)\rangle)$ has the minimum

weight of all hyperpaths in $\Pi_{\kappa_p(v),v}$ since $\langle v, \kappa_p(v)\rangle \in \mathcal{S}$, and hence has a lower weight than p_v. Replacement of p_v by $\pi(\langle v, \kappa_p(v)\rangle)$ in p therefore yields a hyperpath p' in $\check{\Pi}_{\mathcal{S},u}$ with $W(p') \leq W(p)$. Note that $\pi(\langle v, \kappa_p(v)\rangle)$ is disjoint from the hyperpaths induced by the other predecessors, for if it were not, iteration would have ceased earlier when one of the vertices in the intersection hyperpath was removed from Ω.

This leaves the hyperpaths $p \in \check{\Pi}_{\mathcal{S}',\mathcal{S},u}$ for which $k_1 \in h_p$ and for which $\kappa_{p,k_1} = k_2$. We call this space $\check{\Pi}_{\mathcal{S}',\mathcal{S},u,k}$.

We define further equivalence classes finer than those based on the h_p. Two hyperpaths p and p' will be regarded as equivalent under this second relation if $h_p = h_{p'}$ and $\forall w \in P_{p,u_1} : \kappa_p(w) = \kappa_{p'}(w)$. (Note that $P_{p,u_1} = P_{p',u_1}$ by assumption.) These equivalence classes partition Π_u and hence, by intersection, $\check{\Pi}_{\mathcal{S}',\mathcal{S},u,k}$. Let us call the set of these equivalence classes C. Not all the $c \in C$ will have a non-empty intersection with $\check{\Pi}_{\mathcal{S}',\mathcal{S},u,k}$. Necessary and sufficient conditions for the intersection to be non-empty for an equivalence class $c \in C$ are that the set $\{\langle w, \kappa_p(w)\rangle : w \in P_{p,u_1}\} \subset \mathcal{S}$ and that $\exists w \in P_{p,u_1} : \langle w, \kappa_p(w)\rangle = k$. These conditions are clearly necessary from the definition of $\check{\Pi}_{\mathcal{S}',\mathcal{S},u,k}$. That they are sufficient can be shown by constructing a hyperpath in $c \cap \check{\Pi}_{\mathcal{S}',\mathcal{S},u,k}$. This is done as follows. We have the hyperpaths $\{\pi(\langle w, \kappa_p(w)\rangle) : w \in P_{p,u_1}\}$. These do not depend on the representative p, and hence are determined by the equivalence class c. Each of these is in the corresponding $\Pi_{\kappa_p(w),w}$. These hyperpaths are disjoint, for if they were not, the subhyperpaths induced by a vertex in the intersection subhyperpath would have terminated the iteration before the current one. Therefore the hyperpath in $\check{\Pi}_{\mathcal{S}',\mathcal{S},u,k}$ formed by the $\{\pi(\langle w, \kappa_p(w)\rangle)\}$ (which includes by definition π_k), the hyperedge h_p and the vertex u_1 is a member of $c \cap \check{\Pi}_{\mathcal{S}',\mathcal{S},u,k}$, which is therefore non-empty. What is more, this member of $c \cap \check{\Pi}_{\mathcal{S}',\mathcal{S},u,k}$ is the one with the minimal weight, since each $\pi(\langle w, \kappa_p(w)\rangle)$ is the minimal hyperpath in $\Pi_{\kappa_p(w),w}$. We will call this hyperpath π_c. We then have that $\Phi_{\mathcal{S},k}(u) = \arg\min_{c \in C | c \cap \check{\Pi}_{\mathcal{S}',\mathcal{S},u,k} \neq \emptyset} W(\pi_c)$.

Therefore, consider the sets of vertices $Z \subset \mathcal{S} \setminus \{k\}$ that satisfy the following restrictions:

$$\exists h \in H_{\mathcal{G}} : \nu_{\mathcal{G}}(h) = \{z_1 : z \in Z\} \cup \{k_1 \cup \{u_1\}\} \tag{V.22}$$

$$\forall z \in Z : z_2 \cap k_2 = \emptyset \tag{V.23}$$

$$\forall z, z' \in Z : z_2 \cap z'_2 = \emptyset \tag{V.24}$$

$$\bigcup_{z \in Z} z_2 \cup k_2 = u_2 \tag{V.25}$$

These are sets of vertices that are colour-disjoint, and colour-disjoint from k. Together with k, they can be viewed as the in-vertices of the hyperedge h

that contains them (and so they would be if we were dealing with directed hypergraphs). The colours of these vertices including k must combine to produce the colour of u.

Clearly each such set Z defines an equivalence class $c \in C$ by h and the colours of the z_i and k. The union of the hyperpaths $\pi(z)$, $\pi(k)$, the hyperedge h and the vertex u_1 is the path π_c defined in the last paragraph. The weight of this path is therefore minimal over all paths in $c \cap \check{\Pi}_{\mathcal{S}',\mathcal{S},u,k}$ as discussed. Similarly each equivalence class in $\check{\Pi}_{\mathcal{S}',\mathcal{S},u,k}$ defines a set of vertices $Z \subset \mathcal{S}$ satisfying the above conditions. Rather than examine the classes in C directly then, we can instead examine the sets Z. For each such set we can compare $W\pi(u)$ with $\sum_{z\in Z} W\pi(z) + W\pi(k) + W(h)$, and change the value of $\pi(u)$ to π_c if the latter is smaller than the former. More efficiently, we can look at all hyperedges containing k_1, and find a set Z satisfying the above restrictions given h, where now u_1 and u_2 are *defined* by equations V.22 and V.25. This ensures that we do not needlessly examine vertices u for which no such set Z exists.

By following this search procedure for all such sets Z for each hyperedge containing k_1, we are guaranteed by the above arguments to set $\pi(u)$ to the $\arg\min_{p\in\check{\Pi}_{\mathcal{S}',u}}$ as required. □

Having proved that equations V.17 and V.19 hold, we now go on to discuss the final stage of the algorithm and show that it indeed will work as advertised.

Define a set $Z \subset \mathcal{S}$ as *completable* iff it satisfies the following conditions:

$$\exists h \in H_{\mathcal{G}} : \nu_{\mathcal{G}}(h) = \{z_1 : z \in Z\} \tag{V.26}$$

$$\forall z, z' \in Z : z_2 \cap z'_2 = \emptyset \tag{V.27}$$

and call h the *completion* of Z. Let the two elements of $\mathcal{S}$ that terminate the iteration be u and v. Note that the subset of $\mathcal{S}$ defined by u and $\left\{\langle w, \kappa_{\pi(v),v_1}(w)\rangle : w \in P_{\pi(v),v_1}\right\}$ form a completable set, the completion being the hyperedge $h_{\pi(v)}$.

For each completable set, we can define a hyperpath in $\Pi_{\subseteq T}(\mathcal{G})$ by taking the union of the $\pi(z) : z \in Z$ and adjoining the hyperedge h. For a completable set Z, call this hyperpath π_Z. Note that this is the minimum weight hyperpath over those hyperpaths in the equivalence class defined by the set Z and the hyperedge h. (This is entirely analogous to the argument concerning the set Z defined in equations V.22 and V.25.) We claim that when the iteration has finished

$$\arg\min_{Z\in\mathcal{Z}} W(\pi_Z) = \arg\min_{p\in\Pi_{\subseteq T}(\mathcal{G})} W(p) \tag{V.28}$$

where $\mathcal{Z}$ is the set of completable sets of vertices. If this is the case, we need only search the set of hyperedges to find the solution to the problem as stated at the beginning of the section. For each h there will be zero or more sets $Z \subset \mathcal{S}$ completed by h. We take the minimum weight π_Z over these sets, and then the minimum of these minima over all $h \in H_{\mathcal{G}}$ to obtain the solution.

PROOF. To see that equation V.28 holds, consider a hyperpath $p \in \Pi_{\subseteq T}(\mathcal{G})$. By removing a hyperedge h containing vertices $\nu_{\mathcal{G}}(h) \subset V_{\mathcal{G}}$ from this hyperpath, we define a set of disjoint hyperpaths $\{p_v : v \in \nu_{\mathcal{G}}(h\}$. Each hyperpath p_v lies in a different set $\Pi_{\kappa_{p_v}(v),v}$. We divide the set $\Pi_{\subseteq T}(\mathcal{G})$ into those hyperpaths for which there exists a hyperedge h such that

$$\bigcup_{v \in \nu_{\mathcal{G}}(h)} K_{p_v,v} \subset \mathcal{S}$$

(we will call such hyper paths *seen*), and those for which there does not exist such an h. The former case clearly defines a completable set $Z = \{\langle v, \kappa_{p_v}(v)\rangle : v \in \nu_{\mathcal{G}}(h)\}$ such that $\pi_Z = p$, while in the latter case no completable set Z exists such that $\pi_Z = p$. The hyperpath π_Z is therefore seen, and it is easy to see that it is the minimum weight seen hyperpath in the equivalence class defined by Z.

There is therefore an onto map from $\mathcal{Z}$ to the set of equivalence classes of seen hyperpaths in $\Pi_{\subseteq T}(\mathcal{G})$. If we can show that every hyperpath that is not seen necessarily has a larger weight than some seen hyperpath, then it follows that

$$\arg \min_{p \in \Pi_{\subseteq T}(\mathcal{G}) | \text{seen}(p)} W(p) = \arg \min_{p \in \Pi_{\subseteq T}(\mathcal{G})} W(p) \tag{V.29}$$

where the predicate 'seen' is self-explanatory. Then, since π_Z is the minimum weight hyperpath in each equivalence class, and the map to equivalence classes is onto, equation V.28 follows.

To this end, consider a subhyperpath p that is not seen. Call the two subhyperpaths of $\mathcal{G}$ induced by a vertex $z \in V_p$, $p_{z,1}$ and $p_{z,2}$. We claim there exists at least one vertex z in the vertex set of p such that $K_{p_{z,1},z} \not\subset \mathcal{S}$ and $K_{p_{z,2},z} \not\subset \mathcal{S}$. Note that by proposition 5.23, $p = p_{z,1} \sqcup p_{z,2}$, and $p_{z,1} \sqcap p_{z,2} = \langle \{v\}, \emptyset \rangle$, so that $W(p) = W(p_{z,1}) + W(p_{z,2})$. Consider $W(p_{z,1})$. By a similar minimality argument to that used in the proof of the invariance of equation V.17, there must be a subhyperpath of $p_{z,1}$, q, in $\check{\Pi}_{\mathcal{S},x}$ for some $x \in \Omega$. By equation V.19 this hyperpath must have weight $W(q) \geq W\pi(x)$. In turn, $W\pi(x) \geq W\pi(u)$ (and similarly $W\pi(x) \geq W\pi(v)$), where u and v are the vertices in $\mathcal{S}$ that stopped the iteration, because vertices are removed from Ω in $W\pi$ order. The same argument applies to $p_{z,2}$, meaning that $W(p_{z,1}) + W(p_{z,2}) \geq W\pi(u) + W\pi(v)$. Hence every not seen hyperpath

has a weight greater than the seen hyperpath formed from u and v, and hence equation V.29 holds. The only remaining step is to show the existence of a vertex z with the desired properties.

If no such vertex exists, then for every vertex z, we must have that $K_{p_{z,1},z} \subset \mathcal{S}$ and $K_{p_{z,2},z} \not\subset \mathcal{S}$ without loss of generality, since p is not seen. Given such a vertex z, consider the hyperpath $p_{z,2}$ for which $K_{p_{z,2},z} \not\subset \mathcal{S}$. We will refer to it as q for brevity. Each vertex in the set of predecessors, $P_{q,z}$ induces a subhyperpath of q. If the coloured vertices for all such subhyperpaths are subsets of $\mathcal{S}$, then the hyperedge h_q that contains z would then be a completion for the predecessor subhyperpaths and $p_{z,1}$. This means that p would be seen, which is a contradiction.

If on the other hand, one or more of the predecessor subhyperpaths of q does not have its coloured vertices a subset of $\mathcal{S}$, we proceed as follows. Call an arbitrary such vertex y. We know that the coloured vertices of $K_{q_y,y} \not\subset \mathcal{S}$ by assumption. Also, $K_{p_{y,2},y} \subset \mathcal{S}$, where $p_{y,2}$ is the subhyperpath of p induced by y that is not q_y, since otherwise we could choose y as our vertex for the proof, violating our assumption that no such vertex exists. Now however we are in the same situation as we were for z, but with y. Clearly this process can be repeated, but we can never consider the same vertex or hyperedge twice since this would violate one-connectedness. Eventually then we must consider a vertex y in a hyperedge h containing a terminal vertex t. There are now two possibilities. If h has valency two, we have the following argument. The coloured vertices of the subhyperpath q induced by y that does not contain t are, by induction using the above process, members of $\mathcal{S}$. On the other hand, the coloured vertex of the subhyperpath $\langle \{t\}, \emptyset \rangle$ is definitely a member of $\mathcal{S}$: it is the vertex $\langle t, \{t\} \rangle$. The hyperedge h is therefore a completion for q and $\langle \{t\}, \emptyset \rangle$, and therefore p is seen, which is a contradiction. If the hyperedge h has valency greater than two, either all the other vertices induce subhyperpaths whose coloured vertices are in $\mathcal{S}$, in which case again we have a contradiction, or one or more does not, in which case we continue the process. Since the hypergraph is finite, we must eventually reach a hyperedge h of valency two containing a terminal vertex (this proof can easily be adapted to show that every hyperpath must contain a number of such hyperedges greater than or equal to the maximum valency of the hyperedges that constitute it), and again we have a contradiction. □

7. PRACTICALITIES

7.1. *Other Possible Problems*

There are other problems we could consider, for example trying to find $\min_{p \in \Pi_T(\mathcal{G})} W(p)$. Call this latter problem B. Problem B seems hard. Removal of the stopping condition on the iterations of algorithm A would not help, since the disjointness condition that we used several times in the proof of correctness would no longer hold. An analogy with two graph problems also suggests that problem B is NP-hard.

