Université de Nice-Sophia Antipolis

Ecole Doctorale STIC

Sciences et Technologies de l'Information et de la Communication

THESE

pour l'obtention du grade de

Doctor en Sciences

de l'Université de Nice-Sophia Antipolis

Mention: Informatique

Algorithms & Perceptual Analysis for Interactive Free Viewpoint Image-Based Navigation

présentée par

Gaurav CHAURASIA

dirigée par George DRETTAKIS a Inria Sophia Antipolis

Février 18, 2013

Jury

Président	M. Frédéric PRECIOSO	*Université de Nice Sophia Antipolis*
Rapporteur	M. Christian THEOBALT	*Max-Planck-Institut für Informatik*
Rapporteur	M. Marcus MAGNOR	*Technische Universität Braunschweig*
Examinateur	M. Jean PONCE	*Ecole Normale Supérieure Paris*
Examinateur	M. Fredo DURAND	*Massachussets Institute of Technology*
Directeur de thèse	M. George DRETTAKIS	*Inria Sophia Antipolis*

Université de Nice-Sophia Antipolis

Ecole Doctorale STIC

Sciences et Technologies de l'Information et de la Communication

Dissertation

submitted in partial fulfillment for the degree of

Doctor of Science

of the Université de Nice-Sophia Antipolis

Specialization: Computer Science

Algorithms & Perceptual Analysis for Interactive Free Viewpoint Image-Based Navigation

presented by

Gaurav Chaurasia

under the supervision of George Drettakis at Inria Sophia Antipolis

Februrary 18, 2013

Thesis committee

President	Prof. Frédéric Precioso	*Université de Nice Sophia Antipolis*
Reviewer	Prof. Dr. Christian Theobalt	*Max-Planck-Institut für Informatik*
Reviewer	Prof. Dr. Marcus Magnor	*Technische Universität Braunschweig*
Examiner	Prof. Jean Ponce	*Ecole Normale Supérieure Paris*
Examiner	Prof. Fredo Durand	*Massachussets Institute of Technology*
Advisor	Dr. George Drettakis	*Inria Sophia Antipolis*

Contents

Acknowledgments

First of all, I thank the French government for the *allocation de recherche* (research fellowship) that financed most my doctoral studies.

I am most thankful to my advisor, George Drettakis, for his thorough involvement and patience. His incredible zeal for research and a near supernatural ability of fulfilling multiple deadlines at the same time has been a source of motivation. He pushed me towards goals which seemed unreasonable at the time, but later formed the backbone of this thesis. More often than not, he was the reason that kept complacency at bay.

I am grateful to Ravi Ramamoorthi for hosting me at the University of California Berkeley, and to Fredo Durand and Sylvain Paris for supervising me during my stay at Massachusetts Institute of Technology.

Though we did not collaborate, Adrien Bousseau had an important influence on my work. I attended his thesis defence as an audience member, and since then I have strived, with limited success, to match the quality of research he conducted during his doctoral studies.

I owe many thanks to all my co-authors, especially Olga Sorkine-Hornung for her guidance during the course of my main projects, and Peter Vangorp for his guidance on two other projects. I had the fortune of working with an incredibly capable group of students and researchers – Jonathan Ragan-Kelley, Sylvain Duchene, Ares Lagae, Bruno Galerne, Pierre-Yves Laffont, Emmanuelle Chapoulie, Rachid Guerchouche, Stefan Popov, Christian Richardt, Carles Bosch, Jorge Lopez-Moreno, Evanthia Dimara and Sylvain Lefebvre.

I must thank Sylvain Paris for repeatedly reviewing drafts of my papers during critical deadlines. I also thank Martin Eisemann, Michael Goesele and Marc Pollefeys for their advice on setting up comparisons with previous work.

I thank all the members of the jury: Christian Theobalt, Marcus Magnor, Fredo Durand, Jean Ponce and Frédéric Precioso for spending precious time on my dissertation.

Finally, I attribute some of the shortcomings of this thesis to my family, friends, flatmates and *la belle France* who gave me enough reasons to escape work every now and then.

Abstract

We present image-based rendering that allows free viewpoint walkthroughs of urban scenes using just a few photographs as input. Commercial applications such as Google Streetview, Bing Maps etc. use rudimentary forms of image-based rendering for urban visualization; more sophisticated approaches use the full 3D model of the scene as input. As the quality of 3D model degrades, rendering artifacts are observed which drastically reduce the utility of such applications. In this thesis, we propose image-based approximations to compensate for the lack of accurate 3D geometry. In the first approach, we use discontinuous image warping guided by quasi-dense depth maps which improves visual quality compared to previous methods that rely on texturing 3D models. This approach involves a small degree of manual intervention to mark occlusion boundaries in the input images. We build upon this in the second approach by developing a completely automatic solution that is capable of handling more complex scenes. We oversegment input images into superpixels and warp them independently using sparse depth. We introduce depth synthesis to create approximate depth in poorly reconstructed regions of the image and use this with our image warps for generating high quality results. We compare our results to many recent algorithms and show that our approach extends very well to free viewpoint navigation.

We also perform perceptual analysis of different image-based rendering artifacts in separate user studies under controlled conditions. We use vision science to investigate perspective distortions produced when a single image is projected on a planar geometry and viewed from novel viewpoints. We use the experimental data to develop a quantitative framework for predicting the level of perspective distortions as a function of capture and viewing parameters. In another study, we compare artifacts caused by smooth transitions (blending images) with abrupt transitions (popping) and develop guidelines for selecting the ideal tradeoff under different capture and rendering scenarios. We use guidelines from these studies to motivate the design of our image-based rendering systems described above.

We demonstrate an application of our approach for cognitive therapy. We create the first virtual reality application that uses image-based rendering instead of traditional computer graphics. This drastically reduces the cost of modeling 3D scenes for virtual reality while producing highly realistic walkthroughs.

Overall, we believe our work is a significant step towards free viewpoint image-based rendering designed on sound perceptually-based foundations.

Résumé

Nous présentons une approche de rendu à base d'images qui permet, à partir de photos, de naviguer librement et générer des points de vue quelconques dans des scènes urbaines. Les approches précédentes se basent sur un modèle géométrique complet et précis de la scène. La qualité des résultats produits par ces méthodes se dégrade lorsque la géométrie est approximative. Dans cette thèse, nous proposons une approximation basée sur l'image pour compenser le manque de précision de la géométrie. Dans une première approche, nous utilisons une déformation discontinue des photos guidée par des cartes de profondeur quasi-denses, ce qui produit de meilleurs résultats que le plaquage de texture utilisé par les méthodes précédentes, en particulier lorsque la géométrie est imprécise. Cette approche nécessite quelques indications utilisateur pour identifier les bordures d'occlusion dans les photos.

Nous proposons ensuite une méthode entièrement automatique basée sur la même idée de déformation d'image. Cette méthode permet de traiter des scènes plus complexes avec un plus grand nombre de photos. Nous évitons l'intervention utilisateur en sur-segmentant les images d'entrées pour former des superpixels. Nous déformons chaque superpixel indépendamment en utilisant l'information de profondeur clairsemée. Nous proposons également un algorithme de synthèse de profondeur approximative pour traiter les zones de l'image où la géométrie n'est pas disponible. Nous comparons nos résultats à de nombreuses approches récentes et montrons que notre méthode permet une navigation virtuelle libre.

Nous avons aussi étudié les défauts du rendu à base d'images d'un point de vue perceptif. Dans une première études controlées, nous avons évalué la perception des distorsions de perspective produites lorsqu'une seule image est projetée sur une géométrie planaire. Les données obtenues lors de cette étude nous ont permis de développer un modèle quantitatif permettant de prédire les distorsions perçues en fonction des paramètres de capture et de visualisation. Dans une autre étude nous comparons les défauts visuels produits par des transitions d'images douces ou abruptes. Nous avons déduit de cette étude des conseils pour choisir le meilleur compromis entre les deux types de transition. Ces deux études ont motivé des choix de conception de nos algorithmes de rendu à base d'images.

Enfin, nous démontrons l'utilisation de notre approche pour la thérapie cognitive, ce qui représente la première application de réalité virtuelle à base d'images. Notre méthode permet de réduire consid-

érablement le coût de modélisation 3D d'une scène de réalité virtuelle tout en produisant des visites virtuelles très réalistes.

Representative Publications

CABRAL, M., VANGORP, P., **CHAURASIA, G.**, CHAPOULIE, E., HACHET, M., and DRETTAKIS, G., 2011. A multimode immersive conceptual design system for architectural modeling and lighting. *IEEE Symposium on 3D User Interfaces (3DUI)*, 15–18.

CHAURASIA, G., SORKINE, O., and DRETTAKIS, G., 2011. Silhouette-aware warping for image-based rendering. *Comput. Graph. Forum (Proc. EGSR)*, 30(4):1223–1232.

VANGORP, P., **CHAURASIA, G.**, LAFFONT, P.Y., FLEMING, R.W., and DRETTAKIS, G., 2011. Perception of visual artifacts in image-based rendering of façades. *Comput. Graph. Forum (Proc. EGSR)*, 30(4):1241–1250.

CHAURASIA, G., DUCHENE, S., SORKINE-HORNUNG, O., and DRETTAKIS, G., 2013. Depth synthesis and local warps for plausible image-based navigation. *ACM Trans. Graph.*, 32(3):30:1–30:12. ISSN 0730-0301.

VANGORP, P., RICHARDT, C., COOPER, E.A., **CHAURASIA, G.**, BANKS, M.S., and DRETTAKIS, G., 2013. Perception of perspective distortions in image-based rendering. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 32(4):58:1–58:12. ISSN 0730-0301.

CHAPOULIE, E., GUERCHOUCHE, R., PETIT, P.D., **CHAURASIA, G.**, ROBERT, P., and DRETTAKIS, G., 2014. Reminiscence therapy using image-based rendering in VR. In Proc. IEEE Virtual Reality (short paper). To appear.

Chapter 1

Introduction

Computer graphics traditionally has four stages: (a) geometric modeling, (b) optional animation, (c) material/lighting design and (d) rendering. Over the last three to four decades, computer graphics has evolved immensely and can now achieve ultra-sophisticated special effects. One of the most important goals in computer graphics has been to generate results that are indistinguishable from reality. Looking at the state of the art in geometric modeling, computer animation and global illumination, it is fair to say that almost all natural and man-made phenomena can be simulated with astonishing levels of realism, given enough time and resources.

In effect, the process of modeling and rendering involves much manual as well as computational effort. Computational complexity of each of these stages can be an important factor, but one of the most significant bottlenecks is manual modeling and texturing which can be very expensive and time consuming. For example, suppose an application requires showing a generic urban town square. It will take a small group of artists days to model every detail of a typical square – the geometry, material properties, textures, lighting etc. using state of the art commercial software. An interesting aternative would be to acquire an existing scene and use the data to automate some or most of the modeling process. This basic intuition, coupled with the advent of digital cameras led to the conception of image-based approaches [Shum *et al.*, 2006] where handheld cameras serve as the acquisition device. Image-based approaches can be divided into two classes: those that attempt to synthesize new images from the same viewpoint but with different appearance, and those which attempt to change the viewpoint while keeping the same appearance. The former are known as *relighting* approaches while the latter are known as *image-based rendering* approaches. This thesis focuses on image-based rendering; the core problem statement is to "capture a scene in a few photographs and visualize it from a novel viewpoint in a region around the input viewpoints".

Conceptually, it would be possible to obtain physically correct renderings only if the scene can be reconstructed in full detail from the input photographs. To this end, appearance manipulation approaches require the illumination and material properties of the scene [Debevec, 1998; Laffont *et al.*, 2012] and viewpoint manipulation techniques require geometric reconstruction of the scene [Buehler *et al.*, 2001]. As discussed in Chapter 2, rapid advances have been made in the field of 3D reconstruction. However, perfect pixel accurate reconstructions are still very hard to obtain, more so if the scene is very complex. In the absence of perfect geometric reconstruction of the scene, it will be theoretically impossible to obtain physically correct rendering of the scene from novel viewpoints. Therefore, our target is to generate *plausible* results, to which end we investigate the following questions:

- How can we generate *plausible* free viewpoint navigation using image-based rendering?
- How can we define the notion of *plausibility* in the context of image-based rendering?

The former is *the* classical question that image-based rendering research has attempted to answer. Our work advances the state of the art by developing new techniques for image-based rendering in our specific context, which we explain in the following sections. One of the most important factors that distinguishes us from previous work is that we directly place ourselves in the context of *free viewpoint navigation*, while most previous approaches have targeted view interpolation where the target is to generate views that interpolate the positions and orientations of two input viewpoint.

1.1 Context

Being a data-driven approach, image-based rendering is sensitive to input data and deployment conditions. Therefore, it is important to clearly identify the use case for our systems. We target street-side (sub)urban imagery captured using handheld or vehicle mounted cameras (see Figure 1.1). Typically, such imagery contains architecture, vegetation, people, vehicles etc. We only handle static scenes, so we assume that there are no moving elements such as people or traffic in the photographs. The density of capture is typically 2 to 5 meters between different viewpoints. We intend to render these scenes from viewpoints oriented roughly in the same direction as the input viewpoint, within 30° of the average orientations of capture cameras. Our goal is to allow the user to navigate in a zone of 5 to 7 meters around any of the input viewpoints. Examples of novel camera paths can be seen in the result sections of Chapters 3 and 4, one such example novel view is shown in Fig. 1.1.

This is an increasingly relevant context since the successful deployment of virtual tourism [Snavely *et al.*, 2006] and street-side visualization [Kopf *et al.*, 2010] as commercial systems like Microsoft Photosynth, Google Streetview etc. Currently these systems are restricted to transition between viewpoints; the displayed result is faded from one image to the next. The addition of *free viewpoint* image-based rendering to these systems will allow the user to have a much more powerful immersive experience.

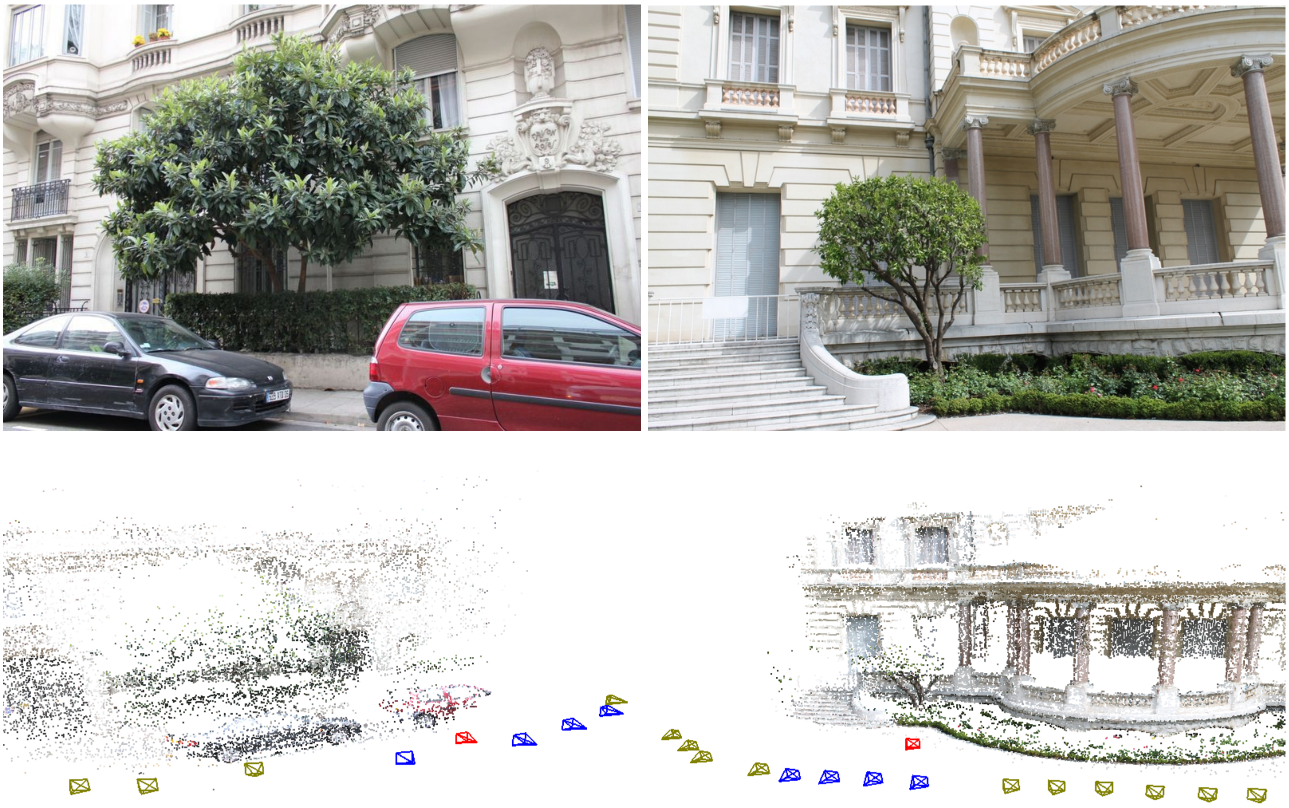


Figure 1.1: Examples of urban imagery used for image-based rendering. The top row shows one input images and bottom row shows the top view of the scene with placeholders for input viewpoints. The novel viewpoint, shown in red, lies far from the path interpolating the input viewpoints, which we refer to as *free viewpoint*. The input viewpoints closest to the novel viewpoint are shown in blue.

1.2 Problem statement

Recent advances in computer vision and graphics make it possible to take 10-20 photographs, use automatic camera calibration [Snavely *et al.*, 2006] and multi-view stereo to obtain depth/disparity maps [Goesele *et al.*, 2007; Furukawa and Ponce, 2009], and then use surface reconstruction [Kazhdan *et al.*, 2006; Fuhrmann and Goesele, 2011] to obtain a 3D model. The resulting 3D geometry or *proxy* can be rendered by reprojecting the input photographs onto the proxy and blending closest views [Buehler *et al.*, 2001]. This is indeed a powerful approach that generalizes to our context which can be expected to give perfect results if each stage of the pipeline provides perfect output.

In practice, the above pipeline has several limitations. First, multi-view stereo approaches have difficulty producing 3D geometry of sufficiently good quality for foreground objects with complex shapes such as trees, or sharp depth discontinuities such as vehicles parked in front of façades, or poorly textured objects such as walls, or busy textures such as vegetation. Such situations are very frequent in urban scenes which are our main focus. Consequently, image-based rendering approaches that rely on 3D models can suffer from artifacts for such scenes. Second, most methods have been developed only for small displacements between input viewpoints and only handle view interpolation [Zitnick and Kang, 2007; Fitzgibbon *et al.*, 2005; Mahajan *et al.*, 2009; Stich *et al.*, 2011]. Urban visualization has to be scalable to very large scenes. Therefore, baselines between input viewpoints are expected to be large. Moreover, view interpolation restricts the utility of image-based rendering because it is somewhat equivalent to a video.

Our central problem statement is to develop new image-based approaches that demonstrate previously unattempted levels of sophistication – free view navigation, while assuming sparser resources in the form of input images, for more complex scenes where preprocessing stages, namely multi-view stereo, are not expected to give completely accurate results. We distinguish ourselves from previous work by pursuing a harder set of challenges:

- urban scenes in their full complexity,
- minimal captures with wide displacements of up to 5-7 meters between input views,
- complex disocclusion effects due to irregular silhouettes of multiple foreground objects,
- plausible occlusion and parallax in spite of poor quality 3D reconstruction in many regions, and
- free viewpoint navigation.

The secondary problem is to quantify image-based rendering artifacts. There are almost no metrics for quantifying rendering quality of different image-based rendering approaches. This is because of the sheer number of factors that affect the final result, e.g. scene complexity, number of input images, simulated view positions etc. We target perceptual analysis of rendering artifacts; to which end we develop experimental setups, stimuli and protocols for principled studies and demonstrate the utility of such studies to image-based rendering setups.

1.3 Main intuitions

Multi-view stereo algorithms generate a 3D point cloud of varying levels of accuracy and density depending upon scene content. The densely reconstructed regions are typically planar regions with sufficient structured texture. Other regions can have a much smaller set of reconstructed samples. Current techniques for surface extraction [Kazhdan *et al.*, 2006; Fuhrmann and Goesele, 2011] and plane fitting [Sinha *et al.*, 2009; Gallup *et al.*, 2010] perform very well for densely reconstructed regions and completely ignore the other regions, either estimating them as blobs or merging them with some dominant plane. Since, these approaches are global optimizations, they often tend to ignore small clusters of 3D points on poorly reconstructed objects because such regions get outweighed by other well reconstructed objects. As a result, rendering artifacts ensue in such regions.

