Watching Pigs

John S. Zelek, Neil Bruce, Rekha Kanwar
School of Engineering, Univ. of Guelph
Guelph, ON, N1G 2W1, Canada.
jzelek@uoguelph.ca

Abstract

Animal health and welfare can be assessed by examining their behaviour, in particular their movements. In a novel application area, we explore using visual tracking, optical flow computations and attentional techniques for the purpose of monitoring a collection of pigs in a pen. In this paper we only explore the monitoring of a single pig. Techniques are presented that are found to perform at rates of 10 to 15 Hz on a non-dedicated Pentium 2 computer equipped with an inexpensive framegrabber. The results are encouraging, albeit preliminary. However, we envision that in the near future some useful tools will result from this work so that animal behaviourists can automatically collect data for performing non-invasive animal behaviour studies. We have yet to explore methods for recording and delivering the computed information in a form that ethologists can quickly ascertain an animal's health.

Introduction

Animal health and welfare can be assessed by examining their behaviour, in particular their movements. A lack of motion generally indicates a lack of welfare. Our focus for this study is focussed on pigs in a confined pig house, but the principals can be extended to observations of other animals in confined farm settings. Health can be ascertained by examining the number of times visiting a feeder, distance travelled, the social interaction of the pigs, etc. (Bigelow & Houpt 1987; Gonyou, Chapple, & Frank 1992). The use of image and visual analysis in the study of animals in a relatively new approach. An example application is the measurement of a pig's 2D body area from a top-down view, which is found to correlate with body weight (Brandl & Joergensen 1996). It is a great benefit to those that perform animal analysis to be able to monitor the animal's activities without disturbing them. The method proposed for applied ethological studies is the placement of a video camera and using computer vision techniques to analyze the streaming video information.

We have been recently engaged in research using inexpensive, off-the-shelf video cameras (i.e., typically ones sold in department stores for video conferencing applications) in the application of tracking humans for the purpose of providing perceptual capabilities for a mobile robot navigating, avoiding and interacting within a crowded room of people. We were able to acquire and process visual data in this configuration for tracking various features at rates of 10 to 20 Hz on a non-dedicated 333 MHz Pentium 2 machine. We were introduced to a completely different application area as a result of some contacts at our university with a researcher in Denmark who is examining and tracking the locomotion of pigs in a pen in order to analyze their social behaviour. The Danish researcher provided us with a 6 hour video tape of a collection of pigs in a pen. This paper reports on the application of some tracking and motion analysis tools from our current work to animal tracking and the direction for future investigations. The goal of the original people tracking experiments was to provide an inexpensive, fast (with real-time rates of at least 10 Hz) system that uses simple techniques yet is reasonably reliable.

The video data of the pigs in a pen was captured in a configuration consisting of a camera pointing downwards from the ceiling to the centre of a pigpen. *Change of scale* is not an issue because pig size in our configuration is invariant. Gray-level tracking as opposed to colour tracking was initiated because of the lack of colours in a pig pen (including the pig).

Our main goal was (and continues to be) to investigate the feasibility of automatically tracking pigs using simple tracking and motion capture techniques with an inexpensive off-the-shelf camera and framegrabber setup. Our objectives are that the resulting techniques be noise tolerant, be able to perform at a reasonable speed, be reasonably accurate and simple (Nishihara 1984). The first three criteria correspond to robustness, real-time performance and sufficient accuracy with respect to a particular task. The fourth criteria, simplicity, helps in analyzing the algorithm.

We explored simple tracking methods and optical flow techniques to examine if the behaviour of the pigs can be captured via these automated methods. The aspect of recording the information in a manner that can be analyzed by an animal behaviourist has yet to

Contact: John S. Zelek, jzelek@uoguelph.ca

be addressed. The main contributions of this work is not necessarily the techniques used but rather the application of a combination of existing computer vision techniques to the problem of non-invasive automated monitoring of the behaviour of pigs and other animals in order to ascertain and monitor their health. Current techniques used to gather this data involve manually observing pre-recorded video data or actually being physically present and observing and making notes on the pig's behaviours, a very tedious and labourious procedure. We have explored automating this procedure using blob tracking, optical flow and attentional methods.