Consider a weighted graph and a subset T of the vertices. Problem C is to find the least weight tree with leaves a subset of the vertices in T. Clearly the solution must be a path not just a tree, because given any such tree we can construct a number of paths of lower weight than the tree between vertices in T by picking a pair of leaves and taking the unique path in the tree between them. In fact, problem C can be solved using algorithm A for $\mathcal{G}$ a graph. Note that the triviality of the solutions to this problem in **Graph** is caused by the fact that a vertex of degree higher than two in a tree can, by removal of enough edges, be converted to a vertex of degree two without creating any more leaves. In **Hypergraph** this is not true. The 'vertices of higher degree' are in fact hyperedges of higher valency, and this valency cannot be changed. This means that problem A in **Hypergraph** may have solutions that are not simply paths, but hyperpaths.

Now consider the problem of trying to find the minimum weight tree with leaves *all* the vertices of T. Call this problem D. Problem D is NP-hard by a reduction of the Steiner tree problem. Given an instance of the Steiner tree problem, we can attach 'spurs' to each vertex in the Steiner set S. This means that for each $s \in S$, we create a new vertex s', and an edge of weight zero from s to s'. Now any tree with the $\{s' : s \in S\}$ as leaves defines a Steiner tree, and any Steiner tree defines a tree with the $\{s' : s \in S\}$ as leaves. The weights of these trees are the same since the new edges have zero weight. Therefore an algorithm for problem D would also solve the Steiner tree problem. The reduction is polynomial time and so problem D is NP-hard.

More explicitly, it can be seen that an analogous algorithm to algorithm A without the stopping condition could be constructed for problem D, but that it would fail for the same reason that algorithm A fails for problem B: the disjointness condition (which amounts to a greedy assumption) would be violated.

Unlike problems A and C, it is not the case that problem D is a graphical instance of problem B, nor does there seem to be a simple reduction of problem D to problem B. Nevertheless, the strong analogy between the

problems we are considering, and the fact that the same failure mode occurs for algorithms in the two cases, suggests that problem B is hard.

7.2. *Complexity*

In practice we do not store the map π explicitly. It is sufficient to keep, for each vertex, the weight of the corresponding hyperpath, and pointers to its predecessors in that hyperpath. We need to access the set $\mathcal{S}$ indexed by vertex. This can be done using a hash table. The same is true of the heap. Similarly, we need to access the hyperedge set by vertex. Each vertex could have pointers to the edges containing it for example. In our application, the hyperedges are defined using a predicate on the vertex labels, so that the other vertices in the hyperedge can be found easily.

The algorithmic complexity can be computed as follows. We define $V = \text{card}(V_{\mathcal{G}})$, $E = \text{card}(H_{\mathcal{G}})$, d is the maximum valency of the hyperedges in $\mathcal{G}$, and we use T to mean the cardinality of the set of terminal vertices.

The cardinality of $\mathcal{C}$ is $O(V.2^T)$. There may be this many iterations since on each iteration, we remove a single vertex k from the heap. This takes time $O(\log(V.2^T))$ using a binary heap. We then examine the hyperedges containing k. We may examine each hyperedge $O(d)$ times, once for each vertex in it except the last. Thus the work in the iteration stage is $O(V.2^T.\log(V.2^T) + d.E.W)$, where W is the amount of work we do on each examination of a hyperedge. For each examination we must check to see if there is an appropriate set of vertices in $\mathcal{S}$. If we index into $\mathcal{S}$ using the vertices in the hyperedge, we will find $O(d)$ sets of at most $O(2^T)$ vertices each, each set representing the coloured versions of a vertex in the hyperedge. If we check the possible combinations of these coloured vertices to see if they satisfy equations V.22 and V.25, we can spend at most 2^{dT} time. If we change the value of π on each of these occasions (or in other words perform a reduce key operation on Ω for every combination), we will spend $W = 2^{dT}.\log(V.2^T)$ time.

In the final stage of the algorithm, we examine each hyperedge again. The procedure is much the same as during the iteration stage. If we index into $\mathcal{S}$ using the vertices in the hyperedge, and then examine all combinations of the coloured versions of these vertices to see if they are completable by the hyperedge, we spend 2^{dT} time at most per hyperedge.

Putting everything together, we find that the time complexity of the algorithm is asymptotically $O(d.2^{dT}.E.\log(V.2^T))$. The algorithm is thus mercifully polynomial in the vertex set and the edge set. It is however exponential in the size of the set of terminal vertices and the maximum valency of the hyperedges. This is not surprising: 2^{dT} is somehow the natural measure since we are dealing with the power set of the vertex set. If $d = 2$, and

$T = 2$, this complexity becomes $O(E.\log(V))$ as expected from Dijkstra's algorithm.

7.3. *Application*

In order to apply the algorithm described above to the optimization problems in which we are interested, we must map the continuous problem onto the discrete. We do this as follows. We discretize the space $\mathcal{D} \times \mathcal{D}$ using a rectangular lattice, precisely as we did in the last chapter. This defines the vertex set and the edges of valency two in the hypergraph. Recall now that the discontinuities correspond to triples of points in $\mathcal{D} \times \mathcal{D}$. These triples always take the form $\langle x, y\rangle$, $\langle y, z\rangle$, $\langle z, x\rangle$ as discussed in section 2.2. These triples of points define the hyperedges of valency three of the hypergraph for the solution of the problem. The definition of hyperpath ensures that the parts of the hyperpath involving edges of valency two are just ordinary paths in the rectangular lattice. Using the hyperedges of valency three however, these paths can join in a trivalent tree structure exactly as required by the discussion in section 2.2.

The set of terminal points T required in the problem statement in section 5 is given by the user. A number of points in the image domain are chosen. Let us call these *source* vertices. For each source vertex s, there is a terminal vertex $\langle s, s\rangle \in \mathcal{D} \times \mathcal{D}$. Thus the terminal vertices are those vertices that match to themselves: they lie on a local axis of symmetry.

The weights for the hyperedges of valency two are given by integration of all terms in the energy functional except E_D along the hyperedge embedded in $\mathcal{D} \times \mathcal{D}$, again just as in the previous two chapters. The weights of the hyperedges of valency three are computed as the value of $(J, J)^{\alpha/2}$ as in section 3.2.

We have now defined an instance of problem A. Subject to the usual limits of discretization accuracy, the solution to this problem will be a solution to the continuous optimization problem in which we are interested. The complexity of the algorithm in terms of the image size of n pixels is as follows. The number of vertices $V = n^2$. The number of hyperedges of valency two is $k.n^2$ where k is a constant of the order of 10. The number of hyperedges of degree three is $a.n^3$, where a is another constant, this time of order 1. The maximal valency is $d = 3$. This boils down to a complexity of $O(8^T.n^3.\log(n))$. The space resources required are not as large as they might seem. Vertices can be created on the fly, so that the whole graph does not need to be kept in memory. Indeed, only a tiny fraction of the graph is explored in most cases because the algorithm, likes Dijkstra's algorithm, is as parsimonious with memory as it is possible to be. Nevertheless, the memory must be at least the number of iterations, since one vertex is removed from the heap on each

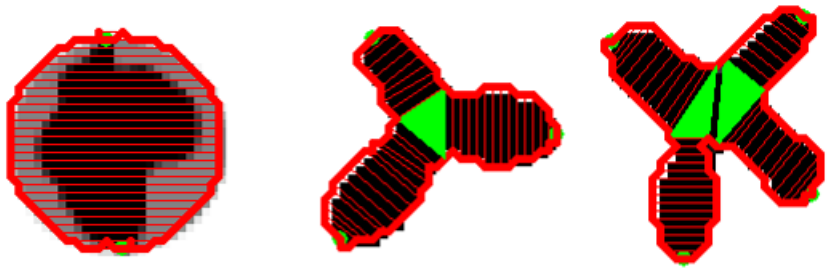


FIGURE 34. Three examples of optimal hypergraphs. In these cases the parameters were adjusted to achieve the number of parts shown. The terminal points were at the top and bottom of the circle and at the ends of the lobes. They are drawn as green dots, which are partly concealed by the red curves. The thinner red lines inside the red curves show the self-matching.

iteration. In practice this is a gross underestimate. For an image of n vertices, this quantity is $n^2.2^T$. For a 256×256 image, this is already 10^{10}. This means that with present resources, it is only possible to run the algorithm on small images (50×50).

Figures 34 to 37 demonstrate the running of the optimal hyperpath algorithm on some synthetic and real images. The different λ parameters were adjusted by hand in these images. The space of parameters is large and, especially on real images, an optimal setting for different images is not to be expected. One of the benefits of targets, to be discussed in the next chapter, is the stability they confer, so that results are not so parameter dependent. The use of the energy density form with a minimum ratio weight hypercycle algorithm would achieve a similar end in an even cleaner fashion since, as we saw in section 4.4, it effectively sets the parameters according to the image. The demonstrations show the disadvantages of an extensive energy. Apart from the need to initialise, skipping means that the optimal hyperpath may often have a topology a lot simpler than the number of terminal points.

8. POSTSCRIPT

This chapter has extended the work of [LG97, LGK98, Liu97] to real images, while also constructing a general theoretical framework using hypergraphs for the discretization and optimization of functionals on paths with a higher than two-point dependence on the points in the path. (One way to say this is that three-point correlation functions for the probability distributions involved no longer factor into products of two-point functions.) This new

FIGURE 35. A difficult real image. Two terminal points were used. They are shown as green dots.

FIGURE 36. The leaf image on the left shows how the extensive nature of the energy can force a simplification of the topology. In addition to the green dots in the image, there were four more terminal points at the tips of the leaf lobes. Skipping has allowed the model to find only a one-part region. The two right hand images show the effect of increasing the area term. The co-circularity energy itself has a tendency to make the region found slimmer. The balance between this term, the extensive nature of the energy and the area term is complicated.

ability is distinct from the fact that we have been examining boundaries in $\mathcal{D} \times \mathcal{D}$, which, as we saw in the last chapter and in section 4.1, need only involve graphical structures if the coupled triples of discontinuities are not permitted.[6] However, the use of path and hyperpath structures in $\mathcal{D} \times \mathcal{D}$ did

[6]There is almost certainly a beautiful algebraic version of the work described in the last few sections of this chapter that bears the same relation to hypergraphs and hyperpaths that 'path algebras' (or 'closed semirings', 'semilattice-ordered monoids' or any of a number of other terms for the same thing) bear to graphs and paths. The notion of colour introduced for the purposes of problem A extends this hypothetical unsorted theory to a sorted one, the colours

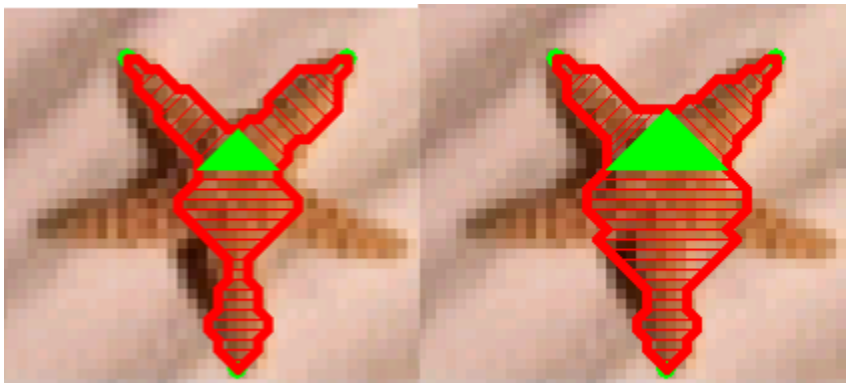


FIGURE 37. Again the effect of the area term can be seen, as can the effect of skipping. Note how the co-circularity term forces symmetry on the region even when the data does not support it. This is a form of completion, except now proceeding with a symmetric model rather than a generic one.

enable us to examine the space of boundaries and not just paths in $\mathcal{D}$ itself, and to incorporate new structural information about the shape and symmetry of these boundaries.

The disadvantages of using an extensive energy as we have for the greater part of this chapter are, as always, twofold. First, we have to constrain the space of solutions to avoid trivialities. In algorithmic terms, this means initialization by the user. Second, smaller length boundaries are preferred. This produces a tendency to favour small number of terminal vertices in the optimal part structure, and to 'skip' across sections of the image that do not strongly support a boundary themselves, but that are short and connect two sections that do strongly support a boundary. Some examples can be seen in the demonstrations.

This second difficulty can be addressed in two ways. One would be to solve problem B, but as we discussed in section 7.1, this is likely to be impossible to do efficiently. Preferable anyway to this approach would be to use an intensive energy. This would stop skipping because the average weight of such a short skip would be much greater than that of a longer but more strongly supported way round. The paper by Meggido [Meg79] shows how to convert an algorithm for an extensive energy to one for an analogous intensive one in rather general circumstances. We have not investigated whether this approach would work for algorithm A, but it seems likely.

---

acting as sorts. The same type of relation holds between categories and n-categories, which are similar to sorted hypergraphs. The relation of n-categories to logical theories raises the question of the meaning of hyperpaths in that context. They look rather like proofs.

The problem with this latter approach is that it does not remove the need for constraints on the space of solutions, as table 2.2 makes clear. When there are no tripled discontinuities, section 4.1 shows how to do this. When there are such discontinuities, the most natural way to remove this need in light of the work in the previous two chapters would be a generalization of the minimum ratio weight cycle algorithm to hypergraphs. This would involve an appropriate definition of hypercycle, as we discussed at the end of section 4.1. The definitions would seem to flow naturally from sections 5 and 5.1. The algorithm however is another matter. A further (and not exclusive) possibility is to implement soft constraints on the space of solutions using targets, as we are about to discuss.

We have completed two stages of the planned visual system, by constructing the probability distributions $\Pr(P = p|I = i)$. The final stage of the process is to incorporate the labelling of the parts, thus giving a probability distribution $\Pr(Q = q|I = i)$, the MAP estimate for which is the output of the visual system. This again will be accomplished using targets.