The main intuition is to actively utilize *all* the 3D points obtained from multi-view stereo, even the smallest clusters on poorly reconstructed objects. We delineate different depth layers of the scene using silhouettes and use the 3D points in each region separately. Typically, depth at any pixel allows previous approaches to reproject the pixel into a novel viewpoint. We assume that very few pixels in an image region have depth; we reproject the entire image region into the novel viewpoint by using a *shape-preserving warp*. Our warps are guided by the small number of pixels which have depth and regulated by image-based 2D constraints which seek to minimize the overall distortion in the final result. This high level idea is achieved by different means in Chapters 3 and 4.

Another important idea that contributes heavily to the success of our approaches is that we enforce silhouettes in an image-based manner. Previous approaches depend upon silhouettes being appropriately represented in the reconstructed geometry [Eisemann *et al.*, 2008]. Plane fitting approaches [Gallup *et al.*, 2010] improve this by using graph cuts to reinforce image edges into the fitted planes. These graph cuts can be thought of as joint optimization on image edges and 3D planes which can always cause planes to bleed into erroneous regions if no geometry is available over a significant region or simply due to numerical issues. Intuitively, decoupling silhouettes from geometry estimation will always perform better in terms of accurate silhouettes. We compute image silhouettes in a preprocess and use them to isolate the shape preserving warp of different regions, resulting in high quality occlusions and parallax effects.

Overall, we demonstrate that our *modus operandi* of formulating the whole problem in terms of constraints that target plausible image synthesis is highly effective at compensating for errors in reconstruction.

For perceptual analysis, we enumerate some of the most important artifacts and perform user studies that allow us to correlate perceived visual quality with scene and rendering parameters. The most important insight in the design of perceptual experiments is to isolate the artifacts using simplified setups that allow principled analysis, while remaining sufficiently close to actual image-based rendering setups that are interesting from an application as well as research perspective. Our setups allow us to control the degree of artifacts using a small number of parameters in the stimuli; user studies under these conditions give a direct relationship between rendering parameters and perceived quality.

We start with the study of perspective distortions, which are inherent in any method that reprojects an image captured from one viewpoint into another viewpoint. We then study the more complex case of visual artifacts created by blending content from multiple input images to synthesize any pixel of the novel view. We use the guidelines from these studies using different methodologies in each of Chapters 3 and 4.

1.4 Contributions

High quality image-based rendering The main contribution of this thesis is in the form of image-based rendering algorithms designed for urban environments that are capable of producing high quality results in the absence of accurate 3D reconstruction. It is well-known that multi-view stereo methods [Goesele *et al.*, 2007; Furukawa and Ponce, 2009] can produce impressive results for architecture but their performance degrades on cluttered scenes. Our approach is in line with the recent trend of image-based rendering systems that use 3D point clouds produced by multi-view stereo directly [Goesele *et al.*, 2010; Sinha *et al.*, 2012; Kopf *et al.*, 2013] rather than expecting a 3D mesh which can be extremely hard to obtain for cluttered scenes, especially urban imagery containing vegetation, vehicles, architecture etc.

Free viewpoint navigation This is the first research work in image-based rendering to actively propose free viewpoint navigation where the novel or simulated camera is allowed to navigate quite far from the input viewpoints in the scene. Almost all previous approaches, the earliest to the very latest, have only addressed view interpolation [Chen and Williams, 1993; Fitzgibbon *et al.*, 2005; Zitnick and Kang, 2007; Mahajan *et al.*, 2009; Goesele *et al.*, 2010; Stich *et al.*, 2011; Sinha *et al.*, 2012; Kopf *et al.*, 2013]. This is an important issue because free navigation exposes the true advantage of such systems by allowing a scene captured with just a few photographs to be visualized in rich details in a variety of applications, one of them being head-tracked virtual reality systems, an early prototype of which is also demonstrated in Chapter 6.

Perceptual analysis The classic evaluation method for image-based rendering has always been viewer opinion. Some approaches used image statistics [Fitzgibbon *et al.*, 2005]; however such metrics are suitable when the approach is expected to produce physically-correct results. Most image-based rendering systems target *plausible* or *good looking* results, with some approaches using non-photorealistic effects [Goesele *et al.*, 2010]. The only way to evaluate such systems is by means of perceptual studies. This thesis proposes perceptual analysis of visual artifacts where simple rendering and input data setups are used to isolate the artifacts. The data from the studies allows us to correlate the severity of artifacts with scene or rendering parameters.

1.5 Current and potential applications

The latest versions of commercial products like Bing Maps show massive reconstructed urban areas (see Figure 1.2). This degree of 3D information is sufficient for early experiments with our image-based rendering approaches. It is clear from Figure 1.2 that the 3D information is sufficient to render large structures very well but details such as trees etc. are represented as blurry blobs. The approaches from

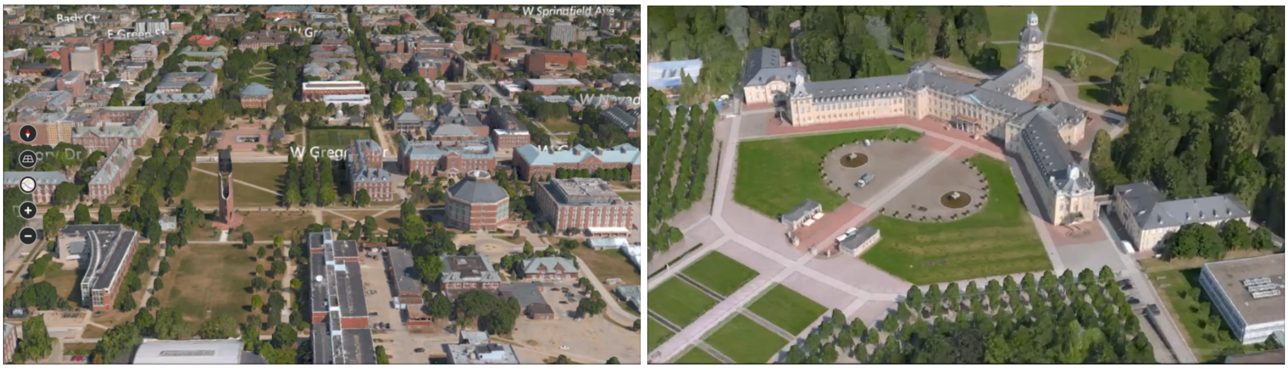


Figure 1.2: Latest version of Bing Maps suggests a fair amount of 3D information is available for complete cities.

Chapters 3 and 4 are designed to work with quasi-dense 3D information and can therefore be applied to these commercial systems to produce high quality walkthroughs at very large scales.

Apart from these large scale systems, image-based rendering has the potential of being useful for any computer graphics application which seeks to visualize existing objects or scenes. There are examples in other branches of computer science where data-driven approaches have greatly simplified workflows which were otherwise completely manual, for example motion capture [Liverman, 2004] which is now considered an indispensable tool for animators. Similarly, it is wasteful to force artists to model existing scenes which can instead be acquired very easily. Acquisition followed by 3D reconstruction, using commercial software such as Autodesk 123D[1], is likely to suffice for some of these applications, especially those which deal with closed objects. Image-based rendering has a role to play for all applications which require open scenes where the process of converting point clouds or depth maps into accurate 3D models is much harder.

We present an example of one such application in the context of virtual reality in Chapter 6. We use image-based rendering to model urban scenes and use virtual walkthroughs in these scenes for Reminiscence Therapy. Examples of other applications can be lightweight games, *quick and dirty* 3D modeling for virtual reality or simulator backdrops etc.

1.6 Overview

The rest of the thesis is organized as follows:

- Chapter 2 gives a discussion on previous work in computer graphics, vision, geometry and perception that is relevant to the techniques described in this thesis.
- Chapter 3 presents a novel image-based rendering approach based on variational image warps that is capable of handling some of the most complicated test cases attempted.

[1] http://www.123dapp.com/catch

- Chapter 4 builds upon the previous approach by developing a local variational warp based on image oversegmentation. This is among the very first techniques to present free viewpoint interactive navigation using image-based rendering.
- Chapter 5 describes perceptual analysis of visual artifacts associated with image-based rendering systems.
- Chapter 6 describes an virtual reality setup using image-based rendering. Though still in early stages of development, this is the very first system of its kind.
- Chapter 7 summarizes the results of this thesis and proposes immediate next steps as well as long term research avenues.

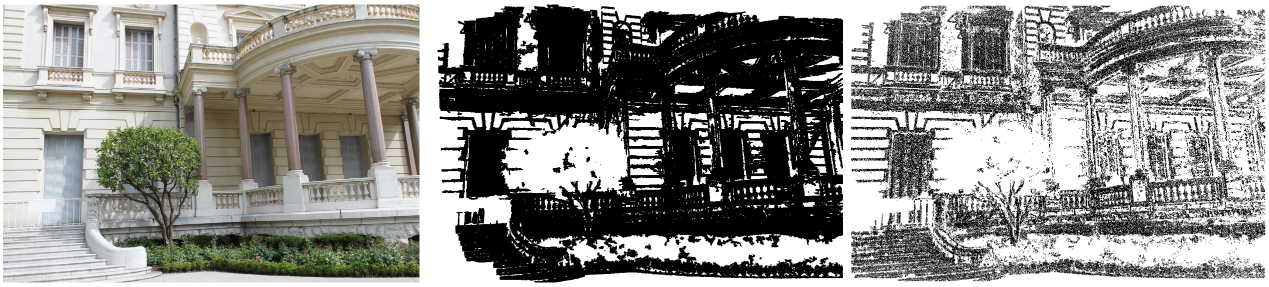
Chapter 2

Previous Work

Image-based rendering has been an active area of research since its inception in the form of image interpolation [Chen and Williams, 1993] and plenoptic modeling [McMillan and Bishop, 1995]. It began as a approach for viewpoint manipulation, which has matured into general spatio-temporal novel view synthesis. Over the last two decades it has borrowed from and inspired research in various branches of computer graphics and vision, while spawning a number of commercial applications like Google Streetview, Microsoft Photosynth etc.

The earliest approaches such as plenoptic modeling [McMillan and Bishop, 1995], light fields [Levoy and Hanrahan, 1996], lumigraph [Gortler *et al.*, 1996] and view dependent texture mapping [Debevec *et al.*, 1996], were self-contained. They did not require any preprocessing. As the complexity of scenes increased, the use of 3D geometry became prevalent [Buehler *et al.*, 2001; Eisemann *et al.*, 2008] because it helped reduce the number of input images while improving robustness towards occlusions. Since then, image-based rendering has been associated with 3D reconstruction and other computer vision techniques related to geometry estimation.

The overall goal of this thesis is to generate image-based rendering results for urban scenes where the main requirements are (a) simple capture setup using handheld cameras, and (b) free viewpoint walkthroughs, where free viewpoint means that the novel viewpoints may not be on a path joining input viewpoints. As we show in later sections, multi-view stereo suffers from artifacts as the complexity of input scenes grows. Image-based rendering approaches tend to use the 3D geometry as the only constraint to reproject input images into target viewpoints. This can lead to a variety of rendering artifacts. Our intuition is to compensate for the lack of 3D geometry by using image-based approximations. We reproject input images using 3D geometry as a soft constraint which is regulated by shape-preserving constraints that are inspired by image warping applications. Moreover, looking at the importance of silhouettes in plausible view synthesis, we extract silhouettes using image segmentation rather than depending upon depth maps or 3D models to provide accurate object boundaries. This again serves to



(a) Input image (b) [Goesele *et al.*, 2007] (c) [Furukawa and Ponce, 2009]

Figure 2.1: Multi-view stereo. Depth maps for input image in extracted using [Goesele *et al.*, 2007] and [Furukawa and Ponce, 2009]. Reconstructed regions are shown in black while unreconstructed regions are shown in white. Note the uneven distribution of depth samples and complete lack thereof in some regions.

compensate for the lack of correct silhouette localization in 3D reconstructions.

Given the above, we see that our work draws from multiple research domains – stereo reconstruction, image-based rendering, image warping and image segmentation. In this chapter, we discuss the state of the art in each of these domains.

2.1 3D reconstruction

Despite impressive advances in recent years, state of the art 3D reconstruction can still give inaccurate results which can produce rendering artifacts when used in existing image-based rendering frameworks [Buehler *et al.*, 2001; Eisemann *et al.*, 2008; Sinha *et al.*, 2009; Goesele *et al.*, 2010]. We discuss recent advances in multi-view stereo and their limitations in this section.

Stereo reconstruction has been one of the most active domains of research in computer vision. Seitz *et al.* [2006] present a comprehensive overview of the early work in multi-view stereo reconstruction. Most of early research in 3D reconstruction focused on isolated objects such as statues. Snavely *et al.* [2006] opened the doors to reconstruction of vast open urban scenes. Here we review only the recent developments which are relevant to image-based rendering.

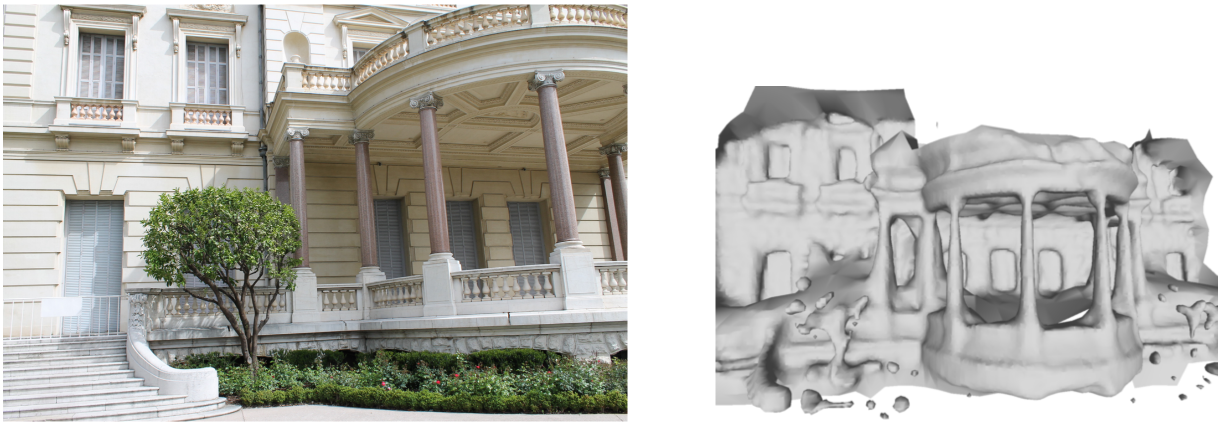
Multi-view stereo The development of large scale structure-from-motion [Snavely *et al.*, 2006] has provided a stable solution to the long standing problem of automatic marker-less camera calibration for large unordered imagery. This development has greatly advanced multi-view stereo research allowing researchers to experiment with a wide variety of datasets i.e. outdoors, indoors, community photo collections etc. and not just separate objects. Goesele *et al.* [2006] use plane-sweep stereo for estimating depth maps for each input view. Sinha *et al.* [2007] use a volumetric graph cut to estimate full 3D geometry of an object. While the results of all these approaches are compelling, their most important limitation is that they are designed for closed objects that have been photographed from all sides. They need to be initialized with a bounding box and do not give good results for unbounded scenes such as urban imagery.

The above limitation has inspired large scale multi-view stereo systems. Goesele *et al.* [2007] present a multi-view stereo approach for large community photo collections from the internet. Pollefeys *et al.* [2008] present a real time approach for very large scenes captured using vehicle mounted cameras similar to Google Streetview. Patch-based multi-view stereo [Furukawa and Ponce, 2009] matches feature points between input images, estimates their depths and uses these to estimate depths of neighboring patches. Extensions of [Furukawa and Ponce, 2009] have been used to reconstruct city level reconstructions in [Agarwal *et al.*, 2009, 2010] from hundreds of thousands of photographs. Other approaches such as [Labatut *et al.*, 2007; Hiep *et al.*, 2009] estimate depth of a large number of interest points followed by Delaunay triangulation; the final 3D model is generated by computing an inside-outside cut on the tetrahedra resulting from the Delaunay triangulation. The results are denser than [Furukawa and Ponce, 2009] and seem closest to those provided by the commercial solution Autodesk 123D[1]. We use [Furukawa and Ponce, 2009] to recover a 3D point cloud from input images; the choice of multi-view stereo algorithm is not critical and can be replaced by [Goesele *et al.*, 2007; Hiep *et al.*, 2009] or commercial products like Autodesk 123D or Acute3D Smart3DCapture[2].

While significantly different in implementation, these multi-view stereo approaches are fairly similar in principle. They match image features between images in unorganized photo collections and estimate their depth. They then use this depth to initialize depth estimation for neighboring image patches. The result of these multi-view stereo approaches is thus fairly similar. As shown in Figure 2.1, these approaches give very good results for regular structures like façades. However, the quality is much worse for texture-poor regions, busy textures and irregular geometry etc. The distribution of reconstructed points or depth samples is highly irregular and very sparse in some regions. The localization of silhouettes is also inaccurate on many scene objects.

Surface extraction from 3D point clouds The 3D point clouds computed by multi-view stereo have to be converted into polygon meshes in order to be rendered as continuous solid objects. These approaches, known as *surface reconstruction*, can be classified into two types - reconstruction from unorganized point clouds and techniques that use underlying structure in point cloud data. Prominent examples of the former include [Hoppe *et al.*, 1992], Moving least squares [Levin, 1998], Point set surfaces [Alexa *et al.*, 2001] and Poisson surface reconstruction [Kazhdan *et al.*, 2006]. These approaches reconstruct a *watertight mesh* which is not appropriate for *open* urban scenes. However, this problem

[1] http://www.123dapp.com/catch
[2] http://www.acute3d.com/smart3dcapture/



(a) Input image

(b) [Kazhdan *et al.*, 2006]

Figure 2.2: Surface reconstruction. (a) Input image, (b) Poisson surface reconstruction [Kazhdan *et al.*, 2006] on 3D point cloud extracted from [Furukawa and Ponce, 2009]. The quality of 3D models is far from perfect for complex scenes which manifests as artifacts in image-based rendering.

can be resolved by manually removing spurious triangles from the 3D mesh that these approaches add to create watertight meshes.

The second class of algorithms exploit structure and are known as depth map fusion techniques [Fuhrmann and Goesele, 2011]. The 3D point clouds obtained from depth maps are inherently structured because any two neighboring pixels in a depth map give two connected points in 3D space. These do not suffer from the watertight surface assumption but require almost pixel dense depth maps. These approaches are not ideal for our experiments because depth maps for the scene we intend to treat can be erroneous (see Figure 2.1).

As shown in Figure 2.2, the 3D mesh obtained from Poisson surface reconstruction [Kazhdan *et al.*, 2006] are far from perfect. The irregular density of 3D points and the complexity of the underlying geometry of the scene make surface reconstruction a very hard problem. The errors in 3D models manifest as rendering artifacts when used with image-based rendering approaches such as [Buehler *et al.*, 2001; Eisemann *et al.*, 2008].

We use Poisson surface reconstruction [Kazhdan *et al.*, 2006], as recommended by Furukawa and Ponce [2009], to create 3D models for use in other image-based rendering approaches [Buehler *et al.*, 2001; Eisemann *et al.*, 2008] for the sake of comparisons. We experimentally observed that, however erroneous (see Figure 2.2), it gave the best 3D models among existing approaches, which makes for fair comparisons.

Piecewise-planar reconstruction Some techniques exploit the fact that most man-made structures are piecewise planar and use this prior to directly generate planar geometry, thereby circumventing surface reconstruction altogether. These approaches compute the 3D point cloud and plane fitting in

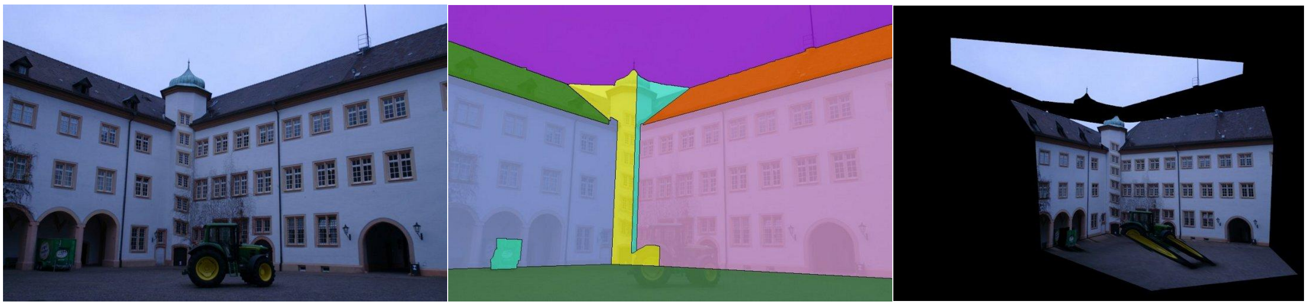


Figure 2.3: Piecewise-planar reconstruction [Sinha *et al.*, 2009]. (a) Input image, (b) extracted planes, and (c) reprojection of geometry in a novel view. Clearly, the foreground tractor gets merged with the background façade and results in rendering artifacts.

two separate steps [Furukawa *et al.*, 2009; Gallup *et al.*, 2010] or as a joint optimization [Mičušík and Košecká, 2009; Sinha *et al.*, 2009]. Furukawa *et al.* [2009] use manhattan-world priors and generate axis-aligned 3D planes to approximate the scene geometry. Sinha *et al.* [2009] use a general piecewise-planar prior to generate 3D planes; which is extended in Mičušík and Košecká [2009]; Gallup *et al.* [2010] to street-level imagery captured using vehicle mounted cameras similar to Google Streetview. The planar geometry generated by all of these approaches is very compact since it only consists of a small number of planes.