Methods

Tracking

Visual tracking involves following an object around a scene. Most tracking methods rely on one of the following methods:

- *edge detection* (i.e., complex model and shape tracking (Baumberg & Hogg 1994)),
- *region-based correlation* techniques such as SSD (sum-squared difference methods), and
- *simple segmentation techniques* which are commonly referred to as *blob tracking.*

The edge tracking methods tend to be more robust and accurate but require higher computational cost. The *blob tracking* methods are more computationally efficient and work effectively in well structured environments (e.g., white on black). The primary goal of tracking is object localization which is not necessarily the same as achieving a complete segmentation, but the goal of minimizing false positives is essential.

We have experimented with tracking pigs using a blob tracking method as implemented as part of the XVision software package (Hager & Toyama 1998). The blob tracker thresholds each pixel value in the region of interest, and then uses the centre of mass of those points that are above the threshold to track the region. The blob tracker is initialized with a membership function that tests pixels for membership in the blob region by whether or not the pixel is between the lower and upper thresholds. Furthermore, the blob tracking only occurs if a minimum percentage of pixels in the region satisfies the membership requirement. The best performance will occur if you are tracking a black/white dot on a white/black background. No matter where the blob tracker is initialized, it will try to centre itself on a region containing 100% of the points that satisfy the membership criteria. For this reason, the best performance occurs when the blob region to be tracked can be contained entirely in the region of interest.

One of the problems with a simple blob tracker is its brittleness in complex environments. One can argue that our environment is very structured, i.e., the pigs are bright in intensity level when compared to the floor of the pig pen. This is the rationalization behind

(a)

(b)

(c)

Figure 1: **Three Successive Frames** from the video we used for performing our analysis.

using the blob tracker for tracking pigs in a pen. We anticipated having difficulties when the pigs are next to each other. In addition, the twisting of the pigs bodies should also give us difficulty. Others have extended the blob tracker to the colour domain (Wren *et al.* 1997; Rasmussen, Toyama, & Hager 1996) but the pig pen scenario is chiefly a gray-level image. For robustness, in the future, we expect to add additional visual cues to supplement the visual tracker because of the inherent brittleness of the simple blob tracker (Ramussen & Hager 1998).

Optical Flow

Optical flow is what results from the recovery of the 2-D motion field (i.e., the projection of the 3D velocity profile onto a 2-D plane; or the resulting apparent motion in an image). Most optical flow techniques assume that uniform illumination is present and that all surfaces are Lambertian. Obviously this does not necessarily hold in the real-world, but we assume that these conditions do hold locally. Optical flow describes the direction and speed of motion of features in the 2D image as a result of relative motion between the viewer and the scene. If the camera is fixed, the motion can be attributed to the moving objects in the scene. Optical flow also encodes useful information about scene structure: e.g., distant objects have much slower apparent motion than close objects.

The algorithm we used for computing optical flow information is a correlation-based technique that has been shown to exhibit real-time performance (Camus 1997). In this technique, the motion of a pixel at $[x, y]$ in one frame to a successive frame, is defined by the determined motion of the patch P_μ of μ by μ pixels centred at $[x, y]$, out of $(2n + 1) * (2n + 1)$ possible displacements, where n is an arbitrary parameter dependent on the maximum expected motion in the image over two successive image frames in a temporal sequence. The motion of the patch is simulated for each potential displacement of $[x, y]$ (given by n) and a match strength M is calculated for each displacement. Let ϕ represent a matching function. If I_1 is the first image examined and I_2 is the next successive image in a temporal sequence, then the match strength for a point $[x, y]$ for a simulated displacement (u, w) is calculated by:

$$\forall u, w : M(x, y; u, w) = \\ \sum \phi(I_1(i, j) - I_2(i + u, j + w)), (i, j) \in P_\mu \quad (1)$$

This can be efficiently calculated by taking into account certain redundancies in the calculation. Let m represent the simulation of all possible displacements of pixels $[x, y]$ between I_1 and I_2, and M be the function that results by applying ϕ (which is a smoothing operator) over m. The smoothing is only done over similar (u, w) displacements. In addition, since most averaging windows share common values amongst neighbours, this is taken into account when computing m and subsequently M to reduce the computational time.