CHAPTER VI

TARGETS

A final layer is added to the model. It takes the form not of further representations, but of the choice of a target in the already existing representation space $\mathcal{P}$. A metric on this space then enables us to guide the solution towards the target, which stands for a class of possible shapes. The metric usually involves a correspondence between parts. This can be purely topological, it can be geometric, and it can even involve image data. It enables a labelling of the target part structure to be transferred to the object found. An optimization algorithm for the limited case of a fixed boundary is described.

So far we have described a model for regions in images in which two representations are used. The first is the boundary of the region, while the second is the shape of the region, as encoded in local symmetry information. As always, the representations serve two purposes. One is to narrow the probability distribution towards the statements we wish to make about the image. The other is to encode different types of information about the region. For example, the minima of the energy functionals we have so far constructed might be described as 'articulated objects'. These minima are not only localised by the functionals. For each minimum we also know the region and its part structure.

To make the model even more specific, we could introduce further representations and functionals based on these extended spaces. Instead we take a different route. We define a (potentially degenerate) metric on the space of representations that we are already considering, and then pick an exemplar from this space. We can now increase the specificity of the functional by adding the distance between any configuration and the exemplar to our energy. We will thus tend to find configurations closer to the exemplar. We will call such exemplars *targets*, and will denote them $\tau \in \mathcal{P}$.

Targets serve the same two purposes as the other structures we have introduced. Certainly they narrow the probability distribution with which we are dealing towards themselves, but they can also be used to encode a type of

information that would be hard to incorporate in another way. Among other things they can be used to restrict the size of the boundary, and the topology and geometry of the part structure. Targets can also be labelled; the parts need no longer be anonymous. If the metric depends on a correspondence between the part structures and boundaries of the two points in $\mathcal{P}$ concerned, then a labelling of the target is translated into a labelling of the boundary and its part structure, as we now explain.

1. The Full Space $\mathcal{Q}$

The full space with which we will deal is constructed as follows. Given $x, y \in \mathcal{P}$, we will define a space of morphisms between the structures represented by these points, $M(x, y)$. These morphisms will compose and there will be identities, so that $\mathcal{P}$ becomes a category. We will pick a target $\tau \in \mathcal{P}$. We then define the space $\mathcal{Q}$ as $\bigcup_{p \in \mathcal{P}} M(p, \tau)$. There is a projection, as we discussed, from $\mathcal{Q}$ to $\mathcal{P}$, that takes $\mu \in \mathcal{Q}$ to its domain $\mathrm{dom}\, \mu$.

Each point in $\mathcal{Q}$ thus encodes a great deal of information. Via the projections, first to $\mathcal{P}$ and then to $\mathcal{B}$, it encodes a boundary and a part structure for that boundary. In addition, it encodes a correspondence between this data, and the similar data encoded in the target. So, for example, as we will see, with an appropriate choice of space of morphisms, a point q in $\mathcal{Q}$ matches the part structure of $\pi_{\mathcal{P}}(q)$ to that of τ. This is precisely a labelling of the part structure $\pi_{\mathcal{P}}(q)$.

The reasons for extending $\mathcal{P}$ to the space of morphisms are the same as the reasons for extending the space of boundaries $\mathcal{B}$ to $\mathcal{P}$. We discussed in section 2 how minimizing over the part structure for a fixed boundary left us with an energy on the space of boundaries alone that was of lower entropy than $E_{\mathcal{B}}$, but that it was both useful and necessary to leave the part structure explicit. It was useful because then we had easy access to the extra information provided by that part structure, and it was necessary because the above mentioned optimization was impossible to compute ahead of time. The same is true of the space of morphisms $\mathcal{Q}$. We want to leave the matching between the structures of the target and the other points explicit so that we have access to that matching: for example the labelling of the parts. At the same time, it is also true that it is rather hard to define a metric on $\mathcal{P}$ without minimizing over some space of morphisms. We therefore now discuss such metrics.

2. Energy Functionals

We will define a positive energy on each set of morphisms, $E_{\mathcal{Q}}(\mu)$, where $\mu \in M(x, y)$. (Note that x and y are implicit in μ.) We define a positive

function on $\mathcal{P} \times \mathcal{P}$ by

$$d(x,y) = \min_{\mu \in M(x,y)} E_{\mathrm{Q}}(\mu). \tag{VI.1}$$

We will insist that the energy of the identity morphism is zero, so that $d(x,x) = 0$. The triangle inequality is satisfied if there is an injection of $M(x,y) \times M(y,z) \overset{\alpha}{\rightarrowtail} M(x,z)$ for which $E_{\mathrm{Q}}(\alpha\langle\mu,\nu\rangle) \leq E_{\mathrm{Q}}(\mu) + E_{\mathrm{Q}}(\nu)$ for all $\langle\mu,\nu\rangle \in M(x,y) \times M(y,z)$. This is a sufficient condition for d to be a metric. We will not insist or prove this property for any of the energies we consider since we are always dealing with the distance to a fixed point.

In many cases the sets of morphisms will be the same over subsets of $\mathcal{P} \times \mathcal{P}$, so that the distance on these subsets will also be constant. In this way, the space $\mathcal{P} \times \mathcal{P}$ will be divided into equivalence classes of equidistant points. Choosing a target then 'slices' these equivalence classes, and induces equivalence classes on $\mathcal{P}$ itself. In particular, there will be a subset of $\mathcal{P}$ whose elements are all distance zero from the target. This means that the distance of any other point to any element of this subset is a constant if the triangle equality is satisfied. We can therefore view the distance to the target as a distance to this subset of $\mathcal{P}$, and the combination of the target and the metric thus defines a class of shapes that we wish to find.

In the space $\mathcal{P}$ there is a particularly obvious example of where this type of behaviour might occur. For each map Γ, we can construct a *topology tree* as follows. We create a vertex for each triple of discontinuities, and a vertex for each fixed point. These vertices are joined by edges if there is a continuous path between them in the image of Γ. This is the same as first applying the functor F from **Hypergraph** to **Graph** to construct a tree, and then eliminating all degree two vertices by replacing them by edges between their neighbours. (In this chapter, we will use 'hyperpath' to mean both a member of the class of piecewise continuous maps of which Γ is a member, and the corresponding discrete structure.) We will call this map from $\mathcal{P}$ to the set of degree three trees, $\mathcal{T}$. $\mathcal{P}$ is thus divided into equivalence classes corresponding to trees of different topology. An easy type of metric to use is then one based solely on these isomorphism classes of trees. For example, we could use the edit distance between the trees. We will call such a metric a *topological* metric. We can then add to this metric one based on intra-class differences, or in other words on the geometry, since the sum of two metrics is also a metric. We will call this the *geometric* metric. In practice, we will always take the topological metric to be an inverse delta function. The distance between two points corresponding to the same topology will be zero. If the topologies are different then the distance will be infinite.

For the intra-class metric there are many possibilities. Even though the topologies are the same, there may be more than one isomorphism between the topology trees, corresponding to different matches between (and consequently different labellings of) the part structures. For each of these isomorphisms, the corresponding parts may be geometrically more or less similar. Recall that each part is a continuous path in $\mathcal{D} \times \mathcal{D}$. Let us call the space of paths in $\mathcal{D} \times \mathcal{D}$, Π. The simplest way (and also the algorithmically tractable way) to define an energy is thus first to define a distance between such paths and then add up the distances between the corresponding parts in the isomorphism.[1] To be more formal: assume we have $\Pi \times \Pi \xrightarrow{\varepsilon} \mathbb{R}^+$. Let $M(p, p')$ be the space of isomorphisms between $\mathcal{T}(p)$ and $\mathcal{T}(p')$ for $p, p' \in \mathcal{P}$. This set may be empty. For each point $p \in \mathcal{P}$ there is a map $H_{\mathcal{T}(p)} \xrightarrow{\rho_p} \Pi$ giving the part corresponding to an edge in $H_{\mathcal{T}(p)}$. Then, in the case that $M(p, p')$ is non-empty, we define the energy of $\mu \in M(p, p')$ by

$$E_{\mathcal{Q}}(\mu) = \sum_{e \in H_{\mathcal{T}(p)}} \varepsilon(\rho_p(e), \rho_{p'}(\mu(e))) \tag{VI.2}$$

The idea is illustrated in figure 38.

There are of course many possibilities for the distance ε. The simplest is a constant, thus creating a purely topological energy. A simple possibility is to use squared differences of the lengths of the paths or the areas enclosed by them, or fractional versions of the same. Another possibility is to first scale the paths so that they are the same length, and then take the area of the symmetric difference of the regions enclosed by each path. This allows parts to be stretched versions of one another. The most complicated possibility is to match the parts geometrically by a minimization over some space of maps. This however would add greatly to the time complexity of the algorithm, and moreover violates the spirit of comparing the part structures at an abstract level.

Having defined such an energy $E_{\mathcal{Q}}$, we can then complete our definition of equation II.3:

$$E(\mu, i) = E_{\mathcal{Q}}(\mu, i) + E_{\mathcal{P}}(\operatorname{dom}\mu, i) + E_{\mathcal{B}}(\pi_{\mathcal{P}}\operatorname{dom}\mu, i) \tag{VI.3}$$

where we have replaced q by $\operatorname{dom}\mu \xrightarrow{\mu} \tau$.

We have left the dependence on the image in $E_{\mathcal{Q}}$ for good reason. The distances between parts summarized in ρ may depend on the image data within those parts. What can including dependence on the image do for us at this stage? As well as associating labels with the parts of the target, which

[1] This is a linear energy in the same way as those in equations III.2. By taking products of the space of parts in the target, we can again define polynomial energies that depend on more than one part at once, and hence could be used to encode relative articulation for example.

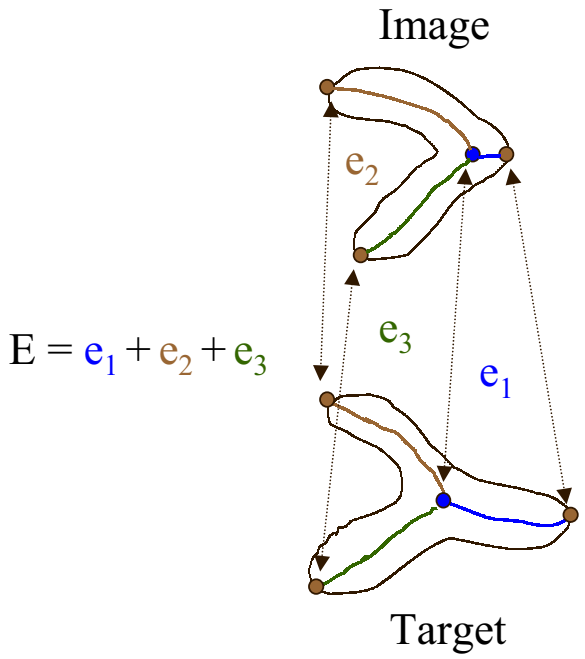


FIGURE 38. Given a tree isomorphism, the corresponding parts are matched individually and their energies summed.

are properties of the target alone and hence are simply transferred to the the part structure we are considering, we can attach other information beyond the geometric data inherent in the part structure and boundary. As a simple example, we could label the part with a colour. Given a corresponding part of a region in the image domain, we can then define ϵ based on a colour matching. In the case we have been considering, this seems particularly valuable, since colour is a useful cue for which part is the head. We can even, if we are considering a correspondence finer than the purely topological, attach objects such as eyes to the part, or colours for certain percentages of the length of a part. In short, we can build up a description of a human being in an image.

3. MICROCOSM: FIXING THE BOUNDARY

Having reached this stage, we must now consider the question of how we are to optimize $E(\mu, i)$ over the space $\mathcal{Q}$. In order to approach this task slowly, we will first consider a reduced task, that is of interest in its own right as a study of shape, as well as as the full problem in little.

The reduced problem is the minimization of E for a fixed boundary. In addition, we will effectively banish the image entirely by ignoring the possible dependence of $E_{\mathcal{Q}}$ and $E_{\mathcal{P}}$ on i. We are thus dealing with a purely geometric problem. It might be described like this: "if I view that shape as an instance of this target shape, what is its part structure and correspondence to my target shape?" This is by analogy to the full problem, which might be stated as: "if I view that image as being one of a scene containing an object that projects to this target shape, where does that object project to in the image domain, what is its part structure, and how does it correspond to my target shape?"

Since the boundary is fixed we can drop the term $E_{\mathcal{B}}$. The remaining terms have an interesting interpretation. Given two boundaries, b and b', we can pick one of them as the exemplar, let us say, b'. If we now compute the part structure of b' by minimizing the term $E_{\mathcal{P}}$, we can use the resulting point p' of $\mathcal{P}$ as a target. If we now minimize $E_{\mathcal{Q}}(\mu) + E_{\mathcal{P}}(p)$, where p is constrained to have $\pi_{\mathcal{P}}(p) = b$, and $\mu \in M(p, p')$, we end up with a number, the minimum energy. This number can be viewed as a distance between b and b', when b' is viewed as the exemplar. Repeating the procedure with b as the exemplar will not in general result in the same energy. The measure is thus asymmetric. This is the scenario suggested in section 1.3 to explain the asymmetry of similarity judgements in human (and pigeon) perception. Picking the exemplar and then computing its part structure without the aid of a target is saying "this shape defines a part structure on its own". Its part structure is one of its properties, just as being a 'round number' is one of the properties of 100. Using the resulting part structure as the target then forces the second boundary to be similar to the target in its part structure even if, without such a comparison, its part structure might be rather different. It is interpreted as an instance of the target. This discussion extends of course to the use of targets in the full problem, and is one of the most interesting aspects of their use.

Prior to incorporating the target though, it is useful to consider the simplification that can be made to algorithm A when we consider a fixed boundary.

3.1. *Computing the Parts of a Boundary*

If, instead of searching the full space of boundaries and part structures as we do in algorithm A, we restrict attention to the different part structures of a given fixed boundary in $\mathcal{D}$, we find that the notion of colour so important to algorithm A can be almost (but not quite) entirely dropped, and that we do not need to choose certain fixed terminal points. We then have an algorithm for the computation of self-matchings for boundary that is an improvement

over that of [Liu97, LG97, LGK98], and that is now derived from a much broader and richer framework.