Apart from the obvious restriction that some scene geometry may not be piecewise planar, the main problem with these approaches is that they tend to merge poorly reconstructed foreground regions with dominant background planes (see Figure 2.3). Clearly, these approaches would fall short of handling cases with poorly reconstructed geometry as shown in Figure 2.1.

Discussion It is clear that state of the art 3D reconstruction can give erroneous results for complex scenes. The depth maps or 3D point clouds can be very sparse for some regions that contain no texture or very busy texture, and 3D model generation can be extremely hard for open urban scenes in the presence of foreground clutter, especially vegetation. Image-based rendering in the absence of 3D geometry can suffer from various rendering artifacts. We design our image-based rendering approaches in Chapters 3 and 4 to compensate for these limitations by using plausible depth synthesis in poorly reconstructed regions and image warps. We compare our results to image-based rendering approaches based on point clouds [Goesele *et al.*, 2010] as well as 3D models [Buehler *et al.*, 2001; Eisemann *et al.*, 2008].

2.2 Image-based rendering

Since the seminal work on plenoptic modeling [McMillan and Bishop, 1995], many image-based rendering algorithms have been developed, such as light fields [Levoy and Hanrahan, 1996] and unstructured lumigraphs [Buehler *et al.*, 2001] among many others. A number of interesting applications have

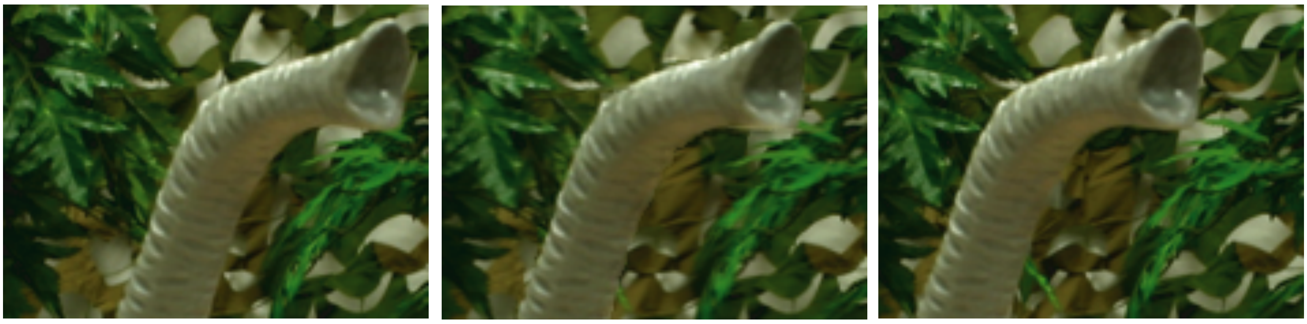


Figure 2.4: Image interpolation without 3D geometry [Mahajan *et al.*, 2009]. Interpolated image (middle) generated from input images (left and right). Note that the baseline between the input image is of the order of just a few pixels, which is one of the most important limitations of such methods.

resulted from this work, e.g., camera stabilization [Liu *et al.*, 2009], video enhancement [Gupta *et al.*, 2009] and commercial products like Google Streetview. This field has been studied in a number of different contexts using a wide variety of approaches which have very different inputs and targets. We classify these approaches by their input data requirements, highlighting why certain classes are unsuitable for our context while others prove insufficient because of restrictive priors or algorithmic constraints.

Image interpolation without 3D geometry A number of approaches use image morphing without explicitly reconstructing 3D geometry for novel view synthesis. Chen and Williams [1993] generate novel views by interpolating dense optical flow between input images. Light fields [Levoy and Hanrahan, 1996] place input cameras on a 2D grid and interpolate views by parameterizing the light rays using a 4D representation – two coordinates each for intersection of light rays on the camera and focal planes. Seitz and Dyer [1996] generate target view on the line joining the optical centers of two input views assuming no occlusions. Lhuillier and Quan [1999] match points and regions between images using a quasi-dense matching algorithm and generate novel views by interpolating matched regions. This is further improved by using constrained triangulation to preserve straight edges in the images [Lhuillier and Quan, 2000]. A detailed analysis of sampling issues for early techniques can be found in [Shum and Kang, 2000]. Pixel correspondences computed using epipolar geometry are used in [Schirmacher *et al.*, 2000] to improve the rendering quality of the lumigraph [Gortler *et al.*, 1996]. A high-quality approach for interpolating two images is presented in [Mahajan *et al.*, 2009]; they produce crisp results by using graph cut to create seamless transitions. Stich *et al.* [2011] present perceptually correct image interpolation; they partition input images into homogeneous regions, match these regions and compute a separate perspective transformation for each region. This serves as a correspondence field between images without explicitly reconstructing 3D geometry. Lipski *et al.* [2010] use the correspondence field estimated by [Stich *et al.*, 2011] and view morphing [Seitz and Dyer, 1996] for spatio-temporal image

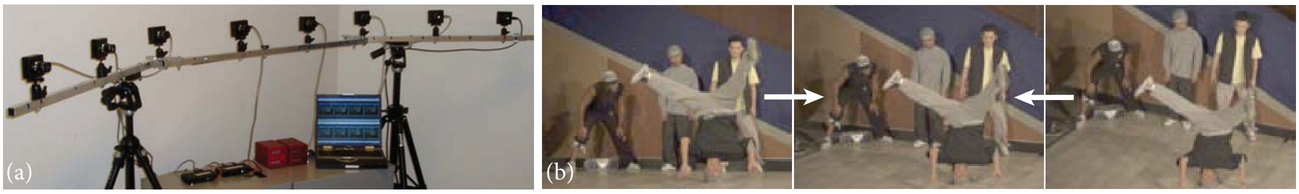


Figure 2.5: Joint stereo and image-based rendering. (a) Studio capture setup used in Zitnick *et al.* [2004]; Zitnick and Kang [2007]. (b) Input images are shown on left and right and interpolated view shown in the middle. These approaches are limited to view interpolation over fairly small baselines.

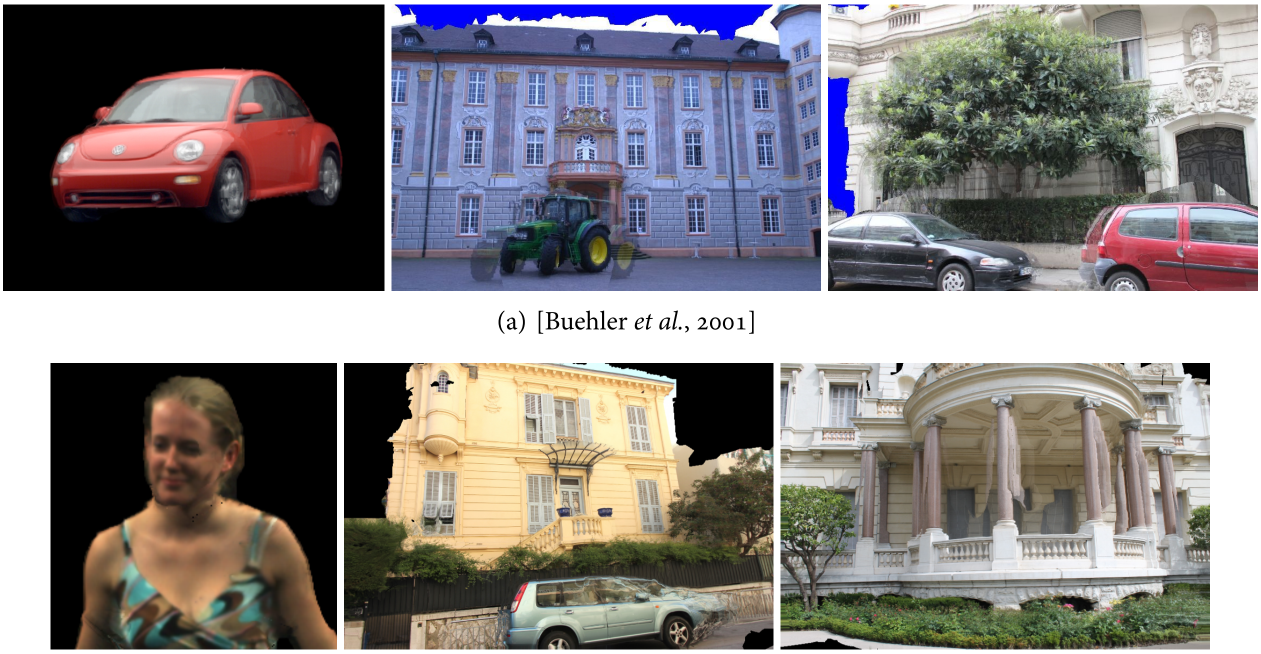
interpolation; the results are used in the demo *Who Cares*[3].

These approaches are powerful and robust but only handle small baselines between images. Mahajan *et al.* [2009] report a 30-pixel maximum baseline. Being largely oblivious to 3D geometry, they have difficulty handling occlusions and are also strictly restricted to view interpolation. These are critical restrictions in our context; handling wide baselines and occlusions are two most important problems. As a result, these approaches prove largely insufficient in our scenario.

Joint reconstruction and image-based rendering Many approaches estimate depth/disparity between image pairs with the sole purpose of image-based rendering. Zitnick *et al.* [2004] compute disparities between pairs of images using Markov random field priors and interpolate them using a layered representation built on top of disparity maps. This is adapted in [Zitnick and Kang, 2007] to work with image segmentation which provides silhouettes. Fitzgibbon *et al.* [2005] use image-based priors to preserve structure in synthesized views in a computationally expensive optimization. Hornung and Kobbelt [2009] extract view-dependent depth maps directly from input images and merge them in real time using a median filter on the GPU. They are designed to interpolate stereo pairs or structured studio captures using capture rigs as shown in Figure 2.5. These approaches exploit the prior knowledge that two cameras are neighbors, and treat multi-view datasets in a pairwise fashion. They seem to be tightly restricted to small baseline studio captures and view interpolation. They are largely based on stereo reconstruction, and can be expected to suffer from the same problems as multi-view stereo (see Section 2.1) if used for unorganized wide baseline urban imagery.

Image-based rendering using 3D geometry Most modern approaches use explicitly computed 3D geometry in order to handle large baselines between input images and free viewpoint image based rendering. The lumigraph [Gortler *et al.*, 1996] was the first approach to suggest use of coarse 3D geometry compared to plenoptic modeling [McMillan and Bishop, 1995] and light fields [Levoy and Hanrahan, 1996]. View dependent texture mapping [Debevec *et al.*, 1996] uses a user-created model of simple architecture and projective texturing to produce compelling walkthroughs. Sillion *et al.* [1997] used

[3] http://graphics.tu-bs.de/projects/whocares/



(a) [Buehler *et al.*, 2001]

(b) [Eisemann *et al.*, 2008]

Figure 2.6: Image-based rendering using 3D geometry. The leftmost images are results taken directly from the respective papers. Middle and right images are results on other datasets generated using our implementation of [Buehler *et al.*, 2001] (see Section 3.6 for details) and authors' implementation of [Eisemann *et al.*, 2008]. Ghosting and misalignment are clearly visible when the proxy is not accurate as shown in these examples.

image-based impostors instead of textured meshes for low level of detail rendering in urban scenes. Heigl *et al.* [1999] present plenoptic modeling that uses 3D geometry of the scene in the form of a 3D plane. These approaches use a drastically smaller number of input images than previous approaches such as light fields [Levoy and Hanrahan, 1996]. With the development of multi-view stereo [Furukawa and Ponce, 2009], automatically reconstructed point clouds and 3D models have replaced manually modeled proxies [Debevec *et al.*, 1996] or single 3D planes [Heigl *et al.*, 1999].

Unstructured lumigraph [Buehler *et al.*, 2001] is a generalized image-based rendering framework. It computes the color of target pixels by backprojecting them on to the 3D geometry and reprojecting into the input views. Contribution from multiple input images is blended using weights computed using relative distances between centers of projection of cameras. The main advantages are that it allows input images to be taken in an arbitrary manner and could extend to free viewpoint walkthroughs. This approach indeed works very well given perfect geometry assuming no occlusions. Floating textures [Eisemann *et al.*, 2008] introduce occlusion handling and a correction pass based on optical flow that alleviates blending artifacts due to inaccurate geometry. The main limitation is that the occlusion handling assumes accurate silhouettes in the 3D proxy and optical flow correction is limited to misalignment of up to 10-15 pixels. Hauswiesner *et al.* [2011] present a real time visual hull computation for dynamic scenes; their image-based rendering approach is similar to [Buehler *et al.*, 2001], adapted for time evolving 3D geometry. In other relevant work, Aliaga *et al.* [2003a,b] present walkthroughs of indoor scenes; they compensate for inaccurate geometry by using a very large number of input images. Ambient point clouds [Goesele *et al.*, 2010] is a view interpolation approach that uses a non-photorealistic rendering style in poorly reconstructed regions. Sinha *et al.* [2012] present image-based rendering for reflective surfaces by reconstructing two depth layers for reflections, which is further improved in the form of a gradient-domain approach [Kopf *et al.*, 2013]. Both of these approaches compute pixel-dense depth maps to interpolate between images. Bhat *et al.* [2007] also use a gradient domain approach to transfer details from a small number of high resolution photographs to a large number of low definition video frames. They use 3D geometry in the form of sparse set of 3D reconstruction to register the photographs and video frames. Other techniques have been developed to browsing video archives [Ballan *et al.*, 2010; Tompkin *et al.*, 2012]; these approaches transition between video streams using 3D reconstruction of the scene. The goal is to provide a smooth transitions from one video stream to another rather than plausible novel views.

Discussion The development of the above approaches which use 3D geometry has entwined image-based rendering with multi-view stereo. These approaches when combined with best possible reconstruction of the input scenes using either of [Goesele *et al.*, 2007; Pollefeys *et al.*, 2008; Furukawa and Ponce, 2009] represent state of the art in image-based rendering systems. We use this combination to show comparisons in Chapters 3 and 4. The inability of 3D reconstruction to produce perfect depth maps or accurate silhouettes or accurate 3D meshes combined with the inability of rendering pipelines to successfully compensate for these errors means that even the most sophisticated image-based rendering system would fall short of our target of rendering the complex urban scenes using as few images as possible in a free viewpoint walkthrough scenario.

2.3 Image warping

A majority of image-based rendering approaches [Buehler *et al.*, 2001; Eisemann *et al.*, 2008; Sinha *et al.*, 2009] use 3D geometry as the only constraint for reprojecting an input image to a novel viewpoint. We seek to compensate for insufficient 3D by using image-based approximations. To this end, we warp input images to novel viewpoints using 3D geometry as a soft constraint which is regulated by other 2D constraints which seek to prevent distortions in the final result. The mathematical tools we use are inspired by image warping applications which allow users to deform an input image in a variety of ways. The challenge for these applications is to generate content-aware warps, manipulating different parts of the input image in different ways without introducing visible discontinuities, deformations or other artifacts, synthesizing plausible images which appear just as consistent as photographs. This has led to

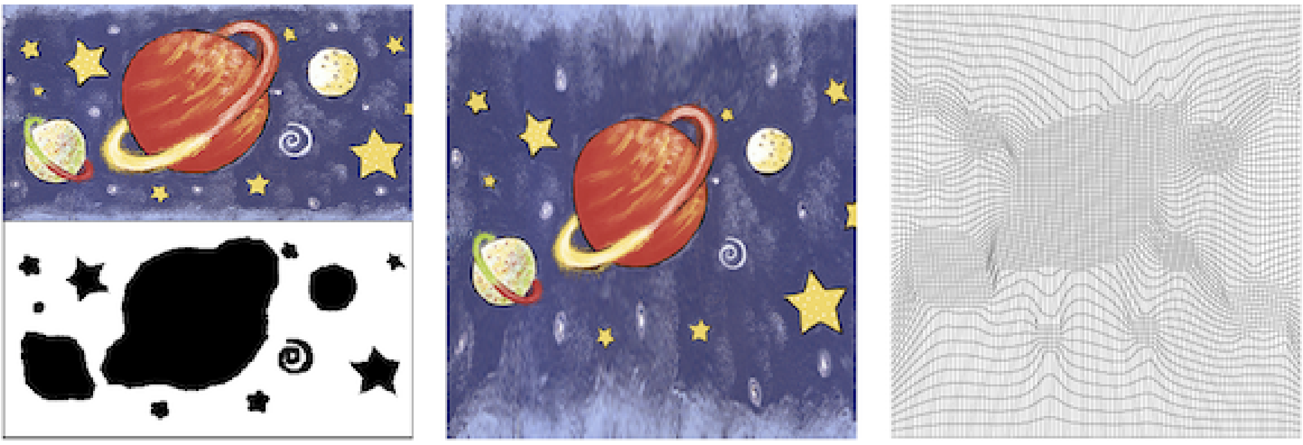


Figure 2.7: Variational warps [Gal *et al.*, 2006] for retargeting. The input image (upper left) is warped to produce the result (middle). The salience mask (lower left) is used to impose constraints on the vertices of a regular warp mesh, which get deformed to the final underlying warp mesh show on the right.

a variety of applications such as retargeting, view morphing, image interpolation etc.

Retargeting using variational warping Manipulating the aspect ratio of images changes the aspect ratio of the content and leads to deformations. Image retargeting approaches try to preserve the aspect ratio of salient image content while shifting most of the deformation to regions which are either unimportant or make it hard to perceive the distortions. Apart from the seam carving approach for retargeting [Avidan and Shamir, 2007], almost all other approaches [Gal *et al.*, 2006; Wang *et al.*, 2008; Zhang *et al.*, 2009; Panozzo *et al.*, 2012; Chang and Chuang, 2012] have used variational warping (see Figure 2.7). A comparative study is presented in [Rubinstein *et al.*, 2010].

These approaches have also been adapted to video retargeting [Wolf *et al.*, 2007; Krähenbühl *et al.*, 2009; Wang *et al.*, 2010] and panoramic imagery [He *et al.*, 2013]. Among related approaches, Carroll *et al.* [2009, 2010] present image warps that allow the user to change the perspective of input images incorporating a variety of user specified constraints like vanishing points, line segments, line orientations, planar regions and fixed points.