Additional efficiency in the algorithm can be obtained by controlling the spatial or temporal sampling (Camus 1997). Spatial sampling is constrained by the size of the figure of interest and temporal sampling is constrained by the top speed of the figure of interest. It is also claimed (Camus 1997) that sampling at various resolutions and interpolating the results can also result in minimizing the effects of occlusion. This is not necessarily relevant in our case and will only be applicable if the pigs crawl all over each other. For assessing the computational efficiency of the optical flow algorithm, the image was subsampled (see results). The computational efficiency of the approach is at the expense of an increase in storage capacity. In particular (where s is a single dimension of a square image, $(2n + 1)^2$ elements define the square plausible pixel displacement region, and $(2\phi_n + 1)^2$ define the square smoothing window), s^2 elements are required for storing each temporal image (minimally two images), and for each pair of temporally displaced images the following is required: $(s - 2\phi_n)^2(2n + 1)^2$ elements are required for storing the simulated motion array; $(s - 2\phi_n)^2$ elements are required for storing temporary smoothing values; and $2(s - 2\phi_n)^2$ elements are also required for storing the resulting optical flow vector field. With regards to computation, the basic algorithm requires $(s - 2\phi_n)[(2n + 1)^2(s + 1) + 4s - 6\phi_n - 3]$ additions/subtractions. Typically both ϕ_n and n are relatively small when compared to s resulting in an algorithm complexity of $\bigcirc(s^2)$. We have not yet optimized the search process for finding the best motion describing a pixel motion. Our brute force search approach requires $(2n + 1)^2(s - 2\phi_n)^2$ comparisons. The motion field is finally presented after performing a quadratic interpolation amongst nearest neighbouring pixels resulting in an additional $16(2 - s\phi_n)^2$ additions/subtractions and $18(s - 2\phi_n)^2$ multiplications/divisions.

Matching techniques as described here typically perform worse than other methods (e.g., intensity-based differential, energy based, phase based (Barron, Fleet, & Beauchemin 1995)), especially in the case of diverging motion sequences (i.e., used to calculate time-of-collisions for robotic applications) which have a non-uniform motion field everywhere and violate the translational model assumed by the matching technique. Even though the algorithm (Camus 1997) that we have used performs slightly worse than other methods, it is significantly faster than other techniques. Diverging motion is not applicable and not of any concern for our application of pig monitoring given the configuration setup. Since the algorithm we used has comparable results to other techniques (Camus 1997), but at a significantly less computational cost, we found it an appropriate tool to use.

Attention

In order to initiate the tracking process, the objects of interest (i.e., pigs in our case) need to be firstly identified and segmented from the background. The easiest way to initiate this is to manually seed the tracking process by the user selecting a window size and its spatial location for the blob to be tracked.

Attention in essence is the localization of the object of interest. Tracking re-localizes based on this initial seed. Image segmentation is very costly and not necessarily feasible for real-time autonomous operation. Interest operators that have been used in the past include ones that maximize a measure of textured-ness or cornerness, such as a high standard deviation in the spatial intensity profile (Moravec 1980), the location of zero crossings in the Laplacian of the image intensity (Marr, Poggio, & Ullman 1979), and corners (Kitchen & Rosenfeld 1996). Another measure of textureness that optimizes its ability to be tracked (Shi & Tomasi 1994) was experimented with, but it also selected textureness in the floor of the pig pen which was of no interest to us (see experimental results).

An indirect yet interesting revelation while performing some of our experimentation with optical flow was that optical flow data could possibly be used to localize moving pigs and seed the tracking process. We are also interested in pigs that are stationary because this may indicate some illness especially if there are prolonged periods of inactivity. Thus, we plan to investigate other attentional methods that operate on static images (Reisfeld, Wolfson, & Yeshurun 1995).

Results

Dr. N. Brandl (from the Danish Institute of Agricultural Sciences) provided us with a six hour video clip of a collection of pigs in a pen (see Figure 1). The video was captured with a camera mounted on the ceiling, pointing downwards onto the pigs. We also assume that additional lighting was brought into the pen. For the sake of exploring the pigs' behaviour without interfering with what they are used to in terms of their artificial environment, it would be ideal not to introduce any additional lighting. We did not realize until later that the actual data was recorded at .3 Hz, indicating to us that maybe animals such as pigs do not need to be tracked with high frame rates. For the purposes of our experiments we assumed that the data was captured in real-time at 30 fps.

The data was analyzed on a non-dedicated, networked Pentium 2 333 Mhz PC running Linux using a *bt848*[1] *RAM*-less framegrabber (this is the type of *pci* framegrabber that usually comes packaged with the inexpensive video cameras found in stores). Albeit, the performance on this type of system can be improved if a large portion of contiguous physical memory is allocated for frame grabbing. This was not done for our experiments but can be done in the future for performance improvements.