In the continuous case, if we fix the boundary then we are no longer dealing with the full space $\mathcal{D} \times \mathcal{D}$. We are instead dealing with $S^1 \times S^1 \subset \mathcal{D} \times \mathcal{D}$, the torus produced by the product of the boundary in $\mathcal{D}$ with itself, with metric properties induced from $\mathcal{D}$. More accurately, we are dealing with $S^1 \times S^1/\sim$, which is an unorientable manifold diffeomorphic to a punctured Klein bottle. The puncture, which is the boundary of $S^1 \times S^1/\sim$ is isomorphic to S^1. This boundary is the diagonal of $S^1 \times S^1$, the set of fixed points. The first advantage then is in the size of the hypergraph we will construct. The number of vertices is now just the square of the number of points in the boundary when we discretize. Since this number will scale as the square root of the number of pixels in the image, the total number of vertices scales as the number of pixels in the image, or the square root of what we had before. This reduction in complexity was to be expected however, since we are clearly drastically reducing the size of the search space. There are other advantages that are less expected though.

A second-rank tensor can be induced on $S^1 \times S^1/\sim$. This is not an orientation: it is symmetric. It is not the induced metric from $\mathcal{D}$ either, which is positive definite. It is rather a pseudometric, with eigenvalues ± 1. As such it defines a *lightcone*, corresponding to those vectors with negative length. These will be called *time-like*. The positive length vectors will be called *space-like*. The tensor is simply $\pi_1^*\omega_1 \otimes \pi_2^*\omega_2 + \pi_2^*\omega_2 \otimes \pi_1^*\omega_1$, where the ω are orientations on the circles and the π are the projections from $S^1 \times S^1$. Notice that the symmetry of the tensor is essential to it being well-defined on $S^1 \times S^1/\sim$ as opposed simply to $S^1 \times S^1$. The ω can typically be taken to be dt_1 and dt_2, where t_1 and t_2 are coordinates on the circles.

Recall that the two maps σ_1 and σ_2 that define the self-matching must have opposite orientations to ensure that μ is orientation-reversing. This is equivalent to the fact that the map $\langle \sigma_1, \sigma_2 \rangle$ from S^1 into $S^1 \times S^1$ (or equivalently the map from Υ into $S^1 \times S^1/\sim$) must have tangent vectors with negative length according to the pseudometric: the hyperpath defined by this map must be *time-like*. Simply put: t_1 must be increasing while t_2 is decreasing or vice-versa.

Now consider a point $\langle p, v \rangle$ in the tangent bundle of $S^1 \times S^1/\sim$. The set of points in the *past* light cone of this point (which is to say the set of points a time-like hyperpath from which could end at p with tangent vector v) must intersect the boundary of $S^1 \times S^1/\sim$. The intersection of the past light cone with the boundary is therefore the set of fixed points from which a time-like hyperpath with distinguished vertex p and tangent vector v at p could have originated. It is easy to see that if for two points $\langle p, v \rangle$ and

$\langle q, u \rangle$, these intersections are disjoint, then any time-like hyperpaths from the boundary to these points must be disjoint too. This does not depend on the exact identification of the terminal vertices of these hyperpaths. In other words, given two points, we can now make a *local* assessment of disjointness for a whole *class* of possible initial fixed points. Figure 39 shows the geometry. The consequence is that we do not need the full notion of colour that was so important in algorithm A to ensure disjointness. We only need a weakened form of it, summarized in the notion of past and future lightcones. Each vertex can come in one of two colours, corresponding to a choice of future light cone. This is enough information to ensure disjointness and we need place no stronger restrictions. The situation is essentially the same as the bi-directional version of Dijkstra's algorithm, except that here we are dealing with hyperpaths in a hypergraph. The stopping condition still asks for two copies of the same vertex with disjoint colours, except that now there are only two colours. The first invariant, equation V.17, will now state that the weight of each vertex in S is the minimum over all hyperpaths whose terminal vertices are a *subset* of the intersection of the past light cone of the vertex with the diagonal. Thus because we have a weaker notion of colour, we have a stronger invariant. This has a very important consequence. By considering figure 39 it is clear that the union of the intersections with the diagonal of the past lightcones corresponding to the same vertex but with different colours is the *whole* of the diagonal. This means that after the post-iteration step in algorithm A, we will have searched all possible combinations of terminal points for the hyperpath. Every subset of the diagonal has been considered as a possible set of terminal vertices. Thus we have no need to pick a set of terminal vertices or indeed to initialize the algorithm at all. All points on the diagonal are set to zero weight at the beginning of the algorithm, and assigned consistent future light cones. Consequently all serve as terminal vertices for hyperpaths.

Figure 40 shows the hyperedges between pairs and triples of points that constitute the hypergraph. These all respect the time-like nature of the hyperpaths.

Some examples of self-matchings computed in this way are shown in the members of the pairs in figures 42 to 46 that do not have connecting lines to the target.

3.2. *Incorporating the Target*

Let us call the hypergraph of the previous section $\mathcal{H}$. Its vertex set is then $V_{\mathcal{H}}$, a discretized version of the product of the boundary with itself. Let the topology tree of the target be S, with vertex set U and edge set E. We can form a hypergraph from these as follows. The vertex set will be $V_{\mathcal{H}} \times 2^U$.

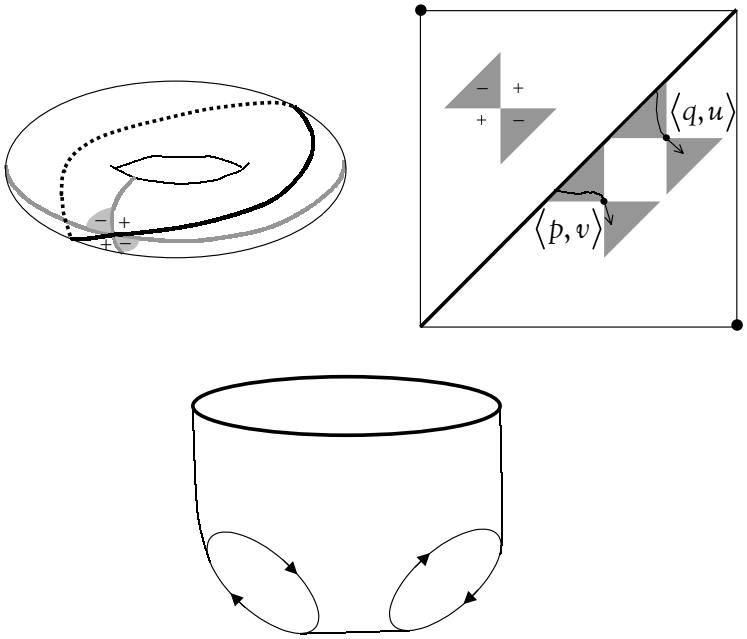


FIGURE 39. The left hand diagram shows the torus $S^1 \times S^1$. In grey are shown boundaries corresponding to the factors. The solid and dotted black lines are the the diagonal of the product. Note the light cone formed by the combination of the product structure and the diagonal. When the torus is 'divided' by the action of the twist arrow, the diagonal will become a boundary isomorphic to S^1. The right hand side of the figure is the same thing, but now the torus has been 'cut open'. The upper and lower sides of the square are identified, as are the right and left sides. Glueing them together recreates the torus in the left of the diagram. Two points are shown, together with their past light cones. This shows how disjointness of the intersection of the past light cones with the boundary guarantees disjointness of time-like hyperpaths to the points. The division by the twist arrow now folds this diagram along the black diagonal line to create a 'triangle', except that the vertices of the triangle are identified, as are two of the edges. The third edge is the boundary. This is shown in the lower part of the figure. The two lower directed boundaries must be identified in such a way that the orientations match. The thick black circle at the top is the diagonal.

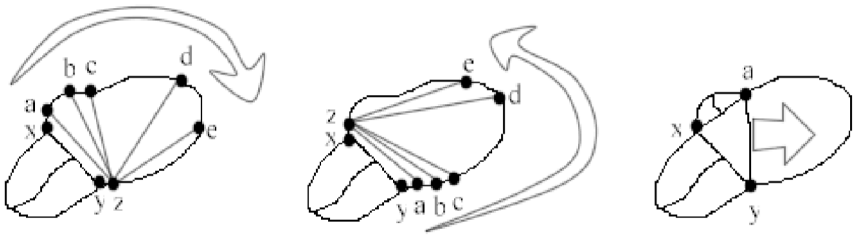


FIGURE 40. The two left hand parts of the figure show the valency two edges containing the vertex $\langle x, y\rangle$. These are extended beyond nearest neighbours to allow for non-isomorphic self-matchings. Both sides of the self-matching must advance at least one step (to $\langle a, z\rangle$ in the figure) along the discretized boundary into the future lightcone of $\langle x, y\rangle$, but one side may advance further (to $\langle b, z\rangle$ etc.). The right hand side shows a valency three hyperedge containing $\langle x, y\rangle$ It can be thought of as a choice of a third point a in the future light cone of $\langle x, y\rangle$, shown by the block arrow.

The hyperedges of valency two are $\{\langle\langle v, c\rangle, \langle u, c\rangle\rangle : \{u, v\} \in H_{\mathcal{H}}\}$. The hyperedges of valency three are of the form $\{\langle u, c\rangle, \langle v, c'\rangle, \langle w, c''\rangle\}$. The conditions on them are that

$$\{u, v, w\} \in H_{\mathcal{H}}$$
$$c'' = c \cup c' \cup \{s\} \tag{VI.4}$$

where $s \in U$ is such that the forest induced by $c'' \subset U$ is a tree. (Further conditions can be imposed if it is desired to preserve the planar ordering of the edges around a tree vertex.) The terminal vertices required for algorithm A are those of the form $\langle d, l\rangle$, where d is in the diagonal of $V_{\mathcal{H}}$, and l is a leaf of S. Thus the terminal vertices of all the hyperpaths considered are matched to leaves of the target topology tree S. We can now apply the version of algorithm A described in the last section, with just two colours. The initial vertices in the heap are the terminal vertices of one colour only. At any stage, the hyperpath $\pi(x)$ represented by any vertex $x = \langle v, c\rangle$ corresponds to a rooted subtree of S induced by the vertices in c, with leaves a subset of the leaves of S. The iteration stage ceases when a vertex $x = \langle u, c, \kappa\rangle$ is drawn from the heap Ω such that a vertex $y = \langle u, c', \neg\kappa\rangle$ is in the set $\mathcal{S}$, where $\neg\kappa$ is the colour opposite to κ, and where $c \cup c' = U$ and $c \cap c' = \emptyset$. The latter condition ensures that the union of the subtrees represented by c

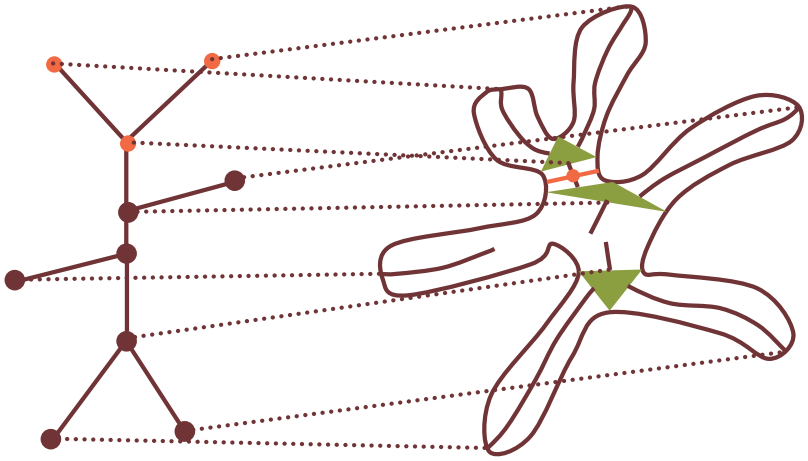


FIGURE 41. The figure shows some of the hyperpaths explored by algorithm A during the iteration stage of matching to the target. A hyperpath is represented as follows. For a vertex $\langle v, c\rangle$, the vertex $v \in V_{\mathcal{H}}$ is represented by the median point of its pair of matching boundary points, for example the red dot on the right. This produces the solid lines within the boundary on the right. The hyperedges of valency three are the green triangles. The value of c for a vertex is represented by the dotted lines from the right hand side to the topology tree of the target on the left. c is the set of topology tree vertices that correspond to ancestors of v in the hypergraph $\pi(v)$. For example, for the red dot, the value of c is the set of red tree vertices.

and c' is the whole of U, and thus that the topology tree of the hyperpath formed by the union of $\pi(x)$ and $\pi(y)$ is isomorphic to S. The post-iteration stage is the same as before, except that we restrict ourselves to completable sets $Z = \{z = \langle v, c\rangle\}$ whose second members c are disjoint and have union equal to U. The proof goes through as before with the appropriate restrictions. Figure 41 shows some of the hyperpaths explored by the algorithm during the iteration stage, and their correspondence to the target tree on the left.

3.3. *Demonstrations*

Figures 42 to 46 show examples of part structures computed for boundaries with and without the use of targets. The metric we use is the inverse delta function metric on topology trees mentioned above, supplemented by

FIGURE 42. An example of an occluded part being restored. The target is in red. Lines show the correspondence between the part structures. Some lines are omitted for clarity.

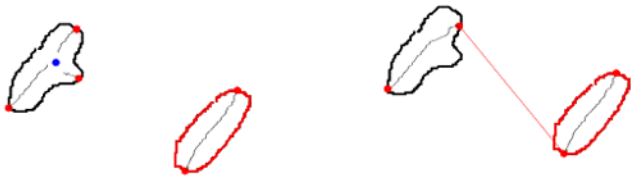


FIGURE 43. An example of a spurious part being removed.

an energy ϵ on part matchings that simply compares lengths. In the figures, the target is always shown in red. Its part structure is computed following the procedure outlined at the beginning of section 3: the boundary is given and the part structure computed in the absence of a target. The lines show the correspondence between the part structures of the target and the boundary.