These approaches provide the basic mathematical tools for variational image warping. All of them overlay uniform triangle or quad meshes on the input image and compute the warp by means of linear [Gal *et al.*, 2006; Wang *et al.*, 2008; Zhang *et al.*, 2009] or non-linear [Carroll *et al.*, 2009, 2010] optimization. They all have (a) one or more fundamental guiding constraints e.g. resized image boundaries in case of retargeting approaches, (b) a set of regularization constraints which preserve the structure of the warp mesh e.g. rigid transform [Gal *et al.*, 2006], and (c) some optional constraints to preserve specific aspects of the image e.g. line constraints in [Carroll *et al.*, 2009]. The warped image can be synthesized by rendering the warped mesh using the original texture coordinates for each vertex. This mathematical framework is referred to as a *variational image warp*. In some cases, per-pixel mapping between

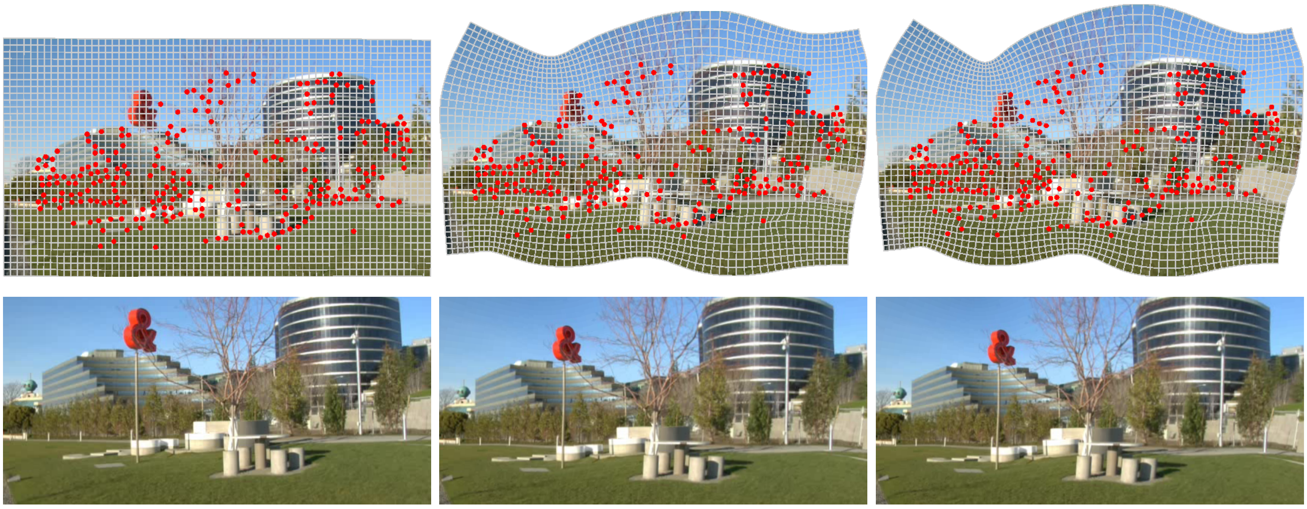


Figure 2.8: 3D image warping [Liu *et al.*, 2009]. The input image (left) is warped into different novel views (middle and right). Top row: warped images showing the underlying warp mesh and the 3D points in red. Bottom row: final image after cropping the warped result.

input and target image may be available from optical flow or per-pixel depth [Didyk *et al.*, 2010]; image warping then reduces to simply mapping every pixel to its target location. In contrast, variational warps seek to minimize an energy function that warps all pixels of an input using a much sparser pixel to pixel mapping. This mathematical framework is also the basis of image warping for other applications as discussed below.

3D image warping In contrast to the 2D warping constraints in all of the above techniques, 3D image warps are guided by sparse/quasi-dense 3D geometry and morph an input image to another viewpoint. The motivation comes from 2D shape manipulation [Igarashi *et al.*, 2005; Schaefer *et al.*, 2006] where the user animates a 2D sketch by pulling a small number of handles and shape deformations are minimized by rigid/conformal/similar/affine constraints. 3D image warping replaces the user handles with 3D reconstructed points or depth samples which can be reprojected into arbitrary viewpoints. Liu *et al.* [2009] use this idea for 3D camera stabilization by warping each video frame from the original viewpoint to a viewpoint on a stabilized camera trajectory (see Figure 2.8). They do not handle occlusions, however this is not a major problem in this context because video frames are warped to viewpoints in close vicinity.

The most important ingredient of 3D image warping is regularization in the form of rigid [Igarashi *et al.*, 2005], affine [Schaefer *et al.*, 2006] or similarity [Liu *et al.*, 2009, 2013] constraints. These approaches demonstrate that a very sparse set of *guiding constraints*, e.g. user handles [Igarashi *et al.*, 2005] or 3D points [Liu *et al.*, 2009], can be successfully compensated by these regularization constraints which mask perceivable deformations. However, occlusion handling and warping over larger baselines are still open problems.

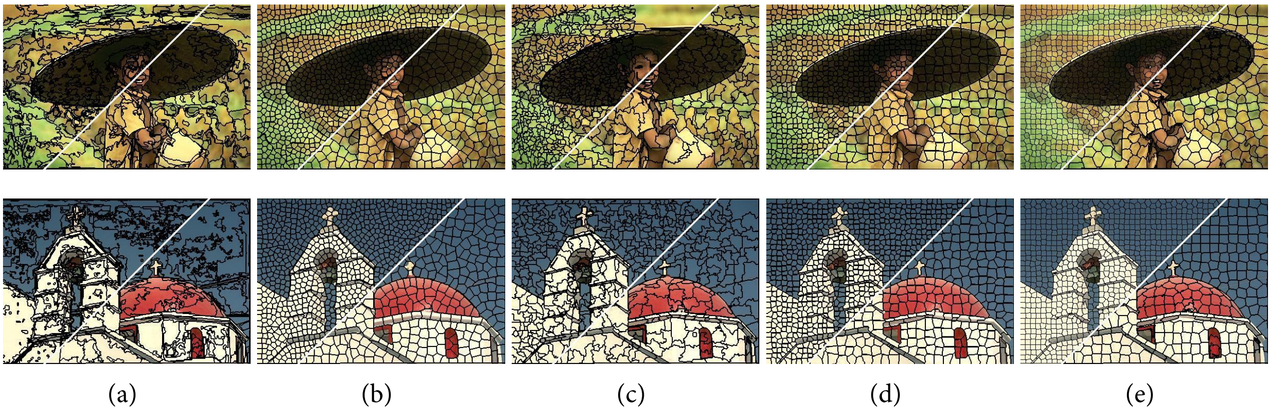


Figure 2.9: Comparison of different oversegmentation algorithms (a) [Felzenszwalb and Huttenlocher, 2004], (b) [Mori, 2005], (c) [Vedaldi and Soatto, 2008], (d) [Levinshtein *et al.*, 2009], and (e) [Achanta *et al.*, 2012].

Discussion The above discussion shows that variational warps are capable of generating plausible warped images using a sparse mapping between input image pixels and target image pixel. In Chapters 3 and 4, we demonstrate *shape-preserving image warping* based on variational warping, that preserve image silhouettes, provide occlusion handling and are robust over wide-baselines between input and target viewpoints.

2.4 Image segmentation

One of the important problems in image-based rendering is occlusion handling and silhouettes. As discussed previously, 3D reconstruction does not always produce perfect results at silhouettes. On the other hand, purely image-based approaches are much better at extracting silhouettes. The state of the art in image segmentation [Hoiem *et al.*, 2007b; Maire *et al.*, 2008; Arbelaez *et al.*, 2011] can classify a very wide variety of objects. However, all image classification approaches are based on machine learning and cannot be expected to segment all objects perfectly especially if an image has multiple prominent objects. This can lead to under-segmentation and parts of objects can be miss-classified. Appendix A compares image classification approaches and shows failure cases.

Oversegmentation Image oversegmentation is the process of dividing the image into hundreds of small regions of homogeneous image content called *superpixels*. Superpixels capture *all* the silhouettes while also producing a large number of redundant boundaries. In contrast, machine learning based approaches can often miss silhouettes in some regions. The redundant boundaries produced by oversegmentation are undesirable, yet benign. The guarantee that all silhouettes are captured by superpixels is an important advantage.

The seminal work on superpixels [Shi and Malik, 1997] used spectral analysis of an $n \times n$ matrix, where n is the number of pixels in the image. This quickly becomes very slow for even medium sized images (around 2 megapixels). Subsequent work on oversegmentation [Ren and Malik, 2003; Felzenszwalb and Huttenlocher, 2004; Mori, 2005; Vedaldi and Soatto, 2008; Levinshtein *et al.*, 2009] improve the computational efficiency and allow for more user control over size and shape of superpixels. State of the art oversegmentation SLIC [Achanta *et al.*, 2012] can segment large images in a few seconds while producing regularly shaped superpixels as shown in Figure 2.9. A detailed comparison of oversegmentation algorithms can be found in [Achanta *et al.*, 2012]. We use SLIC for oversegmentation in Chapter 4.

It is important to note that superpixels form the basis of many image classification and recognition algorithms because they capture all prominent silhouettes in an image. We use superpixels for the same reason. Other applications of oversegmentation include view interpolation [Zitnick and Kang, 2007; Stich *et al.*, 2011], depth estimation [Cigla *et al.*, 2007], improving reconstruction of man-made structures [Mičušík and Košecká, 2010] etc. These approaches directly estimate depth for superpixels and can be considered piecewise planar reconstruction approaches where superpixels provide the piecewise planar regions.

In contrast, we use superpixels for delineating regions with unreliable or poor depth and using special processing for such regions (see Chapter 4). These unreconstructed regions would remain as such even if we used [Zitnick and Kang, 2007; Mičušík and Košecká, 2010]; they fail to reconstruct because of reasons that hold for any 3D depth estimation approach – lack of texture, stochastic texture, complex geometry, insufficient images etc.

2.5 Perception of image-based rendering artifacts

Our target is to synthesize *plausible* novel views of very complex scenes using as few photographs as possible. The available image data is typically much sparser than what would be required to generate perfect novel views. This hampers the level of plausibility because of incorrect parallax, perspective distortions, spatial rendering artifacts like ghosting and temporal artifacts like popping.

There has been recent interest in studying perception for images generated using image-based rendering. The goal of the majority of these approaches is to make "perceptually-optimal" algorithmic decisions that help mask away artifacts, sometimes accompanied by perceptual studies to confirm algorithmic choices.

Perspective distortions The most common problem with reprojecting an image captured from one viewpoint into another viewpoint is perspective distortion. Perspective is an important cue that helps determine the 3D layout of a scene up to a scale factor [Sedgwick, 1991]. Vision science has studied perspective in the context of picture perception focusing on how perspective distortions affect the perception of 3D shape. Consider the picture of a slanted rectangle. Sedgwick [1991] formulates the perceived slant of the rectangle using vanishing points; more specifically, using the angle between a line from the viewer to the vanishing point and a line from the viewer to the rectangle. When a picture is viewed from the center of projection, people are quite accurate at recovering the 3D geometry of the original scene, including the slants of surfaces in that scene [Smith and Smith, 1961; Cooper *et al.*, 2012]. If the viewer's eye is offset from the center of projection, perspective-based cues no longer specify the original 3D scene; instead, they specify a different, distorted scene. Our target is to quantify this effect as a function of capture and viewing parameters.

Rendering artifacts Almost all image-based rendering approaches accumulate visual content from multiple input images as the novel viewpoint transitions across the scene. Misalignment between input images when they are reprojected into the novel viewpoint, caused by inaccurate correspondences between the images, results in rendering artifacts. A majority of approaches blend multiple images, which can lead to ghosting artifacts while avoiding blending can lead to temporal discontinuities known as "popping" artifacts. Some approaches use perceptual studies to detect rendering artifacts or select rendering parameters to mitigate the artifacts. Morvan and O'Sullivan [2009b] present perceptually-motivated compression techniques for the large amounts of image data required for lumigraphs. Berger *et al.* [2009] detect ghosting artifacts in images using image edges; however, they do not analyze the factors that lead to ghosting artifacts. Schwarz and Stamminger [2009] present a perceptually-motivated predictor for popping artifacts for general computer graphics applications. They do not compare popping artifacts to other alternatives such as blending. The most closely related perceptual study on image-based rendering techniques is the study of overall visual quality of panoramic transitions [Morvan and O'Sullivan, 2009a]. They concluded that the magnitude of the depth discontinuity at silhouettes is a key factor in visual quality. This work was an important first step towards the goal of understanding the perception of rendering artifacts. Tompkin *et al.* [2013] compare cross-fading effects using different forms of 3D representations of the scene – full 3D geometry, 3D point cloud, single plane, 2D correspondences, no geometry or abrupt changes with no cross-fading at all. They conclude that using full 3D geometry is by far the best solution. Our study can be considered orthogonal to their work because we fix the 3D geometry of the scene and vary the rendering parameters that control the degree of ghosting or popping (see Section 5.2), while [Tompkin *et al.*, 2013] compare different forms of 3D geometry for generating transitions while keeping the rendering parameters fixed. The above approaches study "explicit visual processes" where the participants are explicitly asked to judge the quality of the stimuli by performing tasks or answering questions. Mustafa *et al.* [2012b] study "implicit visual process" where participants' response to stimuli is measured by an ElectroEncephaloGraph (EEG). Mustafa *et al.* [2012b] show that different rendering artifacts invoke different responses from the brain. This observation justifies the need for the comparative study of different artifacts presented in this thesis. The advantage of implicit studies over explicit psychophysical studies is that the results are not biased by the nature of questions or tasks performed by participants. However, the main drawback is the low signal-to-noise ratio in the data recorded by the EEG, as noted and partially alleviated in [Mustafa *et al.*, 2012a]. The relationship between inferences drawn from explicit and implicit studies is also unclear especially when the two are divergent.

To the best of our knowledge, there are no perceptual studies that investigate ghosting and popping artifacts with respect to each other in the context of image-based rendering. Excessive blending leads to ghosting artifacts but creates smooth transitions between viewpoints while the contrary gives crisp images but popping artifacts in transitions. The study of these opposing artifacts is critical because most image-based rendering approaches present a tradeoff between ghosting and popping artifacts by tweaking parameters until the result "appears good".

2.6 Discussion

A majority of image-based rendering approaches are restricted to studio captures, small baselines and/or view interpolation [Mahajan *et al.*, 2009]. These approaches are not directly applicable to our target. Other approaches that use 3D geometry [Buehler *et al.*, 2001; Eisemann *et al.*, 2008] are promising for our application, but they are severely restricted by their heavy dependence on *accurate* 3D geometry. Multi-view stereo [Furukawa and Ponce, 2009] and surface extraction [Kazhdan *et al.*, 2006] can provide 3D geometry for all classes of scenes, but their results can be very inaccurate in complex scenes. These limitations render the state of the art inadequate for our target of free viewpoint urban navigation.

Instead of relying on 3D geometry alone to provide silhouettes and reprojection constraints, we resort to 2D constraints extracted from images. Image oversegmentation [Achanta *et al.*, 2012] can divide an image into hundreds of *superpixels* which reliably capture occlusion boundaries. These offer a promising alternative for reinforcing silhouettes. Variational 3D image warping [Liu *et al.*, 2009] can synthesize plausible novel views using a small number of reconstructed points, albeit without occlusions over very small baselines only. We pursue this direction of research in the following chapters to design image-based rendering approaches which use 3D geometry as one of the constraints in a system which seeks to preserve the integrity of the final rendered 2D image by using 2D constraints such as variational warping and oversegmentation.

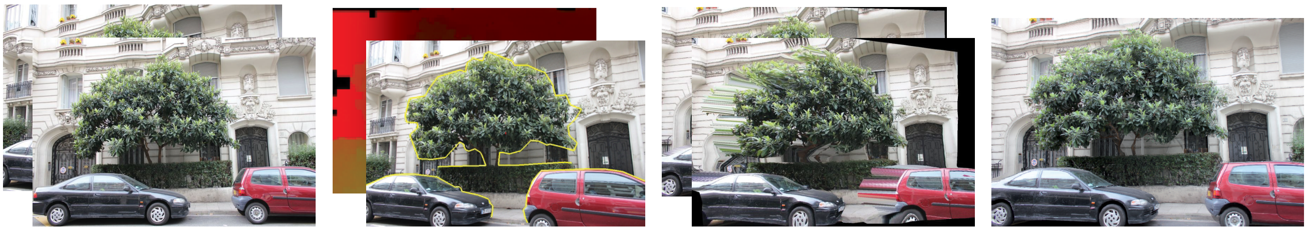
Chapter 3

Silhouette-aware Warping for Image-based Rendering

We present a image-based rendering solution that addresses our target of handling urban scenes with a small number of input images. The solution presented in this chapter demonstrates that our intuition of using image-based constraints to compensate for lack of accurate 3D geometry is indeed a powerful idea. The two main ideas introduced in the chapter concur with the main goals of the thesis: firstly, we use silhouettes extracted from images and quasi-dense 3D point clouds, which improve robustness towards inaccurate 3D models; and secondly, we develop a silhouette preserving image warp that explicitly enforces constraints to reduce distortion in final images. These lead to significant improvement in rendering quality compared to previous work. Our rendering pipelines does not enforce strict restrictions on novel viewing paths such as view interpolation, which combined with improved rendering quality, is an important step towards the ultimate goal of free viewpoint navigation.

3.1 Introduction

State of the art image-based rendering pipelines use the best quality 3D reconstruction [Furukawa and Ponce, 2009] with view dependent texturing [Buehler *et al.*, 2001; Eisemann *et al.*, 2008], resulting in powerful systems which have been shown to handle a wide variety of scenes. However, these systems have several limitations, the most important being their dependence on accurate 3D models which can be very hard to generate for complex scenes. Geometric reconstruction approaches do not give accurate results for foreground objects with complex shapes such as trees, or sharp depth discontinuities such as vehicles parked in front of façades. Such situations are very frequent, especially in urban scenes. Consequently, image-based rendering approaches that rely on accurate geometry can suffer from artifacts for such scenes. While it is possible to improve reconstruction quality using more input images, the re-



(a) Input images (b) User assisted silhouettes & depth samples (c) Silhouette-aware warp (d) Final result after blending

Figure 3.1: (a) We use 10-20 input photographs and multi-view stereo to create a dense 3D point cloud. (b) With our user-assisted preprocessing, the user designates important silhouettes and we reduce the 3D point cloud to ~6,000 depth samples per image. (c) The silhouettes and depth samples guide a *silhouette-aware warp,* applied to 4 images at each frame. (d) Our renderer generates a high-quality final image which handles hard cases such as trees and other foreground objects.

sults for complex foreground objects do not improve proportionally. Adding more images reduces the disparity, making it easier for stereo reconstruction approaches to match image features. However, this does not help the case of texture-poor surfaces or busy textures such as vegetation because commonly used metrics such as normalized cross correlation remain ambiguous in such situations irrespective of the disparity.

We present a new approach which addresses these limitations. Our central idea is to compensate for incorrect or incomplete geometric information by introducing silhouette preserving variational warping. We focus on scenes containing hard to reconstruct complex foreground geometry within the context of wide baseline image-based rendering. Our main contributions can be summarized as follows:

- the representation consisting of sparse depth constraints and silhouette edges, which enables shape-preserving variational warping for wide-baseline scenarios,
- the introduction of *silhouette-aware* warping in which "elastic" edges absorb distortions caused by (dis)occlusions while depth discontinuities are preserved, and
- an efficient rendering algorithm with a good trade-off between blurring and color discontinuities.

Our approach greatly reduces artifacts compared to the best combination of state of the art techniques, while overcoming the limitations discussed above (see results Figure 3.9 and 3.10). In particular we treat scenes with hard-to-reconstruct objects and viewing paths which do not interpolate the input cameras. We require only a small number of images, resulting in a lightweight capture process.

3.2 Overview

The input to our method is a set of images calibrated using [Snavely *et al.*, 2006] and a 3D point cloud generated using [Furukawa and Ponce, 2009]. The point cloud can be projected into the input images to give depth values at certain pixels, which we refer to as *depth samples*. Our approach has three main steps (see Figure 3.1):

Preprocessing Our approach first selects silhouettes around foreground objects (trees, cars, etc.) for each input image (Section 3.3.1) in a user assisted fashion. These silhouettes are used to correctly handle depth discontinuities. The second step decimates the set of depth samples to a sparse uniformly distributed set. This step also fills in poorly reconstructed regions using depth from neighboring points (Section 3.3.2). The resulting depth samples serve as constraints for our image warp.

Silhouette-aware image warp The depth samples from each input image are mapped to their respective desired final positions by reprojecting them into the novel view. These act as guiding constraints for our image warp in the form of projection energy. A similarity transform energy prevents deformation of warp mesh triangles. We define "elastic" triangles around silhouettes which absorb the distortion because of depth differences. The last energy term minimizes warping artifacts that distort the shape of silhouettes (Section 3.4.1).

Rendering To synthesize any novel view, we pre-select 4 closest input images and warp them with the silhouette-aware warp. At any pixel of the novel view, we compute the blending weights for the 4 pixels from each of the warped images. We then blend the two candidates to give the final result which gives motion parallax. The blending weights are designed to correctly diminish the visual impact of strong distortion produced by the elastic edges around the silhouettes. Finally, we use an optional Poisson synthesis step to alleviate seams.

Our image warping works for poorly reconstructed objects because the silhouettes segment the image into contiguous regions at different depths. The uniform set of depth samples proves sufficient for correct 2D warping of each region, resulting in significant quality improvement compared to methods which rely on accurate 3D models.

3.3 Extracting silhouettes and depth samples

Our approach requires pre-annotated silhouettes (Figure 3.2(a)) and a uniform distribution of depth samples on each image (Figure 3.2(b,c)), both of which can be provided by a variety of approaches. Our core image-based rendering approach is independent of the methodology used for providing either of these.