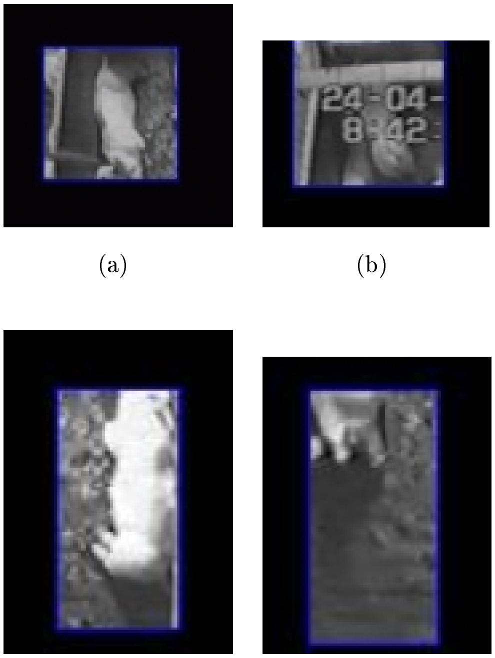
We were able to track individual pigs using the blob based method of the XVision package at a frame rate of approximately 15 Hz. The tracking window was seeded

[1] http://me.in-berlin.de/ kraxel/bttv.html

Figure 2: **Tracking an Individual Pig with Others Nearby.** Tracking an individual pig in the cluster illustrated in the figure is difficult with a simple square or rectangle blob window. The tracker tends to drift on and off other pigs. Additional information about the shape of the pig, identifiable by its contour, could be used to contain the blob.

manually by the operator placing a window with the pig centred. We experimented with different tracking windows (see Figure 3). A square tracking window includes lots of pixel information that is not necessarily associated with the pig (see Figure 3a) which tends to cause problems when other pigs are nearby (see Figure 2). Rectangle windows (see Figure 3c,d) that form a rectangular convex hull on the profile of the pig can also be used, but they are problematic if the pig changes orientation to the horizontal from the vertical or is angled in the scene (see Figure 4). As in the other case, if contour information about the pigs profile is used to contain the blob, better results should be anticipated. Another problematic situation was the superimposed date (see Figure 3b) around the video. This information tended to deviate the tracker away from the pig of interest. The blob tracker proved to be highly effective, given that it is a very simplistic technique, albeit for the situations described above. Defining the extent of the blob region to the silhouette circumscribed by the shape of the pig would definitely alleviate this problem. We are currently exploring fast effective ways of integrating shape information. Since the data we did use for performing our experiments was not recorded in real-time, we do not have an accurate feel for the speed in which pigs move at and thus we do not know what rate of frame processing is required.

We decided to also explore the avenue of calculating the optical flow of the pig video. It is not clear what information this can provide to the behaviorist and how it would be encoded but optical flow does encode the 2D projection of motion and may be a valuable resource. The algorithm for optical flow described earlier was examined with respect to its performance based on varying the input parameters: simulated motion window, averaging window, and the size of the image. The pixel



(a) (b)

(c) (d)

Figure 3: **Different Tracking Windows** (a) The tracker performs well when an individual pig is isolated. (b) The date superimposed on the image gave the tracker problems. (c) When nearby other pigs, the tracker drifts and finally (d) does not contain most of the pig's shape.

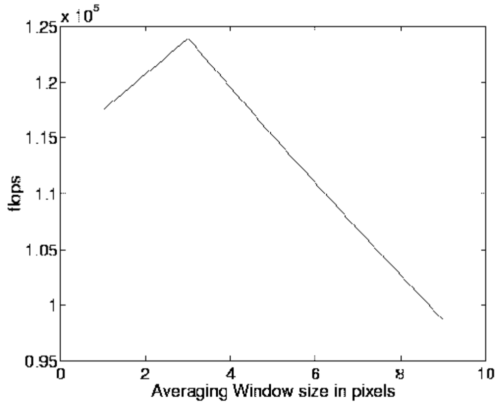
Figure 4: **Angled Body Position vs. Rectangular Window.** A rectangular window is not adequate for tracking the pig if it positions itself at an angle, leading to problems with the tracker not actually only tracking the pig, but also its environment.

averaging parameter (see Figure 5a) did not really affect the performance of the algorithm and this makes sense given that increasing the averaging window actually decreases the area of the image processed (i.e., the border region is not processed). Varying the window size which controls the spatial exploration of all potential motions between successive frames has a linear relationship with the amount of processing required (see Figure 5b). In addition, the algorithm is linearly proportional (i.e., $\bigcirc(n)$) to the number of image pixels (see Figure 5c) n, as indicated earlier. The lowest resolution was chosen based on the size of the figure of interest (i.e., the pig). This ensured that when the image was resampled, the motion of the pig was still distinguishable. This was found by re-sampling the image to 32 by 24 pixels. At this rate, it was found that the algorithm took only 1 msec to compare two successive images and produce the optical flow. Thus, the processing of the images was not the slowest processing component; the frame grabbing and image displaying were.