The examples demonstrate the ability of targets to 'see' parts where they are expected even if the data suggests them only weakly, and to ignore parts where they are not expected even if strongly supported by the data; and thus complete the part structure of the boundary in the presence of occlusions and other distortions. In the case where the boundary is allowed to vary, we would expect this type of behaviour to complete the boundary in areas where it is not visible in a more intelligent way than could be done without the target. An arm or a leg could be filled in for example even if it is occluded in the image, solely because the target suggests it should be present. The results here are very stable to parameter variation as you might expect, since the topology of the result is determined by the topology of the target.

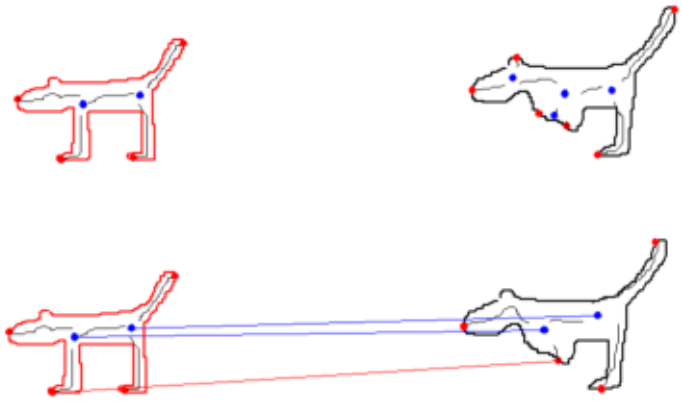


FIGURE 44. Occluded or articulated parts are separated.

4. POSTSCRIPT: THE FULL PROBLEM

In the full problem, we once again allow the boundary to vary. The model itself for the full problem is clear, and ameliorates some of the problems associated with the model in the absence of a target. For example, the correspondence to the target will push up the energy of part structures with topologies that do not match well that of the target (or perhaps it will eliminate them entirely as in the case of the delta function metric). Consequently, the minimum energy self-matching can be made more likely to be one with a complex topology. The target offsets the tendency, due to the extensive nature of the energy, to find simple topologies by skipping.

Secondly, if the energy of the correspondence to the target depends on the geometry, then the target will effectively have a size. The energy of the correspondence will favour those boundaries of a similar extension to the target. Again this counteracts the effect of the extensive energy, and may enable the abandonment of the initialization by moving the energy minimum from its normal trivial position in $\mathcal{P}$ to a point representing a boundary with a finite extension.

These questions are all moot however without the development of an algorithm to minimize the full energy, equation VI.3. As the reader will have guessed from the twin facts that we have already entered the postscript for this chapter and that the next chapter is entitled "Conclusion", such an algorithm must await future research. It seems likely that a variant of algorithm A can be found that would enable the problem to be solved in the case of the constrained problem with fixed terminal vertices. In that case the

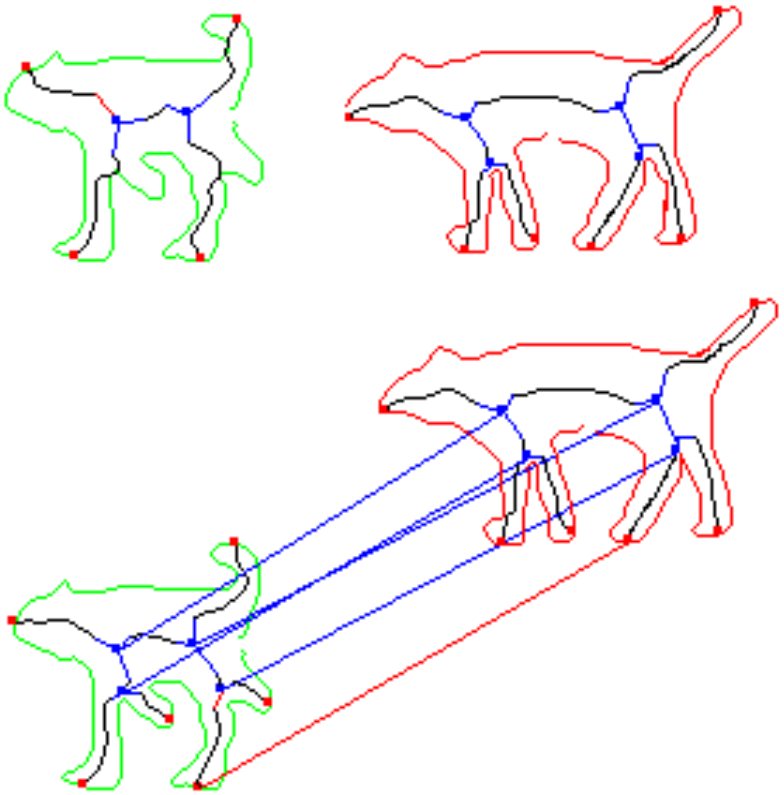


FIGURE 45. Parts shortened by articulation are identified.

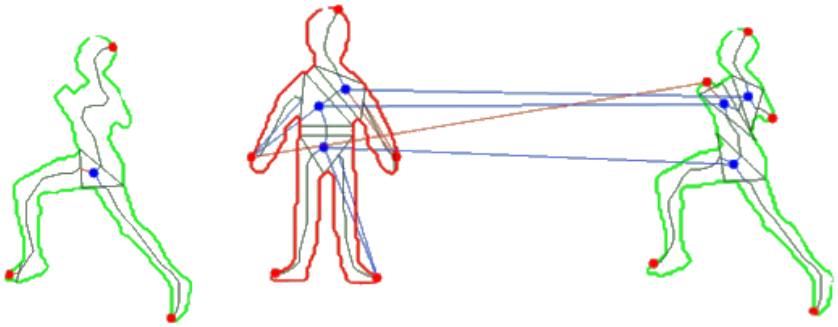


FIGURE 46. Parts shortened by articulation are identified.

problem looks rather like that described in section 3.2, except that the need for disjointness may create problems.

If the terminal vertices are allowed to vary, then we are faced with an 'all-pairs shortest path' problem, except that now it will be an 'all-tuples shortest hyperpath' that we wish to find, but including the target to avoid triviality. Whether such an algorithm can devised, or be made efficient enough to be considered is unknown.

Finally, there is still the possibility of a minimum ratio weight hypercycle algorithm, which would solve all the above problems. It remains to be seen whether and how a target might be incorporated there.

CHAPTER VII

CONCLUSION

We show how all the methods can be viewed as functionals on boundaries of increasing complexity. A summary of contributions is followed by a discussion of the weaknesses of the methods presented and of work to come.

The work described in this thesis fits into a unified framework, which throws light on possible ways forward. Throughout the last four chapters, we have been dealing in one way or another, with energy functionals defined on boundaries in various spaces, represented by embeddings of the circle.

In chapter III we dealt with the space of boundaries in the single image domain $\mathcal{D}$, usually a subset of $\mathbb{R}^2$. This enabled us to represent average properties of the boundary and the region contained within it by defining appropriate energy functionals on this space. While extremely useful, the probability distributions constructed from such energy functionals are rather broad, giving high probability to the boundaries of almost any object in the image, as well as to many boundaries that do not correspond to objects. As we argued in chapter II, to perform object recognition successfully, a number of representations at different levels of specificity are necessary to focus the distribution on the objects we wish to find.

To create more focussed distributions, we introduced extra structure corresponding to more specific representations of objects. In chapter IV this extra structure was information about the position and motion of the boundary in three-dimensional space, as encoded in a correspondence between boundaries in several single image domains. In chapter V, the extra structure was information about the part structure of the boundary, as encoded in a correspondence between the boundary and itself.

Both these latter structures we represented as boundaries Γ in a space $\mathcal{D} \times \mathcal{D}$. Such a boundary can always be viewed as generated by two boundaries in $\mathcal{D}$, Γ_1 and Γ_2. The difference between the spaces used in the two chapters is simply that in chapter V, the Γ_a shared a common image γ, differing only in their co-image, where as in chapter IV the images of the Γ_a were *a priori* unrelated.

In the case of part structures, energy functionals on Γ can be viewed as functionals not only on boundaries in $\mathcal{D} \times \mathcal{D}$, but on boundaries in $\mathcal{D}$ itself. The optimization over the part structure itself can be separated from that over the boundary because of the separation of the Γ_a into image γ and co-images σ. In principle anyway, by optimizing over the co-images for an arbitrary γ, we would obtain a new and complicated functional over the space of boundaries in $\mathcal{D}$, $\mathcal{B}$. This functional would be focused on those boundaries with 'good' articulated part structures. By solving the combined optimization problem over γ and σ, or equivalently over $\mathcal{P}$, we are optimizing this complicated functional in one fell swoop rather than in two stages. One can see the whole process of introducing part structures solely as a method for defining this more sophisticated functional.

The introduction of targets added another layer to the model. As can be seen from the algorithm of the last chapter, the space of boundaries, part structures, and part structure correspondences to the target can be seen as hyperpaths in a certain space, although we did not formulate this explicitly. It is to be expected that a hypercycle formulation of the problem would allow the representation of the full space $\mathcal{Q}$ as a space of boundaries also. In addition, energy functionals on $\mathcal{Q}$ can be regarded as defined on boundaries in $\mathcal{D}$ also. Optimizing over part structure and part structure correspondence defines an even more specific distribution on $\mathcal{B}$.

In chapter IV we dealt also with the case of more than two images, which allowed us potentially to include even more sophisticated information about shape by examining information at three or more points at a time. We did not take advantage of that possibility however, as we used only pairwise energies.

A picture thus emerges of extra structure defining more complex functionals taking into account higher order information about the boundary γ in the image domain. We will discuss this further in section 1.6.

1. Contributions

The two main contributions of this thesis are the definition and application of a new form of energy functional for the segmentation of regions from single or multiple images, and the extension of the notion of functionals on boundaries in real images to include the description of part structure. The importance of the first lies in the properties of the functional itself, which solve some long-standing problems in the computer vision literature. The importance of the second lies rather in the elucidation of a theoretical framework that allows the algorithmic issues to be described cleanly, thus enabling both future developments, and, as a first consequence, the devising of an algorithm to solve the problem of finding boundaries together with their part structures in real images.

1.1. *Energy Functional*

The functional is defined on one-dimensional boundaries (closed curves) in manifolds. The importance of this functional to computer vision consists in several properties deriving from its form, rather than from a commitment to any particular choice of data. This has the consequence that very general types of information can be incorporated into the functional.

The form of the functional enables the following important properties. First, any functional of the new form can be globally optimized using the same polynomial-time algorithm. It is remarkable that polynomial-time optimization is possible in a space as large as that of one-boundaries in manifolds. Previous models in computer vision do not have this property, relying as they do upon local gradient descent methods. The ability to extract the global optimum means that the results are guaranteed to represent a property of the image rather than of the initialization used.

Information concerning a region in an image can be separated into two types. One type of information concerns the relation of the region to the rest of the image. This is associated with the boundary of the region. The other type concerns intrinsic properties of the region itself. This is associated to the interior of the region.[1] The history of computer vision contains many attempts to combine these two types of information, none of them fully successful, and none of them globally optimizable. The new functional allows the incorporation of both types of information on the same footing. It is easy for example to combine information about region colour with information about image function gradients on the boundary.

Without prior information indicating that the search should be directed to large or small regions, the models that we construct should not be biased towards one or the other. In other words, the models should be scale-invariant. The new form of energy functional possesses this property. Previous models are not scale-invariant. If appropriate to the task, a dependence upon scale can be introduced into the model, thus allowing a principled and controlled approach to region size. In addition, a scale-independent energy prevents skipping, in which the boundary pulls away from the data solely because of the extensive nature of the energy.

The new form of functional is defined on one-boundaries. It therefore by definition finds closed curves. This overcomes a problem of previous contour

[1]This classification does not exclude the possibility that information that does not actually lie on the boundary of the region might be compared with the rest of the image. This comparison must however be done in a principled way. Information in the interior may be associated with the boundary at larger scales. This type of information may be used in the functional described here. Unfortunately this introduces the vexed but extremely important question of how best to combine information from across scales.

models in computer vision: the competition between closure and global optimization. The importance of closure is known. Psychological research has noted its contribution to contour salience in human vision, and although occlusion means that the boundaries of objects in images are frequently broken into open segments, human beings have the ability to complete these segments into a closed curve. Nevertheless, the satisfactory optimization of closed contours has remained elusive. Either the attempt has been abandoned completely, as in the gradient descent approaches, or ad hoc and heuristic mechanisms have been introduced. In contrast, global optimization over the space of open curves between two fixed points is relatively simple. Attempts have been made to generalize these approaches to closed curves, but the global nature of topological information such as closure has resulted in failure. The new form of energy functional provides the solution to this problem.

Other Applications. The functional is not restricted to one-boundaries in two dimensions, although it is here that the region/boundary duality adds extra power to the model. The model can also be applied to the task of extracting regions and boundaries from several images simultaneously. Problems of this type arise in computer vision when the data takes the form of stereo pairs or motion sequences of images. The typical approach to these problems is to try to compute a correspondence between the images that is dense on each image. While this is useful in some contexts (for example, visualization, graphics), it is not at all clear that it is the correct approach for all tasks. To find an approaching missile for instance, and to identify its distance or motion, we are not concerned with the background or indeed with any other objects in the image. We would like to find the object and simultaneously know its motion and/or distance. The approach described here performs this task. In dealing with region segmentation from multiple images, we are faced with two distinct types of information. The first is that contained in each image in isolation. The second is the information contained in comparisons between the images. Clearly both are important, in the same way that both region and boundary information are important in the single image case. The functional can incorporate both types of information naturally.

The resulting information can be utilized in an additional way also. Most models that compute dense correspondences use a prior probability on the correspondence map that favours smoothness. Without such a term, the data will typically be over-fitted. Unfortunately, some of the most important information in the correspondence map is contained in points of very high gradient or discontinuity. The prior models adopted for the correspondence maps tend to smooth over this information. The introduction of non-linearities to deal with this issue renders the problems algorithmically intractable, and so an

alternative approach would be useful. By using the information about boundary correspondences produced by the method described here to fix boundary conditions and to weaken the smoothing terms in the vicinity of the extracted boundaries, the performance of dense computations can be improved.