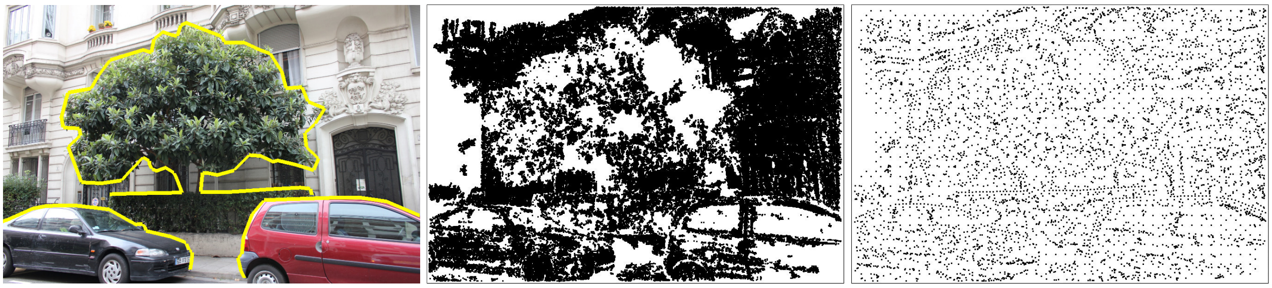

(a) (b) (c)

Figure 3.2: (a) Silhouettes marked at depth discontinuities, (b) original set of depth samples obtained from the 118,049 reconstructed 3D points (shown as black pixels), and (c) uniformly distributed 6324 depth samples selected by our approach. Note the regions with no original depth samples are also filled with new samples.

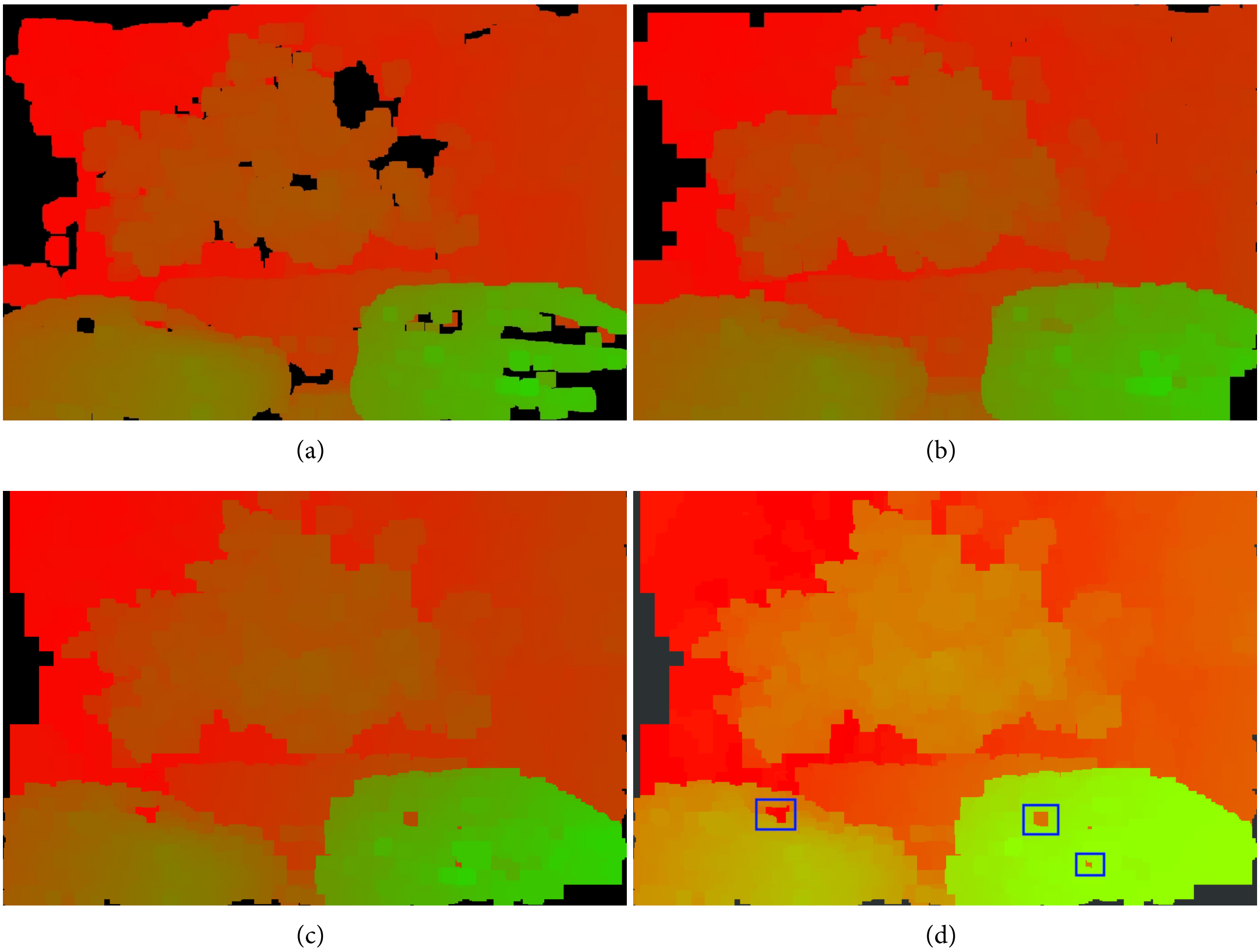
3.3.1 Silhouette selection

Silhouettes can be manually authored in each input image or computed (semi) automatically. Modern image segmentation algorithms [Arbelaez *et al.*, 2011] can be used to extract image boundaries automatically. Considering the importance of silhouettes for a variety of applications (object recognition, 3D reconstruction etc.), segmentation techniques have been adapted for extracting silhouettes or occlusion boundaries from a single image [Hoiem *et al.*, 2007b] or motion sequences [Stein and Hebert, 2009; He and Yuille, 2010]. Even though the edge maps returned by these approaches are impressive, they often have many false positives which need to be removed manually and missing edges have to be manually added. In addition, edge-maps have to converted to binary maps using a dataset dependent threshold. Then, they need to be converted into polygonal curves using chaining [Teh and Chin, 1989] and line segment decomposition e.g., by Douglas-Peucker algorithm [Douglas and Peucker, 1972]. Noisy edge-maps such as those in our scenes can make polyline approximation ambiguous.

We experimented with such approaches extensively, both by applying them directly, and developing extensions. We observed that, in practice, segmentation followed by the same degree of user interaction does not give the same accuracy as manual authoring and can actually take longer than direct manual edge marking. Please refer to Appendix A for a summary of our extensive tests and a comparison of automatic methods with manual authoring.

In view of the above, we prefer manual silhouette authoring over a combination of segmentation and user intervention. Manual authoring took 40-60 seconds for each image in our datasets. This is much faster compared to the time needed to manually create pixel-accurate geometry by hand. For objects such as trees, such geometry creation would require a skilled and experienced modeler and even then would probably require hours of work. We alleviate the need for manual silhouette marking in the next chapter by using image oversegmentation (see Chapter 4) and a different warping strategy.

3.3.2 Depth sample selection


(a)

(b)

(c)

(d)

Figure 3.3: (a) Splat of all input samples with depth ranging from green (near) to red (far) and splat size 21×21, (b) same splat after retaining around 5000 samples, (c) same splat after hole filling, and (d) outlier samples with wrong depth shown in blue box.

The 3D point cloud obtained from multi-view stereo [Furukawa and Ponce, 2009] can be projected into input images to give depth values at certain pixels, which we refer to as depth samples. The goal of depth sample selection is to retain a uniform distribution of depth samples over the image, filling regions that have few or no samples and removing possibly erroneous samples near silhouettes or specular regions.

Decimation We splat the 3D point cloud with a large splat size and depth test enabled. We count the number of pixels that each splatted depth sample covers. We select a subset of desired size that covers the maximum number of pixels (see Figure 3.3(a,b)). The splat size is not critical as long as it is not too small; we used 21×21 in our experiments.

Hole filling The depth samples retained after decimation are splatted on the image. If a window of $n \times n$ pixels does not contain any splatted depth samples, we mark this window as a hole. We add a sample which projects inside this window at the depth of nearest neighbor sample. We choose n the same as the splat size used earlier (see Figure 3.3(c)). It is important to note that the newly added depth samples are generated on a per-image basis and are not photoconsistent. They do not augment the reconstruction; they simply provide constraints for stabilizing the image warp described in Section 3.4.

Silhouette depth samples To avoid mixing foreground and background samples on either side of the silhouettes, we conservatively remove all existing samples within a small distance of silhouettes and replace them with samples using the depth from their respective side. This ensures that the silhouettes clearly separate samples with different depths. We observed this does not compromise warp accuracy because such regions are too small to contain significant depth gradients.

Manual outlier removal This optional step is useful when there are many samples with incorrect depths. Our interface shows samples with color coded depths, which makes it easy for the user to identify such outliers (see Figure 3.3(d)). They can be removed by a simple 'select-and-delete' operation at any stage of the process.

In our examples, 3D reconstruction produced 120,000-200,000 depth samples for each image. Using the process described above, we retained 5000-6000 samples per image. We observed that 6-9K samples did not improve the warp quality and less than 3000 samples led to warping artifacts. The optimal number of samples actually depends on desired output image resolution. A higher desired level-of-detail would require more constraints for the warp, hence more samples. The entire process, including user interaction (if needed), took about 4 minutes for a dataset of 15 images.

3.4 Shape-preserving warp using 3D constraints

Given a novel view, expressed by a camera projection matrix C_N, our goal is to warp the input images $I_1, I_2, \ldots, I_k$ so that they match the actual scene as it would have appeared in that view as faithfully as possible. We then use the warped images in the rendering pipeline (see Section 3.5).

Denote by C_i the camera projection matrix of input image I_i. If we knew the mapping U_i from every pixel $\mathbf{x} \in I_i$ to the corresponding 3D point $\mathbf{p}$, i.e. $C_i(\mathbf{p}) = C_i(U_i(\mathbf{x})) = \mathbf{x}$, then the warp of image I_i into the new view would be simply $C_N \circ U_i$. However, we do not have a dense per-pixel 3D reconstruction of the scene. On the contrary, we wish to use only a small set of depth samples for effective image warping. We therefore replace the per-pixel warp above with a sparse set of constraints on pixel positions and a warp prior that dictates the warping function to be smooth (except at (dis)occlusions) and locally preserve

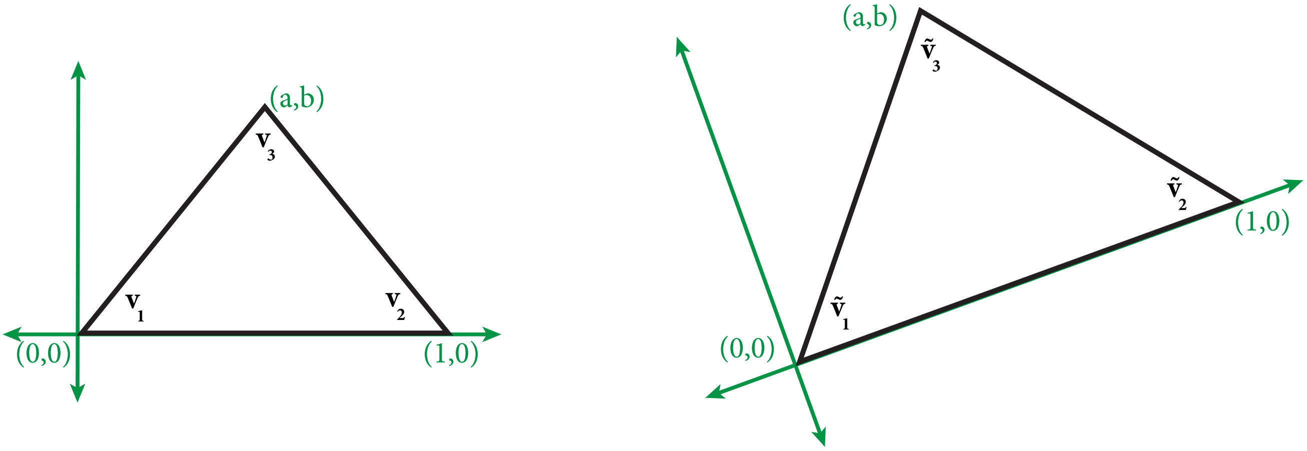


Figure 3.4: Warp mesh triangle before and after a similar transform. The local coordinate frame $\{(\mathbf{v}_2 - \mathbf{v}_1), R_{90} \cdot (\mathbf{v}_2 - \mathbf{v}_1)\}$ attached to the triangle is shown in green.

the shape of the image content. We handle occlusions by explicitly modeling the desired warp behavior along silhouettes, as will be described in Section 3.4.1. Together, the positional constraints and warp behavior priors define an energy functional which we minimize to warp the input images.

Setup In order to compute the variational warp, we discretize the image domain by overlaying a triangle mesh on image I_i. Each face of the warp mesh has 0 or more depth samples. We denote the input warp mesh vertices by $\mathbf{v}$ and their warped positions by $\tilde{\mathbf{v}}$, which are the unknowns in the warp optimization. We define the variational warp as a linear optimization; the full warp can be computed by solving for warped vertex positions $\tilde{\mathbf{v}}$ and rendering the warped mesh with the original texture coordinates.

Reprojection energy Recall that the depth sample preprocessing step from Section 3.3 gave a sparse set $\mathscr{D}_i$ of uniformly distributed depth samples for each input image. For each depth sample $D[\mathbf{x}]$, we locate the triangle T of the warp mesh that contains the depth sample. Denote the vertices of T by $(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3)$ and let the barycentric coordinates of the location of the depth sample at pixel $\mathbf{x}$ in triangle T be (α, β, γ):

$$\mathbf{x} = \alpha \cdot \mathbf{v}_1 + \beta \cdot \mathbf{v}_2 + \gamma \cdot \mathbf{v}_3 \tag{3.1}$$

The reprojection energy measures the distance between the warped position of the depth sample and the ideal reprojected location using the novel view matrix C_N:

$$E_p[\mathbf{x}] = \|\alpha \cdot \tilde{\mathbf{v}}_1 + \beta \cdot \tilde{\mathbf{v}}_2 + \gamma \cdot \tilde{\mathbf{v}}_3 - C_N \cdot C_{I_i}^{-1} \cdot D[\mathbf{x}]\|^2 \tag{3.2}$$

where $C_{I_i}^{-1}$ is the back-projection matrix of image I_i.

Shape preserving energy To minimize the distortion caused by the warp, the warp must be locally shape-preserving. We therefore use a similarity energy term, such that the transformation of each mesh triangle is as close as possible to a similarity transformation. Analogous energy terms were used in [Liu *et al.*, 2009; Zhang *et al.*, 2009; Wang *et al.*, 2010]. Consider a mesh triangle T with vertices $(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3)$ and attach a local orthogonal frame to it: $\{(\mathbf{v}_2 - \mathbf{v}_1), R_{90} \cdot (\mathbf{v}_2 - \mathbf{v}_1)\}$, where R_{90} is a counterclockwise rotation by 90 degrees. Assume that $\mathbf{v}_1$ is the origin of the local frame; $\mathbf{v}_2$ can then be expressed simply as $(1,0)$ in the local coordinate system as shown in Figure 3.4, and $\mathbf{v}_3$ as (a, b) given by:

$$\begin{aligned} a &= \frac{(\mathbf{v}_3 - \mathbf{v}_1)^T \cdot (\mathbf{v}_2 - \mathbf{v}_1)}{\|\mathbf{v}_2 - \mathbf{v}_1\|}, \\ b &= \frac{(\mathbf{v}_3 - \mathbf{v}_1)^T \cdot R_{90} \cdot (\mathbf{v}_2 - \mathbf{v}_1)}{\|\mathbf{v}_2 - \mathbf{v}_1\|} \end{aligned} \tag{3.3}$$

These local coordinates can be used to express each vertex of the triangle as a linear sum of the basis vectors of the local frame:

$$\begin{aligned} \mathbf{v}_1 &= \mathbf{v}_1 + 0 \cdot (\mathbf{v}_2 - \mathbf{v}_1) + 0 \cdot R_{90} \cdot (\mathbf{v}_2 - \mathbf{v}_1), \\ \mathbf{v}_2 &= \mathbf{v}_1 + 1 \cdot (\mathbf{v}_2 - \mathbf{v}_1) + 0 \cdot R_{90} \cdot (\mathbf{v}_2 - \mathbf{v}_1), \\ \mathbf{v}_3 &= \mathbf{v}_1 + a \cdot (\mathbf{v}_2 - \mathbf{v}_1) + b \cdot R_{90} \cdot (\mathbf{v}_2 - \mathbf{v}_1) \end{aligned}$$

As the triangle undergoes a similarity transform, the local coordinate frame remains orthogonal and local coordinates (a, b), computed from initial positions, remain the same. The final vertex position $\tilde{\mathbf{v}}_3$ can be expressed as a function of local vertex positions and final position of other two vertices $\tilde{\mathbf{v}}_1$ and $\tilde{\mathbf{v}}_2$. The similarity energy term can thus be expressed as:

$$E_s[T] = \|\tilde{\mathbf{v}}_3 - \left(\tilde{\mathbf{v}}_1 + a \cdot (\tilde{\mathbf{v}}_2 - \tilde{\mathbf{v}}_1) + b \cdot R_{90} \cdot (\tilde{\mathbf{v}}_2 - \tilde{\mathbf{v}}_1)\right)\|^2 \tag{3.4}$$

3.4.1 Silhouette-aware warp

The energies described so far are smooth and shape-preserving everywhere. However, the warp should have discontinuities in the vicinity of silhouettes because of the depth discontinuities. When considering a small neighborhood around a silhouette edge, the warp may have a discontinuity perpendicular to the edge (to mimic (dis)occlusion) while remaining shape-preserving in the tangent direction. We model this behavior by conceptually inserting a narrow and highly elastic band parallel to the silhouette that is allowed to absorb heavy distortion due to discontinuity (see Figure 3.5(b)). The shape of the silhouette itself, on the other hand, is preserved by adding a curve-similarity energy term described below, thus avoiding distortion of foreground objects. In order to properly discretize the image domain

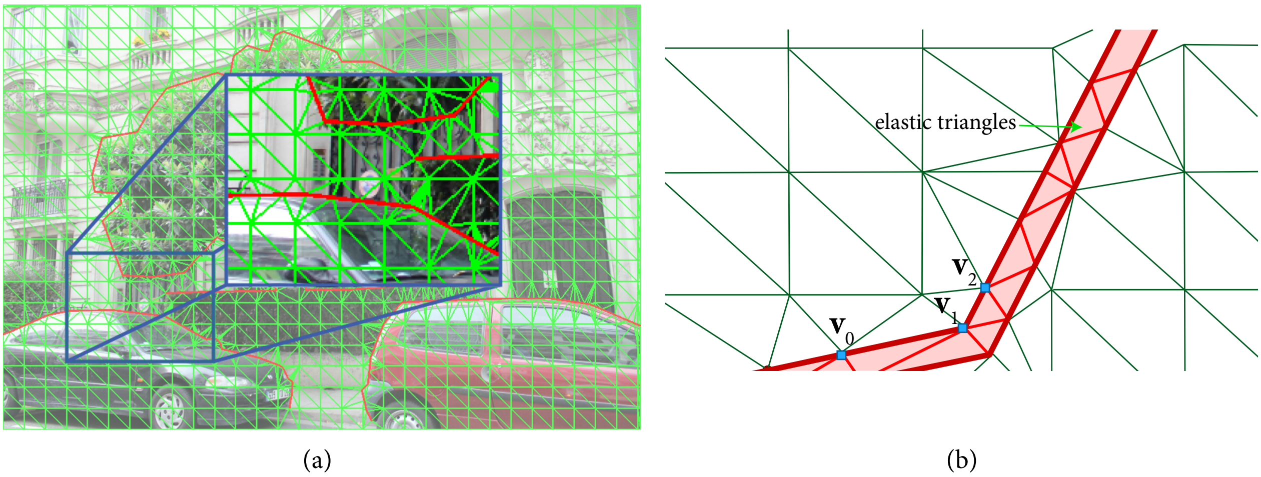


Figure 3.5: (a) Constrained conformal triangulation used as warp mesh with constrained silhouette polylines shown in red, and (b) two parallel constrained polylines (shown in red) added for the silhouette polyline. Any three consecutive vertices on either of these polylines form an edgelet. The triangles wedged between these edges form the elastic band.

and formulate the silhouette-specific energy, we take all the silhouette polylines (from Section 3.3) and duplicate them, offsetting the resulting parallel edges by 2 pixels (see Figure 3.5(b)). We create a constrained conformal Delaunay triangulation where the doubled silhouette edges are the constraint edges (see Figure 3.5(a)). All triangles between silhouette lines belong to the *elastic band* and are excluded from the energy term shape-preserving energy in Equation 3.4, thus allowing the band to be elastic.