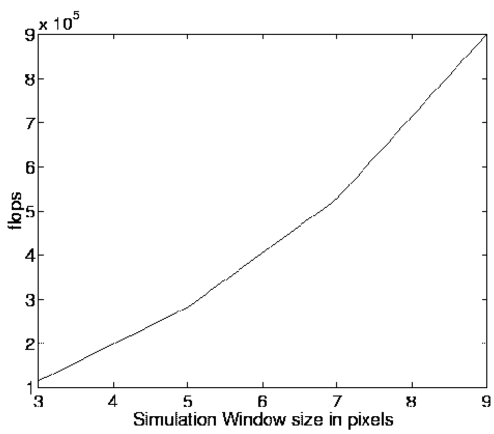
As mentioned earlier, the selection of a seed to initiate the tracker was done manually. We did perform some experimentation with selecting interest regions automatically. One example is using a texture measure (Shi & Tomasi 1994), as illustrated by Figure 9. This image illustrates that many other points were selected in addition to features of various pigs. Practically, the seeding module really needs to be a *pig detector*. This is an area of current investigation.

Discussions

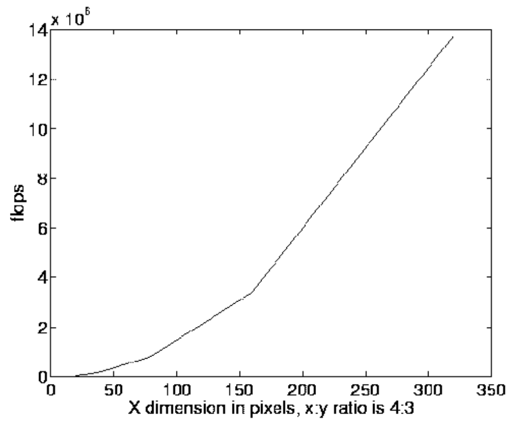
The results are preliminary but we have shown that the techniques presented do show some promise in capturing the behaviour of pigs automatically at real-time rates using inexpensive hardware. We found that the blob tracker would definitely be improved if pig shape information was integrated. This would have to be integrated without deteriorating performance speed. Given that our test data set was not recorded in real-time (i.e., 30 fps), we do not have an accurate feel for the maximum speed of a pig and do not know what our minimum frame rate should be. The test results were only given for tracking a single pig. Ideally the behaviorist would want to track many if not all the pigs in the pen. Tracking many pigs will slow down performance. Given that our tracker was able to perform at 10 to 15 Hz for tracking a single pig, and it was able to track data recorded at .3 Hz, we envision that the extension to many pigs may not necessarily be a problem. Optical flow data was found to be a possible detector for initiating the blob tracker but we are also interested in tracking stationary pigs because inactivity may be an indication of ill health. An attentional operator that initiates the tracking (i.e., a *pig detector*) is under current investigation.



(a)



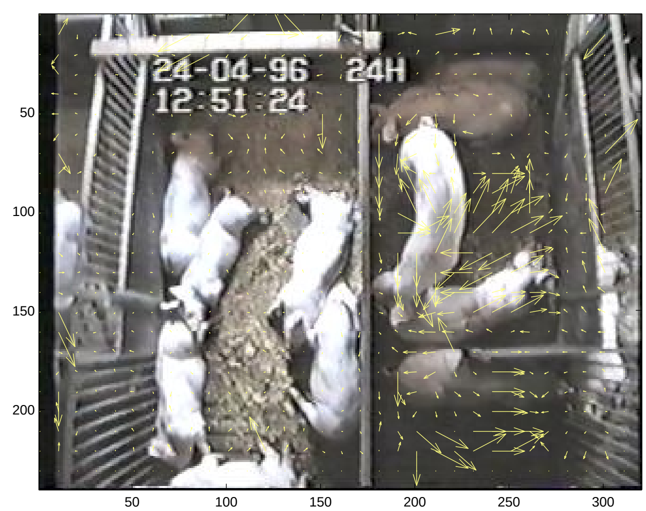
(b)



(c)

Figure 5: **Optical Flow Performance.** (a) Varying the average window size used for smoothing the image had negligible affect. Varying the (b) window used for simulating all potential motions and the (c) image size corresponded to a linear relationship.