1.2. *Hypergraph Framework*

The description of part structure introduced in [Liu97, LG97, LGK98] applied to boundaries in the plane, but no clear theoretical framework was elucidated to suggest how the notion of part structure might be extended to real images, or in other words how to optimize a functional on the combined space of boundaries and part structures, $\mathcal{P}$. The appropriate framework is that of hyperpaths or hypercycles in a hypergraph, a novel generalization of paths in graphs. This formulation is based on the representation of part structures as generalized boundaries in $\mathcal{D} \times \mathcal{D}$, containing well-defined and limited types of discontinuity. An optimal hyperpath algorithm was developed as a generalization of Dijkstra's algorithm to this new setting.

In the special case in which we are searching for an object that has only one part, meaning it has a global reflection symmetry, the hypergraph reduces to a graph. No initialization is necessary and an intensive energy can be used because the minimum ratio weight cycle algorithm applies. The model thus possesses all the benefits described in the previous section.

In the case that discontinuities are permitted, the minimum ratio weight cycle algorithm no longer applies because the hypergraph has hyperedges of valency greater than two. In this case the new optimal hyperpath algorithm solves the problem of finding boundaries and part structures simultaneously in the case that a set of possible terminal vertices is given.

In both cases, the ability to incorporate symmetry information into such models of boundaries in images is new, and promises many further developments.

1.3. *Targets*

The use of targets to further narrow the probability distribution on objects is in itself not new. The use of templates in general falls under this rubric. The coupling of targets, defined in terms of their part structures, to other part structures that are coupled to boundaries that are coupled to image data, the whole being optimized over the full space of structures to obtain an exact MAP estimate, is however a novel form of visual system, and another, somewhat less defined contribution of this thesis. Although this has not been completed algorithmically, the theoretical picture is filled in and completion seems possible. The probable effect of targets has been looked at in miniature in the case of fixed boundaries and found to behave as expected. It remains to

be seem whether the promise of abandoning the sequential picture of visual processing will be fulfilled.

1.4. *Theoretical Weaknesses*

The energy density functional, as was demonstrated in table 2.2, is the only simple form built from linear functionals on boundaries that makes sense in isolation, since it is the only form that does not possess trivial solutions. To improve it means moving to different and more complex structures. The generalization to hypercycles to be discussed later would not change this, since this generalization changes the space on which the functional is defined, but not its form.

In the application to single images that we have described, the functional cannot incorporate second order information about the boundary. In particular it cannot depend on the curvature of the boundary. In previous active contour models, this dependence has taken the form of favouring curves with small integrated squared curvature, thus favouring straight lines over curves and smooth curves over jagged ones with the same average curvature. To some extent, the regularizing effect of the denominator produces the same effect, but it would be useful to be able to control this more effectively.

In the application to multiple images, the fact that we can only describe correspondences between boundaries, and not between regions is counter-intuitive and limits the effectiveness of the model. Finding an optimal $(n-1)$-boundary (or equivalently open n-chain) in n dimensions, whether it be compact or a hyperplane, is soluble in polynomial time using maximum flow techniques (for the hyperplane case see [Ish00]); this would seem to be because of duality. In two dimensions these latter cases collapse to the case of one-boundaries. Unfortunately, in common with most problems beyond one dimension, it appears that the problem of finding an optimal *surface* (an open two-chain) is NP-hard.

In the application to part structures, the problems centre on the use of hyperpaths (as opposed to hypercycles). Whether we use extensive energies, or use algorithms for extensive energies to optimize intensive energies, we still need to constrain the space of solutions to avoid trivialities. This means initialization by the user or, what is equivalent, by some as yet unspecified mechanism. Clearly this is unsatisfactory. The only case where this can be avoided at present is when there is only one part, and hence no discontinuities. In that case the search for the optimal symmetric object can be carried out using the minimum ratio weight cycle algorithm. In the case where discontinuities are permitted, initialization is the only choice at the moment.

1.5. *Practical Weaknesses*

The practical problem with all the methods, with the exception of the application of the energy density to single images, is memory resources. As we have seen in chapter IV and V, both the minimum ratio weight cycle algorithm and the optimal hyperpath algorithm use large quantities of memory, at the limit of what is available for desktop machines, when the input is of the size generated by image data. This is a continual problem in computer vision. Images require a great deal of space, and any computations performed on them require even more. At present, without access to more computational resources, this limits what can be done to images of the order of 50×50 pixels. Better implementations could surely improve on this, but not greatly. The usual caveat applies however: a year and a half ago or less, the the minimum mean weight cycle algorithm required more memory than was currently available on desktop machines.

1.6. *Future Developments*

It would be fascinating to have a hardware device that implemented a parallel version of the minimum mean weight cycle algorithm. Despite the algorithm's theoretical drawbacks, this would sidestep the memory issue and enable extremely fast extraction of regions from images. Parallelization remains a possibility for the minimum ratio weight cycle algorithms also, since negative cycle detection can be cast as matrix multiplication in a path algebra. This opens the way to neural implementations and interpretations.

The absence of good exact algorithms for the optimal surface problem need not be the end of the story if there exist good approximation algorithms, as there are for the minimum quotient cut problem for example. Although the optimal surface problem can be stated in graphical terms, it is more correctly stated in terms of a two-polytope that possesses platelets in addition to edges and vertices. In two dimensions this distinction is academic, but this is no longer so in higher dimensions. (The difficulty can be envisaged if one thinks of trying to find an optimal cycle by looking for subsets of vertices in the line digraph of a given graph.) The development on algorithms on these structures would be a useful endeavour.

Dependence on the curvature of the boundary can be incorporated by lifting the boundary to the tangent bundle of the image domain. This can be done by a choice of frame on the circle, although nothing should depend on this choice. Having done this, the integrals over the boundary in the tangent bundle can include curvature terms naturally. There are a number of hurdles to jump to make such a scheme work. It is necessary to restrict the optimization to boundaries in the tangent bundle that are lifts of boundaries

in the image domain. It may not be possible to achieve this entirely, so that self-intersecting boundaries may become a possibility for example. There is also the question of how best to discretize the tangent bundle. The line digraph of the graph embedded in the image domain is the natural candidate, but this is not a very fine discretization of the tangent space at each point. Nevertheless, the importance of curvature means that this is a worthy line of investigation.

The description at the beginning of this chapter of all the models considered as functionals on boundaries suggests the study of such and related things in their own right. The pattern formation and analysis community for example [Gol97] has examined functionals on products of boundaries in the plane, $S^1 \times S^1 \overset{\gamma\times\gamma}{\hookrightarrow} \mathbb{R}^2 \times \mathbb{R}^2$. Such functionals, quadratic on chains as opposed to linear, can describe self-interacting boundaries, and lead to remarkably complex shapes with specific properties. There are also of course higher polynomials also. Another possibility, touched on in a footnote, is the development of a model for local rotational symmetry along the lines of the self-matching model for local reflection symmetry. This too would be a functional on a boundary. It is worthwhile to concentrate on these one-dimensional structures for two principal reasons. The first is that in two dimensions there is nothing else, and images are two-dimensional. The second is that it is well-known, and our comments above exhibit this, that problems in dimensions above one become extremely hard to solve. This is not so much the so-called 'curse of dimensionality' that plagues numerical work in partial differential equations for example, as it is the loss of an ordering once we move beyond one dimension. One of the great strengths of moving to hypergraphs is that they enable the use of one-dimensional objects (only partially ordered in this case) to describe more complex structures.

In view of the progression from locally optimized closed curves and globally optimized paths (using Dijkstra's algorithm) to globally optimized regions and boundaries using the minimum ratio weight cycle algorithm, it would seem that the most important development to be made is the extension of the minimum ratio weight cycle algorithm to a minimum ratio weight hypercycle algorithm. The theory to describe the structures is in place. It is possible that no such algorithm exists with polynomial time performance, but its existence would solve the problem of initialization and scale-dependence (and hence skipping). If it turns out that such an algorithm does not exist, there is still the possibility of using an intensive version of a hyperpath energy to solve at least the skipping problem and encourage more complex topologies.

The incorporation of targets promises to alleviate some of these problems also, and so a natural parallel development would be the incorporation of targets into the optimal hyperpath algorithm. This seems within reach, since

it should involve a generalization of the algorithm in chapter VI that is inverse to the specialization from arbitrary boundaries to a fixed boundary that we made in the absence of a target. Better still would be a combination of the two approaches in which a target is incorporated into a minimum ratio weight hypercycle algorithm. Such a system might actually be able reliably to identify shapes in images, a possibility that would bring with it a great many applications as well as the possibility of genuine empirical performance testing. It would also confirm the approach to visual system structure that guided the work in this thesis, which in its turn should be of great significance to other recognition tasks and beyond.

APPENDIX A

Differential Forms

We give a very brief description of differential forms and provide a dictionary to convert formulas to the language of vector calculus.

We provide a short dictionary to translate from the language of differential forms to vector calculus, with no attempt at definitions. A good reference for differential geometry using this language is [CBDMDB96].

Briefly, differential forms are linear functionals on antisymmetric products of vector spaces. For manifolds they are defined pointwise on the tangent space at each point. They also allow a beautiful theory of integration on manifolds, and in this capacity they are thought of as *co-chains*, linear functionals on the vector space of chains in a manifold. Their advantages are great concision and uniformity of notation; independence of basis or coordinates; manifest invariance to diffeomorphisms and other transformations; and generality.

Given a coordinate system x^a, a local basis for the space of forms is given by the set of dx^a. These are defined so that they act on the local coordinate basis for vector fields $\frac{\partial}{\partial x^a}$ as

$$dx^a(\frac{\partial}{\partial x^b}) = \delta^a_b$$

. Thus a one form can be expressed as $A_a dx^a$ (summation understood). By convention a zero-form is a function. Bases for higher forms are formed using antisymmetric products of the dx^a, usually notated $dx^a \wedge dx^b$. In other words, forms are antisymmetric covariant tensors. The wedge product $\wedge$ acts on any two forms and results in a form with degree equal to the sum of the constituents. We will usually suppress the $\wedge$. The action of a one-form $\mathbf{A} = A_a dx^a$ on a vector $v^b \frac{\partial}{\partial x^b}$ is then given by $A_a v^a$ (summation understood).

On a manifold with metric, every one-form defines a vector field and vice-versa. An arbitrary one-form therefore means equivalently an arbitrary vector field. The vector field that corresponds to a one-form $\mathbf{A}$ will be denoted $v_{\mathbf{A}}$: $\mathbf{A} \Rrightarrow v_{\mathbf{A}}$. The vector field corresponding to a one-form is that whose inner product with other vector fields mimics the action of the one-form.

There is a derivative operator that takes p-forms to $(p+1)$-forms. It takes the form $A_a dx^a$ to $\frac{\partial A_a}{\partial x^b} dx^b \wedge dx^a$. It act similarly on other forms.

The integral of a n-form $\mathbf{F} = F_{a,b,\ldots} dx^a \wedge dx^b \wedge \ldots$ on an n-dimensional manifold is

$$\int_M \mathbf{F} = \int_M f \, dx^1 dx^2 \ldots dx^n \tag{A.1}$$

where $f = *\mathbf{F}$. An orientation, a nowhere-vanishing n-form, is used to decide which half of the space of n forms is positive at each point.

In two dimensions, if the metric is Euclidean, the Hodge star works like this: $*\mathbf{A} \Rrightarrow v^{\perp}_{\mathbf{A}}$, where $v^{\perp}$ is the rotation of v by $\frac{\pi}{2}$ in a direction specified by the orientation. If the metric is not Euclidean, and if it has determinant g, then $*\mathbf{A} \Rrightarrow \sqrt{g} v^{\perp}_{\mathbf{A}}$. In general, the Hodge star converts p-forms to $(n-p)$-forms, where n is the dimensionality of the space. Since p-forms are integrated over p-chains, the form $\mathbf{A} * \mathbf{B}$ ($\wedge$ suppressed), where both $\mathbf{A}$ and $\mathbf{B}$ are p-forms can always be integrated over the space on which they sit. This defines an inner product on the space of forms. The inner product of two p-forms $\mathbf{A}$ and $\mathbf{B}$ is

$$\ll \mathbf{A}, \mathbf{B} \gg = \int \mathbf{A} * \mathbf{B} \tag{A.2}$$

The pullback of a function f (zero-form) on a manifold N by a map $M \xrightarrow{\pi} N$ is given by

$$\pi^* f = f\pi \tag{A.3}$$

For a one-form it is slightly more complex. Recall that the derivative of π pushes forward a vector in the tangent space at $y \in M$ to one in the tangent space at $x = \pi(y) \in N$: $\pi_* v^a = v^i \frac{\partial \pi^a}{\partial y^i}(y)$ in some coordinate bases x^a on N and y^i on M. Then the pullback of a one-form $\mathbf{A}$ on N is

$$\pi^* A_y(v) = A_{\pi(y)}(\pi_* v) \tag{A.4}$$

The pullback of higher forms is defined similarly. The exterior derivative commutes with pullback.

As in the main text, we will use γ to indicate an arbitrary element of the equivalence class of maps that defines a boundary in a manifold. A region is described in a similar way by an equivalence class of embeddings of the two-dimensional disc, D^2. An arbitrary element of the equivalence class for some region will be denoted S. Coordinates on the co-domain will be denoted x^i. On S^1, it will be denoted t and will be assumed to run from 0 to 1, although all equations are independent of this choice (all topological niceties will be ignored). On D^2, they will be denoted y^a and also assumed to run from 0 to 1.

The integral of a one-form $\mathbf{A}$ on a manifold M over a boundary ∂R in that manifold is given by

$$\text{(A.5)} \qquad \int_{\partial R=\gamma(S^1)} \mathbf{A} = \int_{S^1} \gamma^{ast}\mathbf{A} \qquad \Rrightarrow \int_0^1 dt\,(\gamma'(t), v_{\mathbf{A}}(\gamma(t)))$$

where $(\cdot,\cdot)$ is the inner product on M (we need this to convert from one-form to vector). This is independent of the choice of representative γ, as can be seen by substitution of $\gamma\epsilon$ for γ.