To preserve the shape of the silhouette itself, we require the silhouette curve to locally undergo a shape-preserving transformation. The energy formulation is similar to Equation 3.4, but it is defined on the silhouette curve this time, instead of a 2D region. Consider three consecutive vertices lying on the curve, indexed w.l.o.g. as $e = (\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2)$. We call such a sequence of two curve edges an *edgelet* (see Figure 3.5(b)). A similarity transformation of the edgelet e means that the angle θ between the two edges, as well as the length ratio $\|\mathbf{v}_0 - \mathbf{v}_1\|/\|\mathbf{v}_2 - \mathbf{v}_1\|$, remains the same. We can therefore write the curve similarity energy term as:

$$E_b[e] = \|(\tilde{\mathbf{v}}_0 - \tilde{\mathbf{v}}_1) - \left(\frac{\|\mathbf{v}_0 - \mathbf{v}_1\|}{\|\mathbf{v}_2 - \mathbf{v}_1\|}\right) \cdot R_\theta \cdot (\tilde{\mathbf{v}}_2 - \tilde{\mathbf{v}}_1)\|^2 \tag{3.5}$$

where R_θ is the 2×2 rotation matrix that rotates the edge $(\mathbf{v}_2 - \mathbf{v}_1)$ onto $(\mathbf{v}_0 - \mathbf{v}_1)$. The effect of the above silhouette-aware discontinuous image warp is shown in Figure 3.6. The smooth warp described in Section 3.4 will cause heavy distortion near depth discontinuities (see Figure 3.6(a,b)).

Liu *et al.* [2009] use energies E_p and E_s alone which would allow homogeneous distribution of the heavy distortion over the entire image. In contrast, our silhouette-specific energy term E_b preserves the

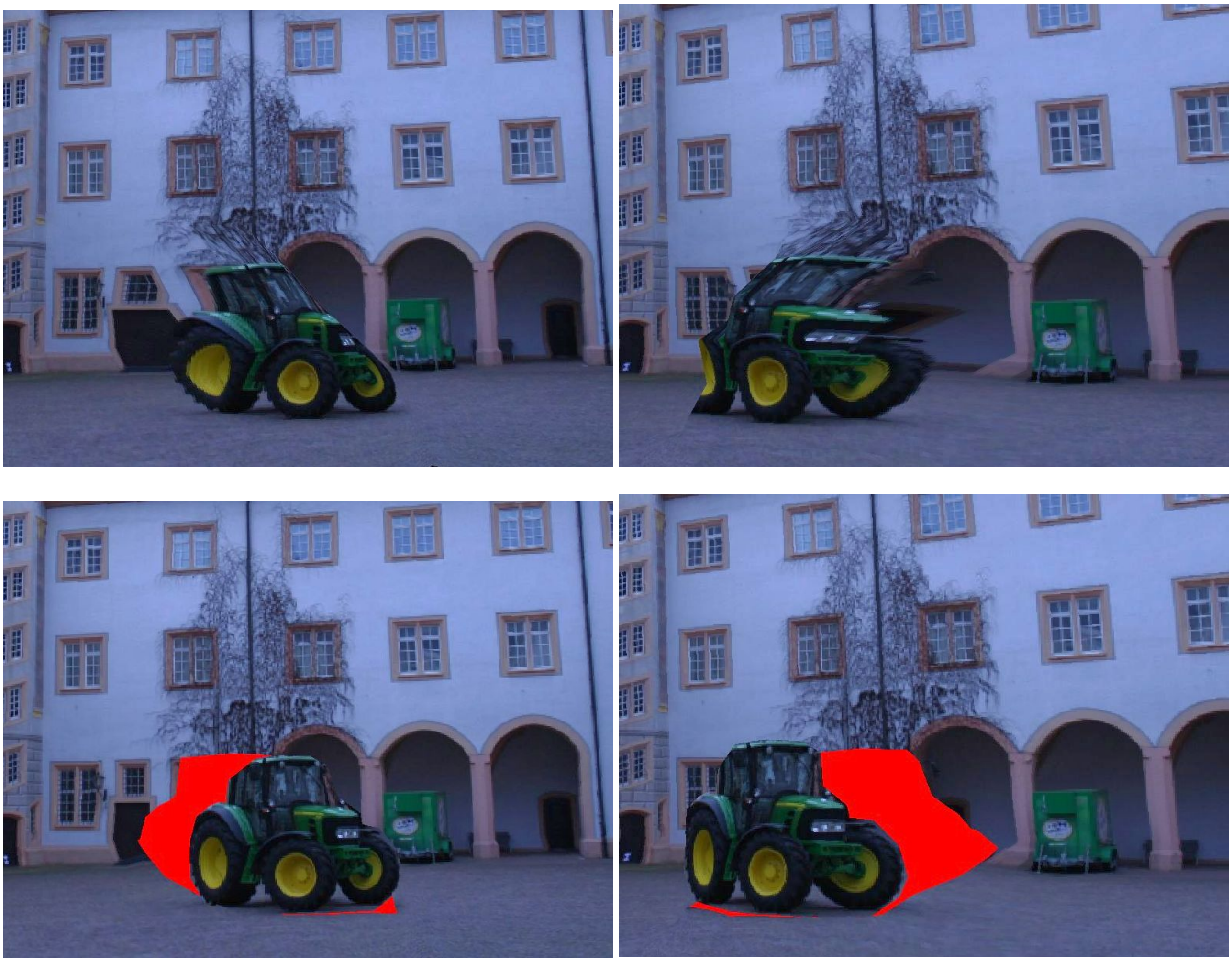


Figure 3.6: Top row: An input image warped to different novel views without any silhouette handling. Bottom row: Same image warped to same views using our silhouette-aware discontinuous warp. The elastic band that absorbs all the distortion is shown in red.

local shape of the silhouettes by absorbing all the distortion in the *elastic* band. When pixels become occluded, the elastic band enables accurate mesh fold-over along the silhouette. When pixels are disoccluded, the elastic band stretches without deforming the silhouette (shown in red in Figure 3.6(c,d)). These bands are later filled using texture from a different image in the final result (explained in Section 3.5). Thus, our image warp is robust to wide-baseline (dis)occlusion.

3.4.2 Total warp energy

The optimal warp minimizes the weighted sum of the reprojection, similarity and silhouette energies from Equations 3.2, 3.4 and 3.5):

$$E_i = w_p \sum_{\forall \mathbf{x} \in \mathcal{D}_i} E_p[\mathbf{x}] + w_s \sum_{\forall T \in \mathcal{T}_i} E_s[T] + w_b \sum_{\forall e \in \mathcal{E}_i} E_b[e] \tag{3.6}$$

Here, $\mathcal{D}_i$ is the set of all depth samples, $\mathcal{T}_i$ is the set of all warp mesh triangles and $\mathcal{E}_i$ is the set of edgelets of image I_i. We use the reprojection E_p and silhouette energy E_b as strong constraints which guide the optimization and the shape preserving term E_s as a weak regularizer to prevent distortions on all triangles outside the elastic band. Thus, we set $w_p = w_b = 2$, $w_s = 0$ (for elastic band triangles) and $w_s = 1$ for triangles outside the elastic band. Every input image I_i has its own warp mesh and an associated linear system E_i; we use OpenMP to warp multiple images on parallel cores.

The energy E_i is quadratic in the unknown warped vertex positions $\tilde{\mathbf{v}}$; we therefore it has a unique minimum that is found by solving the sparse linear equation $\nabla E_i = 0$. We use the direct sparse Cholesky solver Taucs [Toledo, 2003]. Note that the system matrix does not change for novel view parameters views since only the right-hand side of the linear system changes. We therefore precompute the matrix factorization and only perform back-substitutions at runtime.

In our experiments, we found an initial 30×30 warp mesh to be sufficient for 800×600 pixel output frame resolution. This sampling is locally refined to insert the silhouette edges, as described above. We compute the final constrained conformal Delaunay triangulation using CGAL [Rineau, 2010]. Finally, the warped meshes are created by rendering the warped meshes using original vertex positions as texture coordinates.

3.5 Rendering

To synthesize a novel view, we first select a set of four images which can be used for all pixels in the final image. We observed that three to four input images are sufficient to synthesize a novel view. We warp these images to the novel view using our warp formulation from Section 3.4. We then compute blending weights of the contribution from each warped image at each pixel in a pixel shader and retain the best two candidates. We finally blend the candidates using weights which are very similar to unstructured lumigraph rendering [Buehler *et al.*, 2001], except that we use per-pixel blending. Buehler *et al.* [2001] compute blending weights at warp mesh vertices only and used standard OpenGL bilinear interpolation to obtain blending weights for each pixel while we compute the blending weights for each pixel directly in a pixel shader; this gives better results as compared to the original approach.

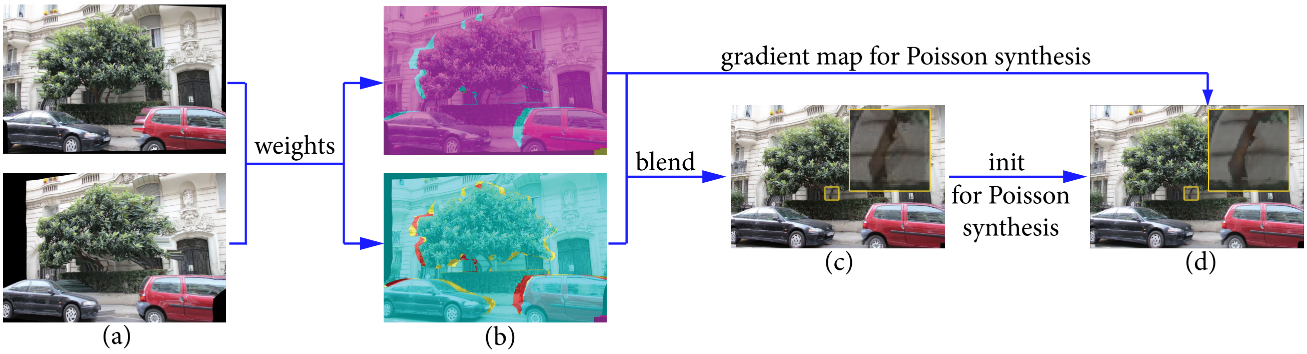


Figure 3.7: Rendering pipeline. (a) Warped images, (b) composite textures $\mathscr{R}_0$ and $\mathscr{R}_1$. Pixels from same warped image are shown in same color. (c) $\mathscr{R}$ generated by blending from $\mathscr{R}_0$ and $\mathscr{R}_1$. (d) Poisson synthesis output $\mathscr{R}'$ using $\mathscr{R}$ as Dirichlet constraints and gradient $\mathscr{G}$ from $\mathscr{R}_0$.

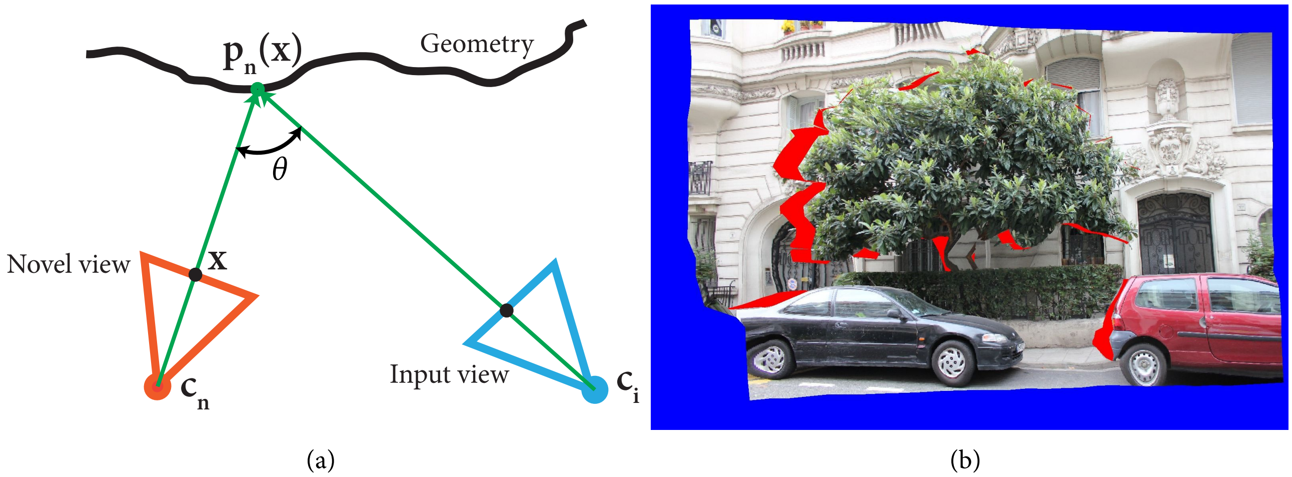


Figure 3.8: (a) Angle θ used for computing the penalty $P_{\text{ang}}(\tilde{I}_i, \mathbf{x})$, and (b) warped image showing pixels outside the warp mesh in blue and pixels inside elastic band in red.

Blending weights For each pixel, we select the best two images for blending using a penalty scheme inspired by [Buehler *et al.*, 2001]. Consider pixel $\mathbf{x}$ of the novel view with center of projection $\mathbf{c}_n$ and a warped input image $\tilde{I}_i$ whose center of projection is $\mathbf{c}_i$. Let $\mathbf{p}_n(\mathbf{x})$ be the point where the ray shot from $\mathbf{c}_n$ through pixel $\mathbf{x}$ intersects the scene geometry. The required geometry is generated by splatting all 3D points as depth samples into the novel view; holes are avoided by triangulating depth samples. Note that this geometry is typically much coarser than a geometric proxy generated by 3D reconstruction [Furukawa and Ponce, 2009] and surface extraction [Kazhdan *et al.*, 2006]. However, our approach is robust to geometric inaccuracies because this geometry is only used for computing blending weights. We use the angle between $(\mathbf{c}_n - \mathbf{p}_n(\mathbf{x}))$ and $(\mathbf{c}_i - \mathbf{p}_n(\mathbf{x}))$ as an angle penalty (see Figure 3.8(a)). We also define a field-of-view penalty that checks whether the pixel lies inside the warp mesh $\tilde{I}_i$ (shown in blue in Figure 3.8(b)). Our last term penalizes elastic band pixels because the texture in such regions is expected to be heavily distorted (shown in red in Figure 3.8(b)).

$$\begin{aligned}
P_{\text{ang}}(\tilde{I}_i, \mathbf{x}) &= \arccos\left(\langle \mathbf{c}_n - \mathbf{p}_n(\mathbf{x}),\ \ \mathbf{c}_i - \mathbf{p}_n(\mathbf{x})\rangle\right) \\
P_{\text{fov}}(\tilde{I}_i, \mathbf{x}) &= \begin{cases} \infty & \text{if } \mathbf{x} \text{ lies outside the warp mesh,} \\ 0 & \text{otherwise} \end{cases} \\
P_e(\tilde{I}_i, \mathbf{x}) &= \begin{cases} \infty & \text{if } \mathbf{x} \text{ lies inside the elastic band,} \\ 0 & \text{otherwise} \end{cases}
\end{aligned} \tag{3.7}$$

The final penalty is given by

$$P(\tilde{I}_i, \mathbf{x}) = P_{\text{ang}}(\tilde{I}_i, \mathbf{x}) + P_{\text{fov}}(\tilde{I}_i, \mathbf{x}) + P_e(\tilde{I}_i, \mathbf{x}) \tag{3.8}$$

We use perceptually-based guidelines from Chapter 5 that excessive blending caused by unstructured lumigraph weights can be objectionable. The best results are obtained when only two images are blended for each pixel in the novel view (see Section 5.2.4). In order to blend only the two most suitable images at each pixel, we create two textures $\mathscr{R}_0$ and $\mathscr{R}_1$, where $\mathscr{R}_0$ is composed of pixels having the lowest penalties from all warped images $\tilde{I}_i$ and $\mathscr{R}_1$ the second lowest. Contiguous blocks of pixels looked up from the same warped image are shown in different colors for $\mathscr{R}_0$ and $\mathscr{R}_1$ in Figure 3.7. We compute blending weights from penalties and store them in the alpha channels of $\mathscr{R}_0$ and $\mathscr{R}_1$, respectively:

$$\begin{aligned}
w_{\mathscr{R}_0}(\mathbf{x}) &= 1 - \Psi \cdot \left(\frac{P(\mathscr{R}_0, \mathbf{x})}{P(\mathscr{R}_1, \mathbf{x})}\right), \text{ where } \Psi \in [0, 1] \\
w_{\mathscr{R}_1}(\mathbf{x}) &= 1 - \Psi \cdot \left(\frac{P(\mathscr{R}_1, \mathbf{x})}{P(\mathscr{R}_1, \mathbf{x})}\right) = 1 - \Psi
\end{aligned} \tag{3.9}$$

The textures $\mathscr{R}_0$ and $\mathscr{R}_1$ are alpha blended to give the texture $\mathscr{R}$. We again use the heuristic of minimizing blending from Section 5.2.4 and introduce an extra factor Ψ that amplifies the difference in the penalties of $\mathscr{R}_0$ and $\mathscr{R}_1$, thereby reducing the degree of blending. For example, $\Psi = 1$ would cause $w_{\mathscr{R}_1}$ to always remain 0. If the ratio of penalties $P(\mathscr{R}_0)$:$P(\mathscr{R}_1)$ is 1:2, setting $\Psi = 0.87$ would cause the ratio of weights $w_{\mathscr{R}_0}$:$w_{\mathscr{R}_1}$ to become 4.33:1. This greatly reduces the contribution of $\mathscr{R}_1$ everywhere except where its penalty is very close to that of $\mathscr{R}_0$. In all examples presented here, we have used $\Psi = 0.87$, which gives good results in all our examples.

Poisson synthesis The blended texture $\mathscr{R}$ may have spatial discontinuities. The patch boundaries or seams of $\mathscr{R}_0$ and $\mathscr{R}_1$ may remain visible, especially the elastic band regions where the texture originates from an image with substantially different view. To create a better final output image, we use Poisson synthesis to inpaint the seam areas more gracefully. We create the gradient map $\mathscr{G}$ from $\mathscr{R}_0$, and its divergence map div $\mathscr{G}$. Note that both textures $\mathscr{R}_0$ and $\mathscr{R}_1$ contain contiguous patches of pixels looked up from the same input images. The gradient inside any of these patches is the same as the gradient of the original image, which helps retain crisp detail. We force the gradient of the final image $\mathscr{R}'$ to 0 for all pixels lying on patch boundaries $\mathscr{B}_0$ of $\mathscr{R}_0$, which amounts to smooth completion of those areas. We synthesize the final output by solving the Poisson equation:

$$\nabla^2 \mathscr{R}' = \operatorname{div} \mathscr{G} \quad \text{subject to} \quad \nabla \mathscr{R}'|_{\mathscr{B}_0} = 0 \tag{3.10}$$

We initialize the Poisson synthesis with the blended result $\mathscr{R}$ and perform a few multigrid Jacobi iterations. This helps smoothen patch boundaries and eliminates ghosting edges because the gradient map is created from texture $\mathscr{R}_0$. This can be seen in the inset of Figure 3.7 (right). Our approach thus prefers smooth spatio-temporal transitions over blending more images.

3.6 Results

To provide fair comparisons, we combine state of the 3D reconstruction [Furukawa and Ponce, 2009; Kazhdan *et al.*, 2006] with the best set of techniques available for free viewpoint wide baseline image-based rendering. We use unstructured lumigraph [Buehler *et al.*, 2001] rendering with per-pixel blending, in contrast to vertex blending used in the original method. We further add visibility checking algorithm [Eisemann *et al.*, 2008] to give occlusion handling. We use this hybrid approach for comparisons with our results in Figure 3.9 and 3.10 (see also video[1]). Given the lack of accurate geometry for foreground objects, previous approaches have ghosting artifacts and incorrect occlusion handling. Visibility checking does not alleviate occlusion artifacts because the proxy used for creating visibility maps is erroneous.