(a)



(b)

Figure 6: **Optical Flow at Varying Resolutions** The original image was 320 by 240 pixels. (a) shows the optical flow between the first two images of the sequence in Figure 1 shown above at 32 by 24 resolution, while (b) shows the optical flow at 40 by 30 resolution. The only change in the image is the shift of the one pig in the right hand side of the image. Qualitatively, the results in (a) are comparable to (b).

(a)

(b)

Figure 7: **Optical Flow of Subsequent Sequence** initiated in Figure 6(b), which is temporally followed by the above (a) and (b).

(a)

(b)

Figure 8: **Continued Optical Flow Sequence** initiated in Figure 6(b), followed by Figure 7. (a), and (b), which is temporally followed by the above (a) and (b).

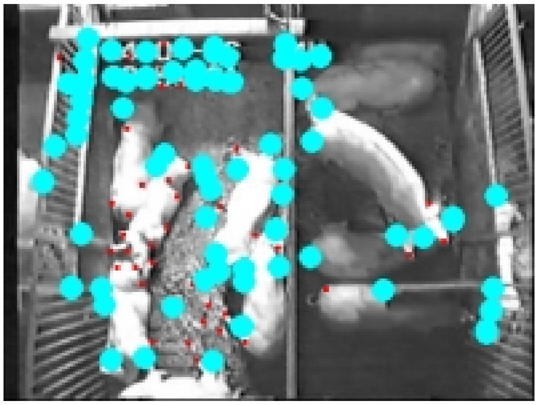


Figure 9: **Interest Points:** computed by the described texture measure are illustrated by the superimposed blobs.

Acknowledgements

The author would like to thank NSERC (National Science and Research Council) of Canada for funding the research in mobile robotics that initiated the application of the vision techniques used for this application area. The authors would also like to thank Dr. Nabil Brandl of the Danish Institute of Agricultural Sciences, Department of Animal Health and Welfare.

References

Barron, J.; Fleet, D.; and Beauchemin, S. 1995. Performance of optical flow techniques. *International Journal of Computer Vision* 12(1):43–77.

Baumberg, A., and Hogg, D. 1994. Learning flexible models from image sequences. In *1994 Euopean conference on computer vision*, 299–308.

Bigelow, J., and Houpt, R. 1987. Feeding and drinking patterns in young pigs. *Journal of Physiology and Behaviour* 43:99–109.

Brandl, N., and Joergensen, E. 1996. Determination of live weight in pigs using image analysis process. *Computers and Electronics in Agriculture* 15:57–72.

Camus, T. 1997. Real-time quantized optical flow. *Journal of Real-Time Imaging* 3:71–86.

Gonyou, H.; Chapple, R.; and Frank, G. 1992. Productivity, time budgets and social aspects of eating in pigs penned in group of five or individually. *Applied Animal Behaviour Science*.

Hager, G., and Toyama, K. 1998. The xvision system: A general-purpose substrate for portable real-time vision applications. *Computer Vision and Image Understanding* 69(1):23–37.

Kitchen, L., and Rosenfeld, A. 1996. Gray-level corner detection. Technical Report Technical Report, University of Maryland.

Marr, D.; Poggio, T.; and Ullman, S. 1979. Bandpass channels, zero-crossings, and early visual information processing. *JOSA* 69:914–916.

Moravec, H. 1980. *Obstacle avoidance and navigation in the real world by a seeing robot rover*. Ph.D. Dissertation, Stanford University.

Nishihara, H. 1984. Practical real-time imaging stereo matcher. *Optical Engineering* 23(5):536–553.

Ramussen, C., and Hager, G. 1998. Joint probabilistic techniques for tracking objects using multiple visual cues. In *1998 International Conference on Intelligent Robots (IROS'98)*.

Rasmussen, C.; Toyama, K.; and Hager, G. D. 1996. Tracking objects by color alone. Technical Report Technical Report, Dept. of Computer Science, Yale University, 51 Prospect Street, New Haven, CT 06520-8285.

Reisfeld, D.; Wolfson, H.; and Yeshurun, Y. 1995. Context free attentional operators: the generalized symmetry transform. *International Journal of Computer Vision* 14(2):119–130.

Shi, J., and Tomasi, C. 1994. Good features to track. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Wren, C.; Azarbayejani, A.; Darrell, T.; and Pentland, A. 1997. Pfinder: Real-time tracking of the human body. Technical Report Technical Report 353, MIT Media Laboratory, Boston, MA.