The integral of a function on M over a boundary requires a metric on the circle to convert the pullback of the function to a one-form. This must be defined as the metric pulled back from M for the integral to be invariant to change of representative.

$$\text{(A.6)} \qquad \int_{S^1} *_\gamma \gamma^* f = \int_0^1 dt\,|\gamma'(t)| f(\gamma(t))$$

If we take the inner product of $g = \gamma^*\phi$ with itself (ϕ is a function or one-form on M) we find

$$\text{(A.7)} \quad \ll g, g \gg = \ll \gamma^{ast}\phi, \gamma^*\phi \gg = \int_{S^1} \gamma^*\phi *_\gamma \gamma^*\phi$$
$$\Rrightarrow \begin{cases} \int_0^1 dt\,|\gamma'(t)| g^2(\gamma(t)) & \text{if } g \text{ is a function,} \\ \int_0^1 dt\,\frac{1}{|\gamma'(t)|}(\gamma'(t), v_g(\gamma(t)))^2 & \text{if } g \text{ is a one-form.} \end{cases}$$

Naturally this is positive.

The integral of a two-form over a region R is given by

$$\text{(A.8)} \qquad \int_{R=S(D^2)} \mathbf{F} \Rrightarrow$$
$$\int_{y^1=0}^1 \int_{y^2=0}^1 F_{ij}(S^i(y^1,y^2)) \frac{\partial S^i}{\partial y^a}(y^1,y^2) \frac{\partial S^j}{\partial y^b}(y^1,y^2)\, dy^1 dy^2$$

For the particular case of $\mathbb{R}^2$, Stokes' theorem becomes

$$\text{(A.9)} \qquad \int_{\partial R} \mathbf{B} = \int_R \mathbf{dB} \Rrightarrow$$
$$\int_0^1 dt\,(\gamma'(t)^\perp, v_{\mathbf{B}}(\gamma(t))^\perp) = \int_{y^1=0}^1 \int_{y^2=0}^1 \nabla \cdot (v_{\mathbf{B}}^\perp)(S(y^a))\, dy^1 dy^2$$

APPENDIX B

The Minimum Ratio Weight Cycle Algorithm

Some details of this algorithm are given here, including a proof of correctness.

The claim in section 4.3 about the relation between the minimum ratio weight cycle problem and a related minimum weight cycle problem is proved as follows:

PROOF. Suppose t^* is the solution to $w_t(C_t^*)$, where C_t^* is the minimum total weight cycle for the weight w_t. Then we have by definition that $\lambda(C_{t^*}^*) - t^*\tau(C_{t^*}^*) = 0$, and hence that $t^* = \frac{\lambda(C_{t^*}^*)}{\tau(C_{t^*}^*)}$. The claim is that t^* is the minimum ratio weight cycle for $W(C) = \frac{\lambda(C)}{\tau(C)}$, and that therefore $C_{t^*}^*$ is a minimizing cycle. Suppose this were not the case. Then there must exist a cycle C such that $t = \frac{\lambda(C)}{\tau(C)} < t^*$. This however would mean that $w_{t^*}(C) = \lambda(C) - t^*\tau(C) < 0$ or in other words that $w_{t^*}(C) < w_{t^*}(C_{t^*}^*)$, contradicting the assumed minimality of $C_{t^*}^*$.

For the reverse argument, suppose that t^* is the minimum ratio weight for $W(C) = \frac{\lambda(C)}{\tau(C)}$ and that C^* is a minimizing cycle. Then by definition, $t^* = \frac{\lambda(C^*)}{\tau(C^*)}$, or in other words, $w_{t^*}(C^*) = \lambda(C^*) - t^*\tau(C^*) = 0$. Now the claim is that C^* is a minimum total weight cycle for weight w_{t^*}, $C_{t^*}^*$, and that its weight is zero: $w_{t^*}(C^*) = 0$. Suppose this were not the case. Then there must exist a cycle C such that $w_{t^*}(C) < w_{t^*}(C^*) = 0$. This however would mean that $\lambda(C) - t^*\tau(C) < 0$, or in other words that $W(C) = \frac{\lambda(C)}{\tau(C)} < t^*$, contradicting the assumed minimality of t^*. □

Time Bound. The pseudo-polynomial bound on the execution time comes about as follows. We define λ and τ on sets of edges by summation. Let τ_0 be the maximum value of τ over E. Let C_1 and C_2 be two cycles with distinct ratios. Then

$$\left| \frac{\lambda(C_1)}{\tau(C_1)} - \frac{\lambda(C_2)}{\tau(C_2)} \right| \neq 0$$

or

$$\left|\frac{\lambda(C_1)\tau(C_2) - \lambda(C_2)\tau(C_1)}{\tau(C_1)\tau(C_2)}\right| \neq 0 \tag{B.1}$$

Since the left hand side of equation B.1 is non-zero, and all the data are integer, the numerator must be at least 1 in absolute value. The denominator is at most τ_0^2. Thus on each iteration, the value of t must decrease by at least $\frac{1}{\tau_0^2}$.

Let λ_0 be the maximum absolute value of λ over E. Then again because the data is integral, the minimum ratio weight must lie in the interval $[-\lambda_0, \lambda_0]$. The algorithm therefore cannot iterate more than $2\lambda_0\tau_0^2$ times. Since on each iteration, the negative cycle detection algorithm has time bound $O(mn)$, the pseudo-polynomial bound on the time is $O(\lambda_0\tau_0^2 mn)$. In our case, the edge weights do not depend on the size of the graph, since they are related to the maximum image function value, which is independent of image size. The pseudo-polynomial bound is therefore polynomial in our case.

Negative Cycle Algorithm and Zero Cycle Detection. The negative cycle detection algorithm used in the algorithm was a dequeue implementation of a modified label-correcting algorithm for computing the shortest path lengths from a source vertex s to all $v \in V$. The generic label-correcting algorithm maintains labels d for each vertex. These are upper bounds on the shortest path lengths. It selects edges $e = \langle u, v\rangle$ one at a time and updates them if $d(v) > d(u) + w(e)$, where w is the edge weight function. The modified label-correcting algorithm instead removes vertices u from a list and updates the vertices v for which $\langle u, v\rangle \in E$. If v is not in the list, it is added. Both these algorithms are pseudo-polynomial [AMO93]. There is an $O(mn)$ implementation of the generic label-correcting algorithm that uses a queue as the list structure, adding updated vertices to the back. The dequeue implementation of the modified label-correcting algorithm is pseudo-polynomial but the fastest in practice, especially on sparse graphs such as lattices. In this version, the list is maintained as a dequeue. The vertices are always removed from the front of the queue but may be added to the front or the back. It is added to the front if it has been in the list earlier. Otherwise it is added to the back. The idea is that if v has been seen before, it will have updated some other vertices, its out-neighbours. If it is updated again, it is best to update these other vertices immediately, rather than first remove them from the list with old (and probably out of date) values, and update their out-neighbours, only to have to update those same out-neighbours a second time when v is eventually removed from the list. The time bound on this implementation is $O(\min(nmw_{\max}, m2^n))$, where $w_{\max}$ is the maximum absolute

value of w over E. Experiments show that in practice this implementation is approximately linear time.

There are several ways of detecting negative cycles while running these algorithms. If the label of a vertex slips below $-nw_{\max}$, then a negative cycle exists. One can also check whether the predecessor graph from each vertex has a cycle. If it does, this cycle must be negative, since no positive cycle can form part of the predecessor graph. This takes $O(n)$ time, and so does not slow down the algorithm if it is done every αn distance updates.

Finally, when the algorithm terminates, the way to find the zero length cycles is simply to adjust the edge weight for each edge $e = \langle u, v \rangle$ to $\tilde{w}(\langle u, v \rangle) = w(e) + d(v) - d(u)$, where d are the shortest path lengths computed by the label-correcting algorithm. Now a new graph G_0 is formed by removing all edges except those with $\tilde{w}(e) = 0$, along with disconnected vertices. Now cycles in G_0 correspond to zero length cycles in G with edge weights $\tilde{w}$. Finding cycles in G_0 is accomplished in the standard depth-first labelling fashion, looking for a back edge. In this way, it is possible to find degenerate minima.

There are even more efficient negative cycle detection algorithms based on a transformation to a matching problem. When $w_{\max}$ is polynomial in n, these offer a time-bound of $O(n^{1/2} m \log(n w_{\max}))$.

General Ratio Problem. As mentioned in the text, the approach used for minimum ratio weight cycles works for any problem of the following form. Given a set X, and two functions f and g, with g positive, find $x \in X$ that minimizes the ratio $f(x)/g(x)$. The principal result is that for combinatorial problems of size n (the number of variables), the minimum ratio weight problem can be solved in no more than $\log(n)$ ordinary optimization problems. Define $F = f - tg$. The bound depends on methods for finding the x with minimum F when F is positive, and for detecting an x with $F(x) < 0$ when it is not. If we can perform only the latter operation efficiently, then the ratio problem may require a polynomial number of such detection operations.

BIBLIOGRAPHY

[AB85] E. H. Adelson and J. Bergen, *Spatiotemporal energy models for the perception of motion*, J. Opt. Soc. Am. **2 (2)** (1985), 284–299.

[AB97] T. D. Alter and R. Basri, *Extracting salient curves from images: An analysis of the saliency network*, Int'l J. Comp. Vis. **27 (1)** (1997), 51–69.

[AMO93] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network flows: Theory, algorithms and applications*, ch. 5, pp. 133–165, Prentice Hall, NJ, U.S.A., 1993.

[Ana89] P. Anandan, *A computational framework and an algorithm for the measurement of vision motion*, Int'l J. Comp. Vis. **2** (1989), 283–310.

[AS95] S Ayer and H. S. Sawney, *Layered representation of motion video using robust maximum likelihood estimation of mixture models and MDL encoding*, Proc. Int'l Conf. Comp. Vis., 1995, pp. 777–784.

[ASZ99a] J. August, K. Siddiqi, and S. W. Zucker, *Computer vision and image understanding*, Computer Vision and Image Understanding **76 (3)** (1999), 231–243.

[ASZ99b] J. August, K. Siddiqi, and S. W. Zucker, *Contour fragment grouping and shared, simple occluders*, Computer Vision and Image Understanding **76 (2)** (1999), 146–162.

[Aya91] N. Ayache, *Artificial vision for mobile robots*, MIT Press, Cambridge, MA, U.S.A., 1991.

[BA91] M. Black and P. Anandan, *Robust dynamic motion estimation over time*, Proc. IEEE Conf. Comp. Vis. Patt. Rec., 1991, pp. 296–302.

[BB81] H. H. Baker and T. O. Binford, *Depth from edge and intensity-based stereo*, Proc. 7th Int'l J'nt Conf. Artif. Intell., August 1981, pp. 631–636.

[BCGJ95] R. Basri, L. Costa, D. Geiger, and D. Jacobs, *Determining the similarity of deformable shapes*, IEEE Workshop in Physics Based Vision, 1995.

[Bel96] P. N. Belhumeur, *A bayesian approach to binocular stereopsis*, Int'l J. Comp. Vis. **19** (1996), no. 3, 237–262.

[Bie85] I. Biederman, *Human image understanding: Recent research and a theory*, Computer Vision, Graphics and Image Processing **32** (1985), 29–73.

[Bla85] A. Blake, *Boundary conditions for lightness computation in a mondrian world*, Comp. Vis., Graphics and Im. Proc. **32** (1985), 314–327.

[Blu73] H. Blum, *Biological shape and visual science*, J. Theor. Biology **38** (1973), 205–287.

[BM92] P. N. Belhumeur and D. Mumford, *A bayesian treatment of the stereo correspondence problem using half-occluded regions*, Proc. Conf. on Computer Vision and Pattern Recognition (Champaign, IL, U.S.A.), 1992, pp. 489–495.

[BT99] S. Birchfield and C. Tomasi, *Multiway cut for stereo and motion with slanted surfaces*, Proc. Int'l Conf. Comp. Vis (Kerkyra, Greece), September 1999, pp. 489–495.

[BVZ98] Y. Boykov, O. Veksler, and R. Zabih, *Markov random fields with efficient approximations*, Proc. IEEE Conf. Comp. Vis. Patt. Rec., 1998, pp. 648–655.

[BZ87] A. Blake and A. Zisserman, *Visual reconstruction*, MIT Press, Cambridge, MA, U.S.A., 1987.

[CB97] R. Cipolla and A. Blake, *Image divergence and deformation from closed curves*, Int'l J. Robotics Research **16 (1)** (1997), 77–96.

[CBDMDB96] Y. Choquet-Bruhat, C. DeWitt-Morette, and M. Dillard-Bleick, *Analysis, manifolds and physics*, Elsevier Science, Amsterdam, The Netherlands, 1996.

[CHMR92] I. J. Cox, S. Hingorani, B. M. Maggs, and S. B. Rao, *Stereo without disparity gradient smoothing: A bayesian sensor fusion solution*, Proc. Brit. Mach. Vis. Conf. (D. Hogg and R. Boyle, eds.), Springer-Verlag, 1992, pp. 337–346.

[CRZ96] I. J. Cox, S. B. Rao, and Y. Zhong, *Ratio regions: a technique for image segmentation*, Proc. Int'l Conf. Patt. Rec., vol. 2, 1996, pp. 557–564.

[DBR66] G. B. Dantzig, W. O. Blatner, and M. R. Rao, *Finding a cycle in a graph with minimum cost to time ratio with application to a ship routing problem*, Theory of Graphs: International Symposium, Gordon and Breach, New York, 1966, pp. 77–84.

[DLR77] A.P. Dempster, N. M. Laird, and D. B. Rubin, *Maximum likelihood from incomplete data via the EM algorithm*, J. Roy. Stat. Soc. **39 (B)** (1977), 1–38.

[DP91] T. Darrell and A. Pentland, *Robust estimation of a multi-layered motion representation*, IEEE Workshop on Visual Motion, 1991, pp. 173–178.