We present the result of our approach on challenging datasets which cannot be reconstructed accurately. Castle-P30 is a standard multi-view stereo dataset [Strecha *et al.*, 2008] with a foreground object (tractor) in very wide baseline images. Piecewise-planar reconstruction [Sinha *et al.*, 2009] gives unacceptable artifacts on the tractor. Aquarium-20 has multiple foreground objects at different depths, which are known to be difficult to handle [Mahajan *et al.*, 2009]. Street-10 and Tree-18 show our results for general urban scenes with vehicles and trees. Tree-18 dataset had many incorrectly reconstructed points on the tree, which were manually removed. The presence of vegetation makes 3D reconstruction and surface extraction very difficult; manually modeling such scenes is also very difficult and tedious. The baselines in our datasets vary from 275 pixels (14% of image height) in Aquarium-20; 260 pixels (16%) in Street-10 to 530 pixels (24%) in Castle-P30. With minimal user input, our approach generates

[1] Results video: http://vimeo.com/62038846

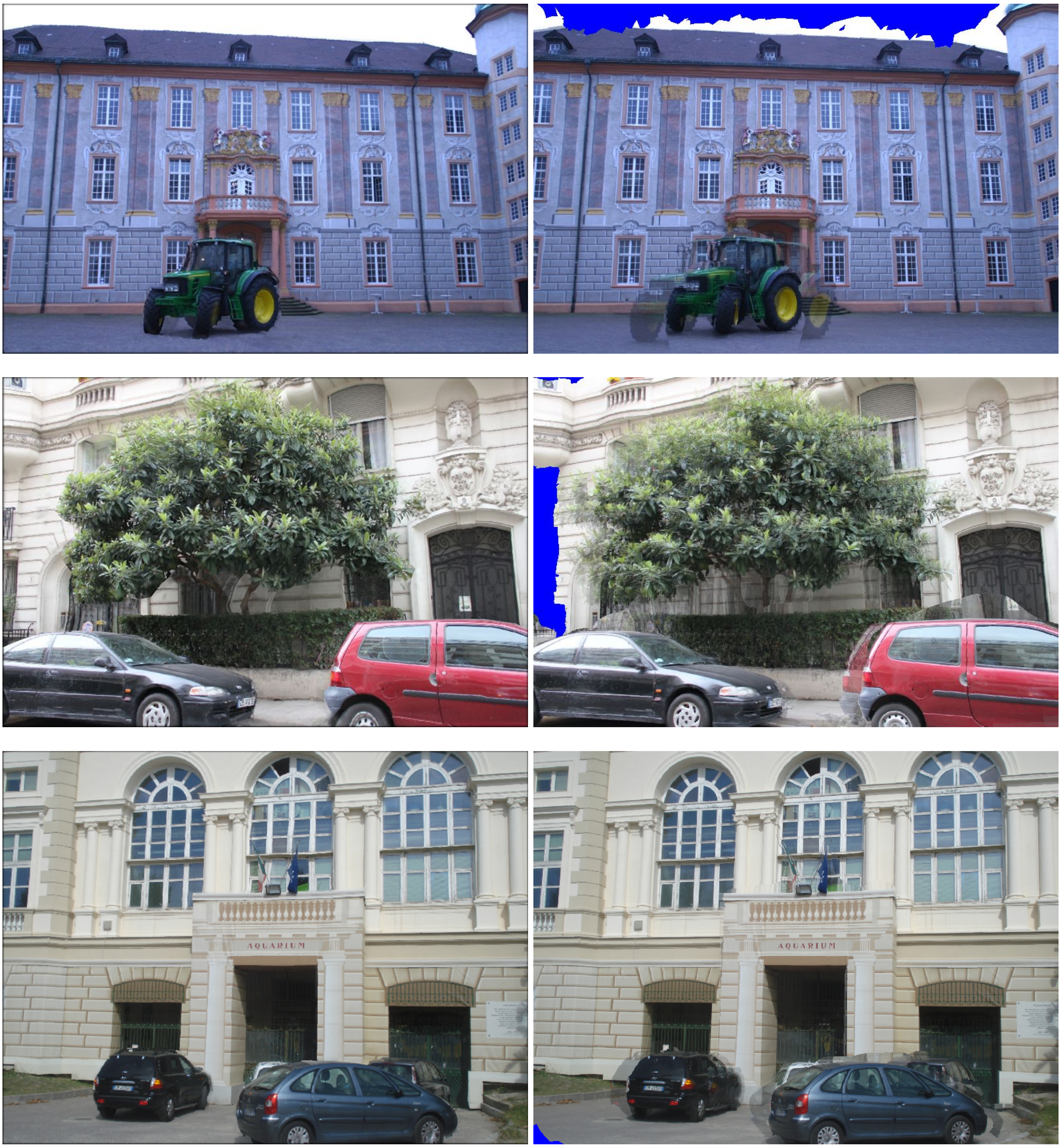


Figure 3.9: Comparison of our result (left) with previous approaches (right) on Castle-P30, Street-10, Aquarium-20 datasets (top to bottom).

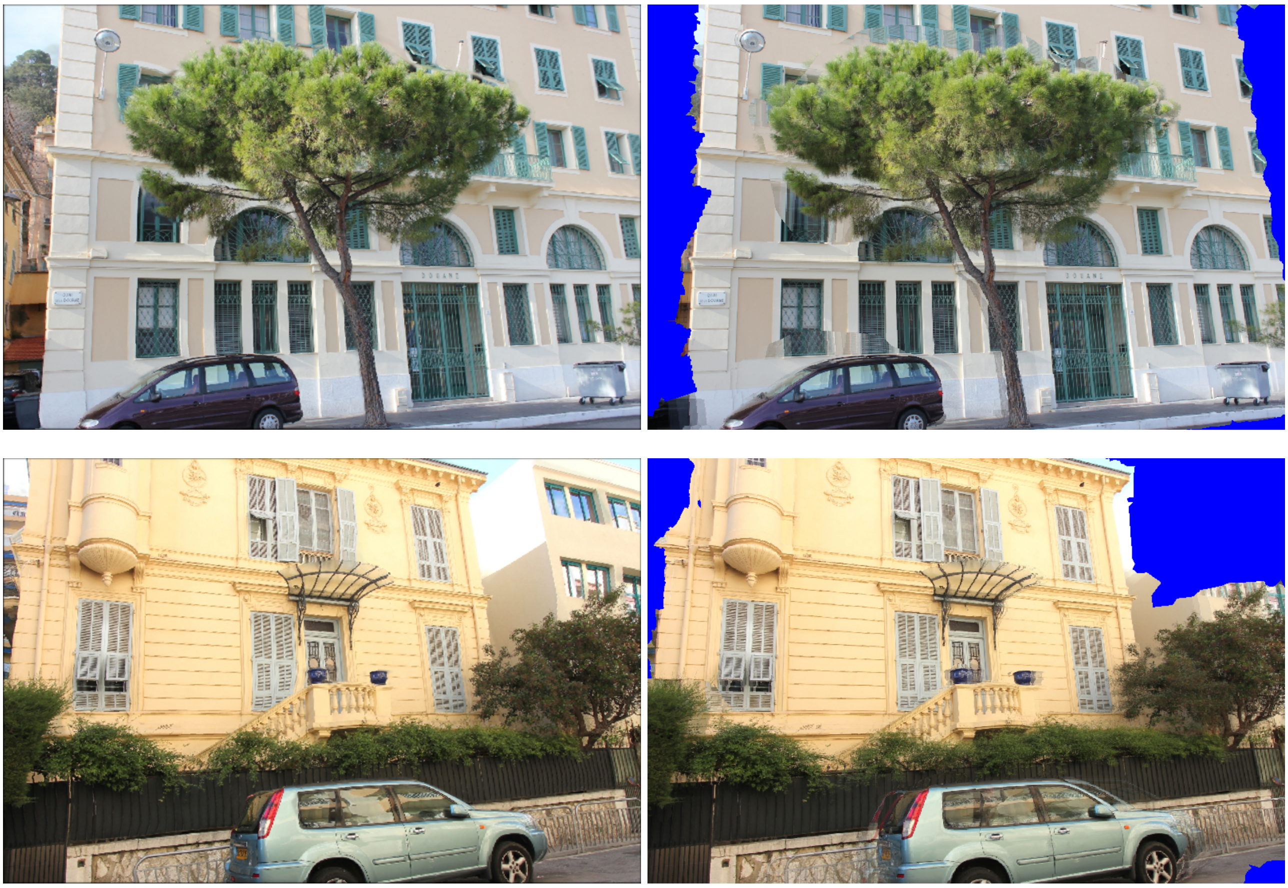


Figure 3.10: Comparison of our result (left) with previous approaches (right) on Tree-18, Yellowhouse-12 datasets (top to bottom).

much improved novel views from a free viewpoint camera path even for poorly reconstructed datasets.

Performance We tested our method on an Intel Xeon (2.8 Ghz) running Windows 7 with an NVIDIA Quadro 6000 GPU. Setting up the warp mesh and factoring the linear system for each input image with TAUCS takes 2-13 seconds. At run time, warping 4 input images on parallel cores takes 8-22 ms. The overall frame rate is 30-35 FPS for 1600×1200 size render targets.

Dataset	Input images	Proxy size	Our approach
Castle-P30	30	49.1 MB	2.5 MB
Street-10	10	25.3 MB	1.5 MB
Tree-18	18	21.2 MB	3.1 MB
Aquarium-20	20	32.9 MB	2.9 MB
Yellowhouse-12	12	26.0 MB	1.8 MB

Table 3.1: Storage for the proxy used by 3D model based approaches [Buehler *et al.*, 2001] compared to the storage requirements of our approach. Proxy sizes are meshes (vertices/faces) without normals/colors/texture coordinates in ASCII `.obj` format.

Storage Our method requires the storage of 5000-6000 depth samples per image. In contrast, detailed proxies can be quite large for complex scenes. (see Table 3.1). This is an important consideration for city-scale applications.

3.7 Limitations

The most important limitation of the approach is that it requires manually marked silhouettes. Our experiments with segmentation approaches [Stein and Hebert, 2009; He and Yuille, 2010; Arbelaez *et al.*, 2011] (see Appendix A) show that these approaches are not directly applicable partly because they are never 100% accurate and partly because we need silhouette polylines to insert into the warp mesh while these approaches only provide highly irregular contours.

We warp the full image as a linear system which is real time but the system is rather big and slow to factorize at the start of the application. Also, the conformal Delaunay triangulation can be numerically unstable at the junction of multiple silhouettes. The global warp also results in distortions when the novel camera is moved significantly away from the input cameras (see Figure 4.11), restricting the free viewpoint navigation zone. The triangulation is also difficult to setup for very thin foreground objects like railings. Lastly, the global warp assumes that the complete scene is in front of the viewing position because projection of any depth sample in a viewpoint that is in front of itself gives unpredictable results, causing the reprojection energy (Equation 3.2) and hence the complete warp to explode. The global warp does not allow the system to ignore such cases, hence the viewpoint can never "walk into" the scene, this is shown in Figure 4.12. We present a different approach addressing these limitations in the following chapter.

Chapter 4

Depth Synthesis and Local Warps for Plausible Image-based Navigation

The silhouette-aware warp approach described in Chapter 3 can generate plausible novel views from sparse irregular depth maps. It also addresses the all important issues of silhouettes and occlusion handling by a combination of image warping and manual silhouette selection. However, this manual step is the most important limitation. In addition, this approach involves a full image warp which involves solving a large linear system that can be slow and numerically unstable at times. The global warp also restricts the free viewpoint capabilities because it produces exaggerated distortions when the novel camera is moved significantly away from the input cameras.

Nonetheless, the silhouette-aware warp proves that shape-preserving warps are very effective at handling sparse depth maps. In this chapter, we build upon this insight and design a completely automated lightweight approach based on image oversegmentation and local shape-preserving variational warp. We first oversegment [Achanta *et al.*, 2012] the input images, creating superpixels of homogeneous color content which preserve depth discontinuities. We then introduce *depth synthesis* for poorly reconstructed regions by building a graph on the superpixel segmentation. Superpixels allow our algorithm to both identify regions requiring depth synthesis and to find appropriate depth exemplars. We then apply a local shape-preserving warp on the superpixels which reproduces all the advantages of the silhouette-aware warp. We improve the rendering algorithm of silhouette-aware warp to further reduce ghosting artifacts. The main contributions are firstly, a depth synthesis algorithm which provides depth samples in poorly reconstructed regions, and secondly, a local shape-preserving warp and rendering algorithm that uses the synthesized depth and oversegmentation to generate plausible novel views.

It is important to note that the goal of our depth synthesis is not to produce photoconsistent depth. The goal is to produce *plausible* depth and use it within the shape-preserving warp to produce plausible, though not physically accurate novel views, even when the user is far from the input cameras. We have applied our approach to 12 different scenes (see Figure 4.8), including one from Microsoft Photosynth and two from [Pollefeys *et al.*, 2008]. We demonstrate interactive navigation sessions for all scenes, which show that our approach pushes the limits of image-based rendering to free viewpoint navigation and complex urban imagery. At the same time, our approach is also more scalable and computationally lighter than a variety of previous approaches including [Eisemann *et al.*, 2008] and silhouette-aware warp (Chapter 3).

4.1 Overview

Preprocessing Our input is a small set of 15-25 images taken from multiple viewpoints. We preprocess the input data using off the shelf computer vision approaches. We first calibrate the cameras using [Snavely *et al.*, 2006] and reconstruct the scene using multi-view stereo [Furukawa and Ponce, 2009]. We project the 3D point cloud into input images to obtain a set of projected depth samples in each image. We then oversegment [Achanta *et al.*, 2012] all the input images creating superpixels that delineate regions of homogeneous color content and preserve depth discontinuities. Our image-based rendering approach is independent of the choice of reconstruction and segmentation approaches; we choose the state of the art for these tasks.

Our approach has two main steps: depth synthesis and *local* shape preserving warp, followed by a three pass rendering algorithm.

Depth synthesis The key motivation for this step is that even after using the best reconstruction, there can be significant regions with no depth. Most piecewise planar stereo [Sinha *et al.*, 2009] and image-based rendering algorithms [Goesele *et al.*, 2010] ignore such regions completely. Instead of discarding such regions, we synthesize *plausible* depth suitable for image-based rendering, which is not necessarily photoconsistent. The oversegmentation and projected depth allow us to identify poorly reconstructed superpixels in each image. Depth synthesis fills in poorly reconstructed superpixels using depth from "similar" superpixels of the image. We create a graph structure with superpixels as nodes and define a careful traversal of the graph which allows us to identify best matching superpixels in terms of color and spatial proximity. We keep the three best matching superpixels and interpolate the depth from these superpixels to add a small set of new depth values into the original poorly reconstructed superpixel. These best matches are generally not immediate spatial neighbors. Thus, our depth synthesis is capable of performing non-local interpolation that preserves depth discontinuities provided by the superpixel representation.

The depth synthesis does not augment the 3D reconstruction because the new depth samples are not always photoconsistent. They serves as approximations suitable for plausible image-based rendering within a regularized framework like our shape preserving warp.

Local shape preserving warp Superpixels now contain reconstructed depth from multi-view stereo or *plausible* synthesized depth. The depth samples may be inaccurate or noisy or not photoconsistent; reprojecting them will lead to visible artifacts in rendering. To allow plausible novel views, we perform a *local* shape-preserving warp on each superpixel individually, in contrast to [Liu *et al.*, 2009] and the silhouette-aware warp (Chapter 3) which warp the entire image. Superpixels correspond to well-defined regions of homogeneous color content, and thus give good results with our local shape-preserving warp.

Rendering Rendering is achieved with a three-pass blending algorithm. We first select four input cameras closest to the novel camera, and warp these images to the target view. The four warped images are then blended, with weights specified by camera orientation but also the reliability of depth information in each warped superpixel. Finally, we fill holes with Poisson synthesis [Pérez *et al.*, 2003].

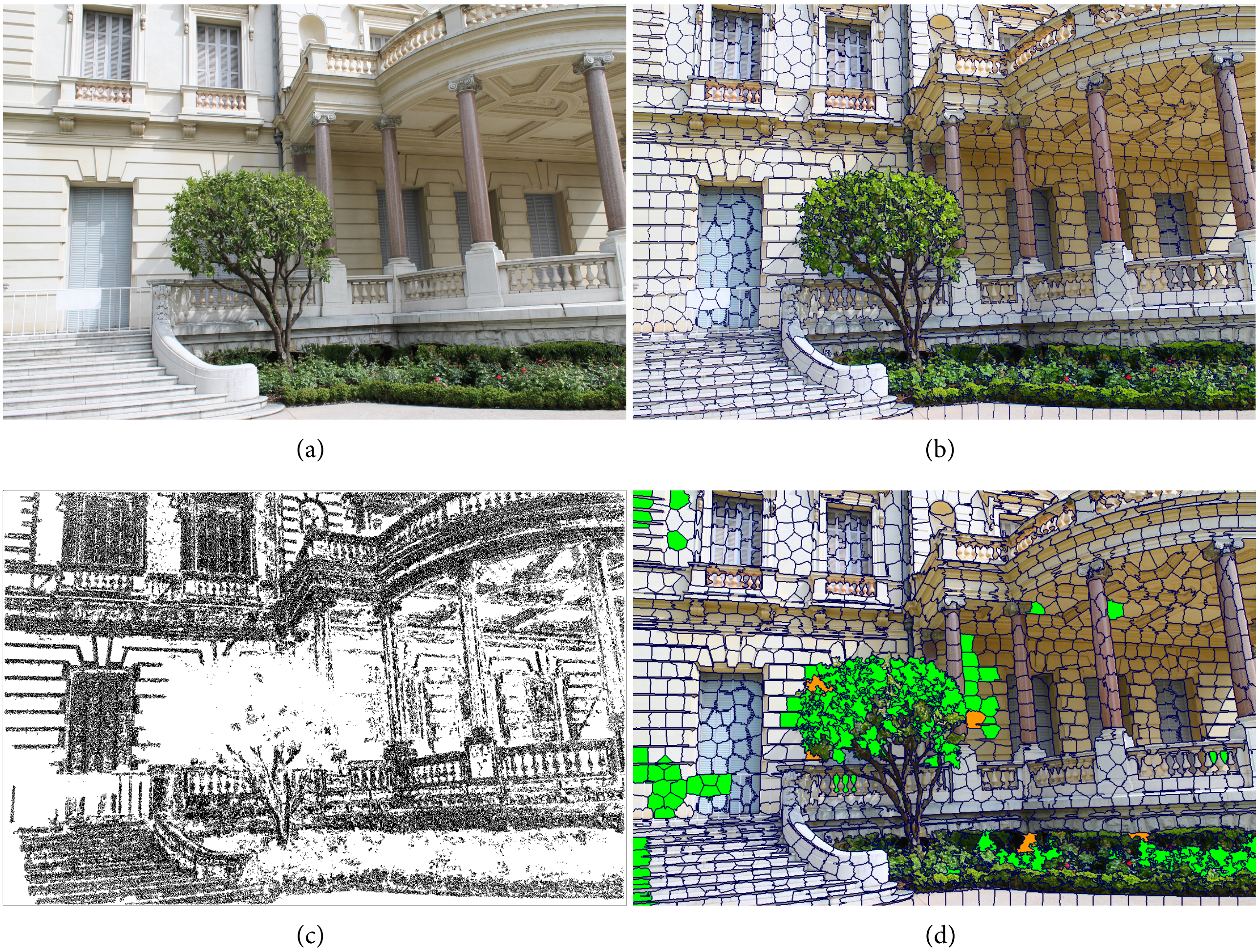
We present an extensive set of example scenes, all containing challenging regions which state of the art multi-view stereo reconstructs poorly. Our algorithm allows plausible navigation for such scenes. We also compare to the two most relevant recent image-based rendering algorithms [Eisemann *et al.*, 2008; Goesele *et al.*, 2010] and the silhouette-aware warp (Chapter 3). Our approach diminishes many of the artifacts of these methods and provides very convincing navigation experiences.

4.2 Depth synthesis

Our input is a set of images of a given scene, taken from different viewpoints. After 3D reconstruction, we use [Achanta *et al.*, 2012] to oversegment each input image, an efficient algorithm that gives superpixels of approximately equal size and with regular shapes (see Figure 4.1(b)), unlike [Felzenszwalb and Huttenlocher, 2004] which gives superpixels of highly irregular shapes and sizes due to lack to spatial compactness.

We denote the set of all superpixels in an image by $\mathcal{S} = \{S_i\}_{i \in \{0 \ldots n-1\}}$. We project the reconstructed 3D points into the image, such that the depth at pixel $\mathbf{x}$ is denoted by $D[\mathbf{x}]$ (shown in Figure 4.1(c)). The set of depth samples inside each superpixel is thus $\mathscr{D}[S_i] = \{\mathbf{x} \in S_i \mid D[\mathbf{x}] > 0\}$. We distinguish two classes of superpixels: those containing less than 0.5% reconstructed pixels, which we call *target superpixels* (shown in green in Figure 4.1(d)) and all others which we consider to have reliable depth.

Our goal is to synthesize plausible depth for a sufficient number of points in each target superpixel. We do this by identifying a set of *source superpixels*, which are spatially close and should ideally belong to the same object in the scene as that of the target superpixel. In addition, our goal is to have a fully automatic algorithm which requires no scene dependent parameter tuning.