[EZ93] J. Elder and S. W. Zucker, *The effect of contour closure on the rapid discrimination of two-dimensional shapes*, Vision Res. **33** (1993), 981–991.

[EZ94] J. Elder and S. W. Zucker, *A measure of closure*, Vision Res. **34** (1994), no. 24, 3361–3369.

[EZ96] J. Elder and S. W. Zucker, *Computing contour closure*, Proc. Euro. Conf. Comp. Vis., June 1996, pp. 399–412.

[Fau93] O. Faugeras, *Three-dimensional computer vision*, MIT Press, Cambridge, U.S.A., 1993.

[FJ90] D. Fleet and A. Jepson, *Computation of component image velocity from local phase information*, Int'l J. Comp. Vis. **5** (1990), 77–104.

[GG84] S. Geman and D. Geman, *Stochastic relaxation, gibbs distribution and the bayesian restoration of images*, IEEE Trans. Patt. Anal. Mach. Intell. **6** (1984), 721–741.

[GLY95] D. Geiger, B. Ladendorf, and A. Yuille, *Occlusions and binocular stereo*, Int'l J. Comp. Vis. **14** (1995), 211–226.

[GM96] G. Guy and G. Medioni, *Inferring global perceptual contours from local features*, Int'l J. Comp. Vis. **20** (1996), no. 1-2.

[Gol97] R. E. Goldstein, *Nonlinear dynamics of pattern formation in physics and biology*, Pattern Formation in the Physical and Biological Sciences (H. F. Nijhout, L. Nadel, and D. L. Stein, eds.), Addison-Wesley, Reading, U.S.A, 1997, pp. 65–91.

[Gri81] W. E. L. Grimson, *A computer implementation of a theory of human vision*, Phil. Trans. Roy. Soc. London **B 292** (1981), 217–253.

[HA89] W. Hoff and N. Ahuja, *Surfaces from stereo: Integrating feature matching, disparity estimation, and contour detection*, IEEE Trans. Patt. Anal. Mach. Intell. **11 (2)** (1989), 121–136.

[HAP94] S. Hsu, P. Anandan, and S. Peleg, *Accurate estimation of optical flow by using layered motion represenation*, Proc. 12th Int'l Conf. Patt. Rec., vol. A, 1994, pp. 743–746.

[Hee90] D. J. Heeger, *Optical flow using spatiotemporal filters*, Int'l J. Comp. Vis. **2** (1990), 181–190.

[HR85] D. D. Hoffman and W. A. Richards, *Parts of recognition*, Cognition **18** (1985), 65–96.

[HS81] B. K. P. Horn and B. G. Schunck, *Determining optical flow*, Artificial Intelligence **17** (1981), 185–203.

[IB94] S. Intille and A. Bobick, *Disparity-space images and large occlusion stereo*, Proc. Euro. Conf. Comp. Vis., 1994, pp. 179–186.

[IG98] H. Ishikawa and D. Geiger, *Occlusions, discontinuities and epipolar lines in stereo*, Proc. 5th Euro. Conf. Comp. Vis, 1998.

[Ish00] H. Ishikawa, *Global optimization using embedded graphs*, Ph.D. thesis, New York University, 2000.

[JB93] A. Jepson and M. Black, *Mixture models for optical flow computation*, Proc. IEEE Conf. Comp. Vis. Patt. Rec., 1993, pp. 760–761.

[JM92] D. Jones and J. Malik, *A computational framework for determining stereo correspondence from a set of spatial filters*, Proc. Euro. Conf. Comp. Vis. (Santa Maria Lighure, Italia), 1992.

[Jul71] B. Julesz, *Foundations of cyclopean perception*, University of Chicago Press, Chicago, U.S.A., 1971.

[Kan71] G. Kaniza, *Contours without gradients or cognitive contours*, Italian J. Psych. **1** (1971), 93–112.

[Kan79] G. Kaniza, *Organization in vision: Essays on gestalt perception*, Praeger, New York, U.S.A., 1979.

[Kar78] R. Karp, *A characterization of the minimum cycle mean in a digraph*, Dis. Math. **23** (1978), 309–311.

[KJ93] I. Kovács and B. Julesz, *A closed curve is much more than an incomplete one: Effect of closure in figure-ground segmentation*, Proc. Natl. Acad. Sci. USA **90** (1993), 7495–7497.

[KO90] T. Kanade and M. Okutomi, *A stereo matching algorithm with an adaptive window: Theory and experiments*, Proc. Image Understanding Workshop DARPA (PA, U.S.A.), September 1990.

[KTZ95] B. B. Kimia, A. R. Tannenbaum, and S. W. Zucker, *Shapes, shocks and deformations i: The components of two-dimensional shape and reaction-diffusion space*, Int'l J. Comp. Vis. **15** (1995), 189–224.

[KWT88] M. Kass, A. Witkin, and D. Terzopoulos, *Snakes: Active contour models*, Int. J. Comp. Vis. **1** (1988), no. 4, 321–331.

[Lan83] E. H. Land, *Recent advances in retinex theory and some implications for cortical computations: Color vision and the natural image*, Proc. Nat'l Acad. Sci. USA **80** (1983), 5163–5169.

[Law66] E. L. Lawler, *Optimal cycles in doubly weighted linear graphs*, Theory of Graphs: International Symposium, Gordon and Breach, New York, U.S.A., 1966, pp. 209–213.

[Ley88] M. Leyton, *A process grammar for shape*, Artificial Intelligence **34** (1988), 213–247.

[LG97] T.-L. Liu and D. Geiger, *Visual deconstruction: Recognizing articulated objects*, Int'l Workshop on Comp. Vis. Patt. Rec., Lecture Notes in Computer Science, vol. 1223, Springer-Verlag, 1997, pp. 295–309.

[LGK98] T.-L. Liu, D. Geiger, and R. V. Kohn, *Representation and self-similarity of shapes*, Proc. Int'l Conf. Comp. Vis., 1998.

[Liu97] T.-L. Liu, *Deformable object recognition with articulations and occlusions*, Ph.D. thesis, New York University, 1997.

[LK81] B. D. Lucas and T. Kanade, *An iterative image registration technique with an application to stereo vision*, Proc. 7^{th} Int'l Conf. Art. Intell., 1981, pp. 121–130.

[LM71] E. H. Land and J. J. McCann, *Lightness and retinex theory*, J. Opt. Soc. Amer. **61** (1971), 1–11.

[LM98] T. Leung and J. Malik, *Contour continuity in region based image segmentation*, Proc. Euro. Conf. Comp. Vis. (Germany), 1998.

[Low85] D. Lowe, *Perceptual organization and visual recognition*, Kluwer International Series in Engineering and Computer Science: Robotics and Vision, Kluwer Academic Publishers, 1985.

[Meg79] N. Meggido, *Combinatorial optimization with rational objective functions*, Mathematics of Operations Research **4** (1979), 414–424.

[MMP87] J. Marroquin, S. Mitter, and T. Poggio, *Probabilistic solutions of ill-posed problems in computational vision*, J. Am. Stat. Soc. **82 (397)** (1987), 76–89.

[MN78] D. Marr and H. K. Nishihara, *Representation and recognition of the spatial organization of three-dimensional shapes*, Proc. Royal Soc. Lond. **B 200** (1978), 269–294.

[MN85] G. Medioni and R. Nevatia, *Segment-based stereo matching*, Computer Vision, Graphics and Image Processing **31** (1985), 2–18.

[MP79] D. Marr and T. Poggio, *A computational theory of human stereo vision*, Proc. Roy. Soc. London **B 204** (1979), 301–328.

[MS85] D. Mumford and J. Shah, *Boundary detection by minimizing functionals*, Proc. Conf. Comp. Vis. Patt. Rec., 1985, pp. 22–26.

[MS89] D. Mumford and J. Shah, *Optimal approximations by piecewise smooth functions, and associated variational problems*, Comm. Pure Math. **15 (5)** (1989), 577–684.

[MT90] F. Makedon and S. Tragoudas, *Approximating the minimum net expansion: Near optimal solutions to circuit partitioning problems*, Proc. 16^{th} Workshop on Graph-Theoretic Concepts in Computer Science, Lecture Notes in Comp. Sci., vol. 484, Springer-Verlag, 1990, pp. 140–153.

[Mum91] D. Mumford, *Mathematical theories of shape: Do they model perception*, Proc. Conf. Geom. Meth. In Comp. Vis., SPIE Proceedings Series, 1991, p. 1570.

[Mum94] D. Mumford, *Elastica and computer vision*, Algebraic Geometry and Its Applications (Chandrajit L. Bajaj, ed.), Springer-Verlag, New York, U.S.A., 1994, pp. 491–506.

[NE86] H. Nagel and W. Enkelmann, *An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences*, IEEE Trans. Patt. Anal. Mach. Intell. **8** (1986), 565–593.

[PMF85] S. B. Pollard, J. E. W. Mayhew, and J. P. Frisby, *PMF: A stereo correspondence algorithm using a disparity gradient constraint*, Perception **14** (1985), 449–470.

[PSZ99] M. Pelillo, K. Siddiqi, and S. W. Zucker, *Matching hierarchical structure using association graphs*, IEEE Trans. Patt. Anal. Mach. Intell. **21 (11)** (1999), 1105–1120.

[PZ89] P. Parent and S. W. Zucker, *Trace inference, curvature consistency, and curve detection*, IEEE Trans. Patt. Anal. Mach. Intell. **11** (1989), no. 8.

[RC98] S. Roy and I. J. Cox, *A maximum flow formulation of the n-camera correspondence problem*, Proc. Int'l Conf. Comp. Vis. (Bombay, India), 1998.

[RDW86] W. Richards, B. Dawson, and D. Whittington, *Encoding contour shape by curvature extrema*, J. Optical Soc. Amer. **3** (1986), 1483–1489.

[RMG+76] E. Rosch, C. B. Mervis, W. D. Gray, D. M. Johnson, and P. Boyes-Braem, *Basic objects in natural categories*, Cognitive Psychology **8** (1976), 382–439.

[SB93] S. Sarkar and K. L. Boyer, *Integration, inference, and management of spatial information using bayesian networks: Perceptual organization*, IEEE Trans. Patt. Anal. Mach. Intell. **15** (1993), 256–274.

[Ser82] J. Serra (ed.), *Image analysis and mathematical morphology i & II*, Academic Press, 1982.

[SK95] K. Siddiqi and B. B. Kimia, *Parts of visual form: Computational aspects*, IEEE Trans. Patt. Anal. Mach. Intell. **17** (1995), 239–251.

[SK96] K. Siddiqi and B. B. Kimia, *A shock grammar for recognition*, Proc. Conf. Comp. Vis. Patt. Rec., 1996, pp. 507–513.

[SKT96] K. Siddiqi, B. B. Kimia, and K. J. Tresness, *Parts of visual form: Psychophysical aspects*, Perception **25 (4)** (1996), 399–424.

[SKTZ99] K. Siddiqi, B. B. Kimia, A. R. Tannenbaum, and S. W. Zucker, *Shapes, shocks and wiggles*, Image and Vis. Comp. J. **17** (1999), 365–373.

[SM97] J. Shi and J. Malik, *Normalized cuts and image segmentation*, Proc. IEEE Conf. Comp. Vis. Patt. Rec. (Puerto Rico), 1997, pp. 731–737.

[SM98] J. Shi and J. Malik, *Motion segmentation and tracking using normalized cuts*, Proc. Int'l Conf. Comp. Vis. (Bombay, India), January 1998, pp. 1154–1160.

[SSDZ99] K. Siddiqi, A. Shokoufandeh, S. J. Dickinson, and S. W. Zucker, *Shock graphs and shape matching*, Int'l J. Comp. Vis. **35 (1)** (1999), 13–32.

[SU88] A. Sha'ashua and S. Ullman, *Structural saliency: The detection of globally salient structures using a locally connected network*, Proc. Second. Int'l Conf. Comp. Vis. (Florida, U.S.A.), 1988.

[TSK97] H. Tek, P. A. Stoll, and B. B. Kimia, *Shocks from images: Propagation of orientation elements*, Proc. Conf. Comp. Vis. Patt. Rec., 1997.

[Tve77] A. Tversky, *Features of similarity*, Psych. Rev. **84** (1977), 327–352.

[TW96] K. K. Thornber and L. R. Williams, *Analytic solution of stochastic completion fields*, Biological Cybernetics **75** (1996), 141–151.

[Ull76] S. Ullman, *Filling in the gaps: The shape of subjective contours and a model for their generation*, Biological Cybernetics **75** (1976), 141–151.

[WA94] J. Y. A. Wang and E. H. Adelson, *Representing moving images with layers*, IEEE Trans. Image Processing Special Issue: Image Sequence Compression (1994), 625–638.

[WA96] Y. Weiss and E. H. Adelson, *A unified mixture framework for motion segmentation: Incorporating spatial coherence and estimating the number of models*, Proc. IEEE Conf. Comp. Vis. Patt. Rec., 1996, pp. 321–326.

[Wei97] Y. Weiss, *Smoothness in layers: Motion segmentation using non-parametric mixture estimation*, Proc. IEEE Conf. Comp. Vis. Patt. Rec., 1997, pp. 520–526.

[WJ97] L. R. Williams and D. W. Jacobs, *Stochastic completion fields: A neural model of illusory contour shape and salience*, Neural Computation **9 (4)** (1997), 837–858.

[WT98] L. R. Williams and K. K. Thornber, *A comparison of measures for detecting natural shapes in cluttered backgrounds*, Proc. Euro. Conf. Comp. Vis., June 1998, pp. 432–448.

[Yui89] A. Yuille, *Energy functions for early vision and analog networks*, Biological Cybernetics **61** (1989), 115–123.

[ZY96a] S. C. Zhu and A. Yuille, *Region competition: Unifying snakes, region growing, and Bayes/MDL for multi-band image segmentation*, IEEE Trans. Patt. Anal. Mach. Intell. **18** (1996), no. 9, 884–900.

[ZY96b] S. C. Zhu and A. L. Yuille, *FORMS: A flexible object recognition and modeling system*, Int'l J. Comp. Vis. **20 (3)** (1996).