(a)

(b)

(c)

(d)

Figure 4.1: (a) Input image, (b) superpixel oversegmentation, (c) projected depth samples, and (d) *target superpixels* marked in green. The superpixels marked in orange could not be assigned depth reliably by our depth synthesis step (Section 4.2.2). These are marked as holes.

There are several ways to achieve this goal; two seemingly straightforward approaches include object classification and interpolation/upsampling of existing depth. Object classification approaches [Andreetto *et al.*, 2008] give remarkable results on some classes of objects, such as man-made structures, animals, humans, etc. However, for cluttered scenes such as ours, which often include vegetation, results can be less reliable. In addition, our experiments with e.g., [Andreetto *et al.*, 2008] indicate very high computation times. Please refer to Appendix A for experiments with state of the art segmentation algorithms.

Interpolation techniques have been used for regions with sufficient depth density (e.g., [Goesele *et al.*, 2010]). For regions with very sparse depth, these techniques result in silhouette flattening and over smooth depth maps which diminish parallax effects during rendering.

We propose an efficient and robust approach which combines image content similarity and spatial proximity in the choice of source superpixels employed to synthesize depth. The irregular shape of su-

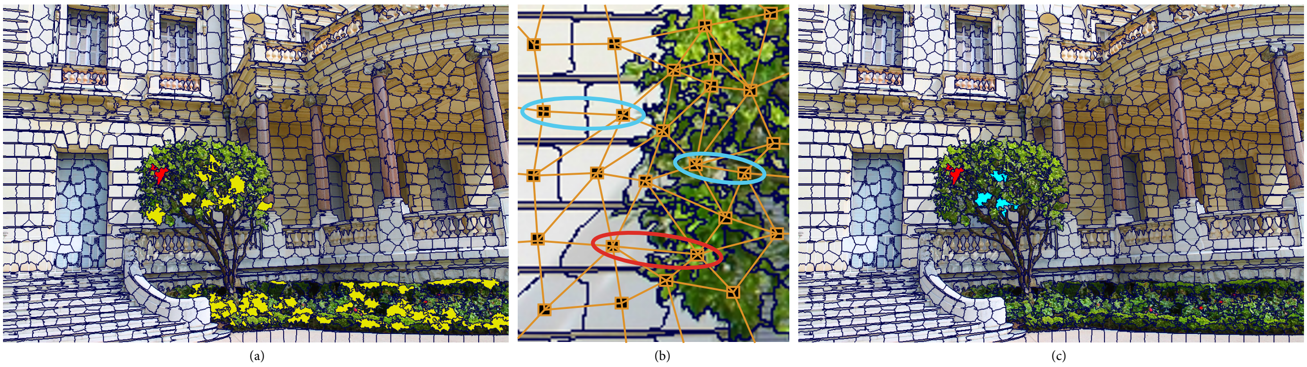


Figure 4.2: Depth synthesis algorithm. (a) Target superpixel (red) and the set of similar neighbors (yellow) in a color-content sense. (b) Superpixel graph computed by treating each superpixel as a node and adding edges between adjacent superpixels with edge length equal to the χ^2 distance between their LAB histograms. The edge length annotated in red is a high because of high color difference while those in cyan are low. (c) The 3 best matches (cyan) selected by the shortest walk algorithm using the superpixel graph.

perpixel boundaries requires definition of appropriate distance metrics and search strategies both for image content and for spatial proximity. We use histogram comparison to identify superpixels with similar image content and a graph traversal approach to provide a robust and parameter-free algorithm. Depth values within target superpixels are synthesized using an interpolation approach based on the distribution of depths in the source superpixels.

4.2.1 Computing similar superpixels

We first compute a set of "most similar" superpixels for each target superpixel. Among many similarity metrics for measuring the affinity of irregular image regions, Grundmann *et al.* [2010] have successfully used χ^2 distance between LAB histograms of superpixels in order to measure color similarity. Other metrics like sum of squared differences (SSD) are less suitable for irregular shapes and sizes of superpixels. Measuring average color of a superpixel performed worse than LAB histogram distance. Therefore, We convert the image into LAB space and create separate histograms for each superpixel with 20 bins in each of L, A and B axes. We concatenate the histograms to give a 60D descriptor $\mathscr{H}[S_i]$ for each superpixel $S_i \in \mathscr{S}$. We compute the nearest neighbors of each target superpixel from all superpixels already containing depth samples using the histogram descriptors space with χ^2 distance metric. This gives a set of "most similar" superpixels $\mathscr{N}[S_i]$. We keep the 40 most similar superpixels, shown in yellow in Figure 4.2(a) for the target superpixel shown in red. We assume that any significant object would be around 5% of image area, equivalent to 40-60 superpixels. We experimented successfully with 40-80 most similar superpixels; higher numbers needlessly increased computation.

4.2.2 Shortest walk algorithm

These neighboring superpixels can belong to very different objects or far off regions of the same object in rich urban scenes. This can occur because of texture-less architecture, stochastic texture (e.g., trees, hedges) or texture repetition (e.g., windows) as shown in Figure 4.2(a). We refine $\mathcal{N}[S_i]$ by selecting the spatially closest superpixels. However, the irregular and highly non-convex shapes of superpixels make Euclidean distance between superpixels very ambiguous. Moreover, the size of the spatial neighborhood is also ambiguous because of the varying sizes of superpixels.

We resolve the above ambiguity using a graph traversal algorithm. We create a 2D superpixel graph by adding edges between any two superpixels which share a common boundary (see Figure 4.2)(b). We assign the edge length between two superpixels as the change in color. We use the χ^2 distance between LAB histograms to measure change in color content. Thus, two superpixels with very similar color content will have a short edge, as annotated in cyan for two superpixels on the wall in Figure 4.2(b). Similarly, any two superpixels on different scene objects are likely to have different color content and thus a high edge length, as annotated in red in Figure 4.2(b). We compute the shortest path between *target superpixel* S_i^T and each *source superpixel* $S_j \in \mathcal{N}[S_i^T]$, which denotes the path between the two superpixels involving the least change in color along the path. This path is computed by minimizing the path cost C over all possible paths from S_i^T to S_j.

$$C(S_i^T \xrightarrow{\gamma} S_j) = \sum_{t=1}^{|\gamma|-1} d(\mathscr{H}[\gamma(t)],\ \mathscr{H}[\gamma(t+1)]) \tag{4.1}$$

$$\tilde{C}(S_i^T \to S_j) = \min_{\gamma \in \Gamma[S_i^T \to S_j]} C(S_i^T \xrightarrow{\gamma} S_j) \tag{4.2}$$

where $\Gamma[S_i^T \to S_j]$ is the set of all paths from target superpixel S_i^T to S_j, γ is one such path of length $|\gamma|$ such that $\gamma(0) = S_i^T$ and $\gamma(|\gamma|) = S_j$, $C(S_i \xrightarrow{\gamma} S_j)$ is the cost of path γ, and $d(\cdot,\cdot)$ is the χ^2 distance between histograms. We implement the above using the Dijkstra shortest path algorithm where the edge weight between two superpixels is the χ^2 LAB histogram distance.

We compute $\tilde{C}(S_i^T \to S_j)$ for all $S_j \in \mathcal{N}[S_i^T]$ and choose a set of three superpixels $\tilde{\mathcal{N}}[S_i^T]$ with the smallest path costs. We then plot the histogram of depth samples contained in $\cup S_k \in \tilde{\mathcal{N}}[S_i^T]$. A single strong peak in the depth histogram or two contiguous peaks (see Figure 4.3(a),(c)) indicate that all $S_k \in \tilde{\mathcal{N}}[S_i^T]$ are at similar depths and can be reached from S_i^T without crossing color discontinuities, which means that the superpixels are likely to belong to the same object. We obtained similar results for 3-6 superpixels with smallest paths costs; numbers higher than 6 often gave multiple peaks in the depth histogram e.g. Figure 4.3(d). If the final depth histogram has more than two peaks or split peaks (see Figure 4.3(d)), then the superpixels selected by our shortest walk algorithm most likely belong to different scene objects. We ignore such superpixels for the moment. We use an iterative approach:

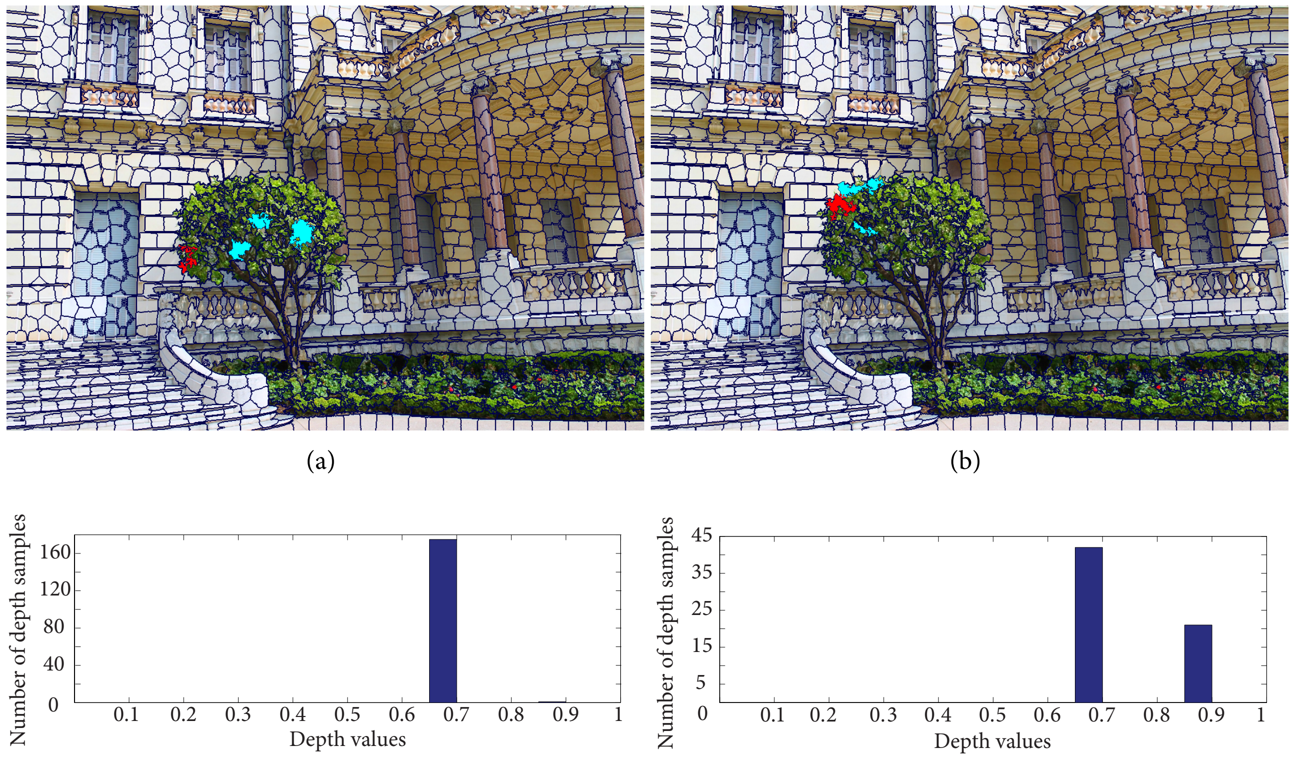

(a) (b) (c) (d)

Figure 4.3: Top: target superpixel in yellow and the *source superpixels* $\tilde{\mathcal{N}}[S_i^T]$ in blue. Bottom: corresponding depth histograms of $\tilde{\mathcal{N}}[S_i^T]$. Depth histogram for the first has a single peak indicating reliable depth. Split peaks in the second indicate that *source superpixels* have depth from a different scene objects. This is true for the source superpixels at the tree silhouette which contains 3D points from the wall behind the tree (see Figure 4.4(left)).

superpixels filled in a previous iteration are used to add depth to remaining superpixels in the next iteration. The algorithm stops when no more superpixels can be assigned depth samples. If no pixels of a particular scene object were originally reconstructed, the superpixels of such an object will find source superpixels from other objects and the final depth histogram is most likely to remain unreliable. We discard superpixels with multiple split peaks and mark them as holes (see Figure 4.1(d)).

Note that we could incorporate spatial distance and Lab histogram distance in a single metric by weighing them appropriately, but this would involve tuning the weights carefully for each dataset depending on image content, object shapes, etc.

4.2.3 Interpolating depth samples

We now interpolate depth samples from the *source superpixels* $\tilde{\mathcal{N}}[S_i^T]$. We create the combined histogram of depth samples from all source superpixels. We then create the joint probability distribution of depth samples by normalizing the histogram bin size by the total area under the histogram. This gives the approximate probability density function (PDF) of depth samples. Using the PDF as interpolation

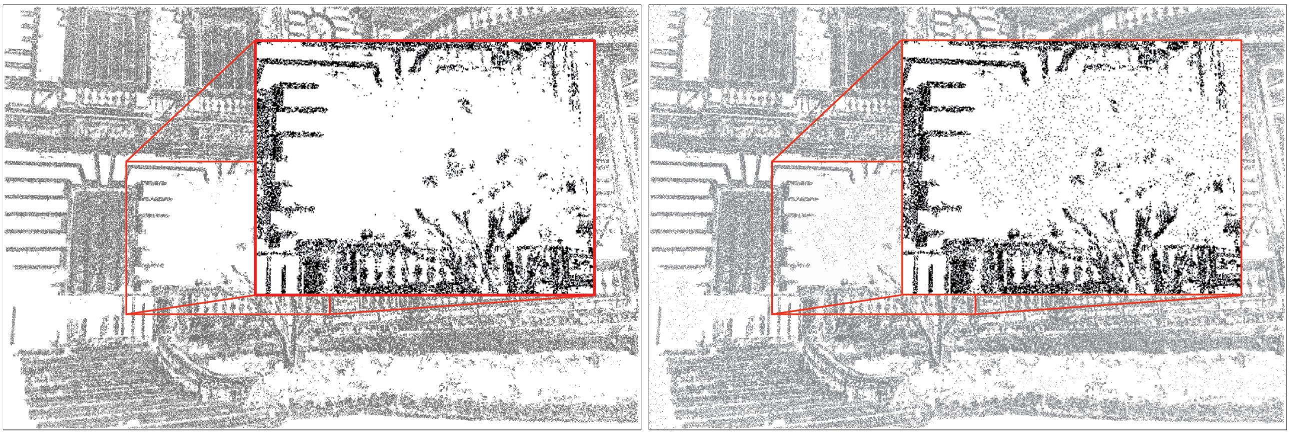


Figure 4.4: Our depth synthesis adds samples with plausible depth (right) values to poorly reconstructed regions shown in the left figure (and Figure 4.1(c)).

weights automatically attenuates the effect of noisy depth samples. We interpolate the inverse of depth values, as depth is inversely proportional to disparity [Goesele *et al.*, 2010]. The final inverse depth at pixel $\mathbf{x}$ of S_i^T is given by:

$$\frac{1}{D[\mathbf{x}]} = \frac{\sum\limits_{S_k \in \tilde{\mathcal{N}}[S_i^T]} \left(\sum\limits_{\mathbf{y} \in \mathcal{D}[S_k]} P(D[\mathbf{y}]) \cdot \|\mathbf{x} - \mathbf{y}\|^{-2} \cdot D^{-1}[\mathbf{y}] \right)}{\sum\limits_{S_k \in \tilde{\mathcal{N}}[S_i^T]} \left(\sum\limits_{\mathbf{y} \in \mathcal{D}[S_k]} P(D[\mathbf{y}]) \cdot \|\mathbf{x} - \mathbf{y}\|^{-2} \right)} \tag{4.3}$$

We add 5-10 depth samples at random pixels in S_i^T. The result for the example in Figure 4.1(c) is shown in Figure 4.4. We got similar results for 5-50 depth samples; higher numbers increased the size of the warp optimization.

Furukawa and Ponce [2009], like any multi-view stereo approach, do not reconstruct sky regions. We identify such regions using the approach described in Appendix B and assign them 99th percentile depth of the image before applying the above depth synthesis. This is an optional step required if there are significant sky regions.

4.3 Local warping of superpixels with depth samples

Depth samples from multi-view stereo can be noisy, especially near silhouettes. In addition, our synthesized depth is only *plausible* rather than photoconsistent or accurate. Consequently, direct reprojection of superpixels using these depth samples, e.g., using the Video Mesh data structure [Chen *et al.*, 2011],

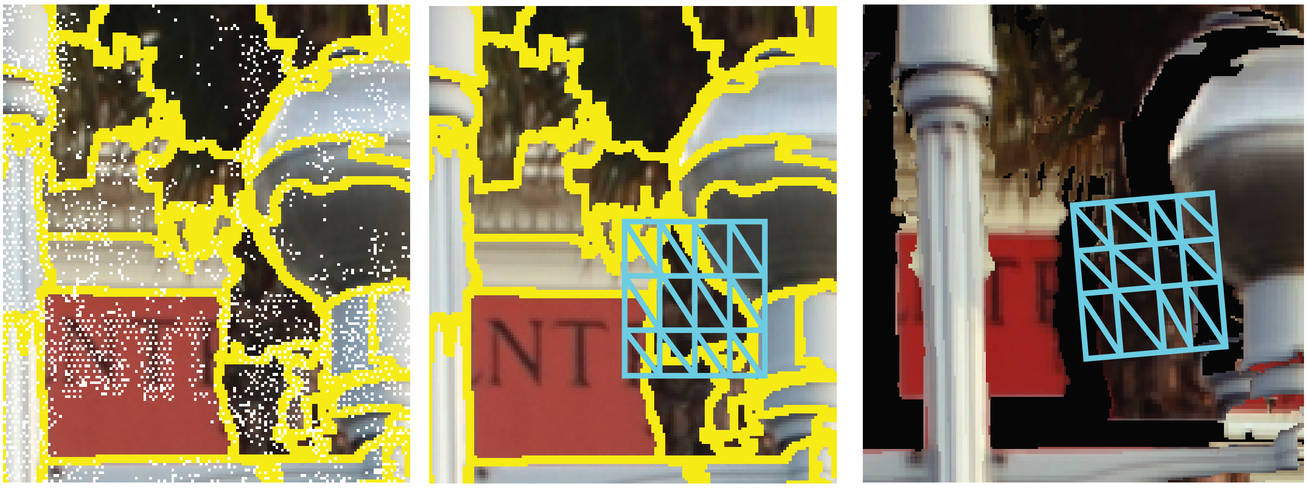


Figure 4.5: Left: Superpixel segmentation showing superpixels at multiple depths as well as depth samples contained inside each superpixel (shown as white dots). Middle: The regular grid which is used as warp mesh, overlaid over each superpixel. Right: Warped superpixels and grid for a novel view. Warping each superpixels independently preserves all silhouettes. Note how background superpixels slide under foreground.

will result in disturbing artifacts. We demonstrate these problems in the Section 4.6.

To alleviate these problems, we adopt a variational warp approach to regularize the final effect of depth samples. In contrast to previous methods [Liu *et al.*, 2009] and Section 3, we do not warp the entire image, but perform an individual local warp for each superpixel, which allows much more freedom to navigate in the scene and reduces some artifacts (see Figure 4.11 and 4.12).

At each frame, we warp each superpixel of each image *individually* to the novel view with projection matrix C_N. Our warp satisfies two energy terms in a least-squares sense: a *reprojection energy* at each depth sample that is reprojected into the novel view, and a *shape-preserving energy* or regularization term for each warp mesh triangle that preserves the shape of the superpixel during the warp.

We create an axis-aligned bounding box for each superpixel and overlay a regular grid which serves as the warp mesh (see Figure 4.5, middle). Each grid triangle contains zero or more depth samples. The unknowns in the warp optimization are the warp mesh vertex positions $\tilde{\mathbf{v}}$. Our variational warp energy is similar to Equation 3.6, except that we do not have any silhouette constraints and superpixel is warped separately rather than warping the entire image, which makes this formulation a local warp.

Reprojection energy For each depth sample $D[\mathbf{x}]$, we locate the triangle T of the warp mesh that contains the depth sample. Denote the vertices of T by $(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3)$ and let the barycentric coordinates of the location of the depth sample at pixel $\mathbf{x}$ in triangle T be (α, β, γ):

$$\mathbf{x} = \alpha \cdot \mathbf{v}_1 + \beta \cdot \mathbf{v}_2 + \gamma \cdot \mathbf{v}_3 \tag{4.4}$$

The reprojection energy measures the distance between the warped position of the depth sample and the reprojected location using the novel view matrix C_N:

$$E_p[\mathbf{x}] = \|\alpha \cdot \tilde{\mathbf{v}}_1 + \beta \cdot \tilde{\mathbf{v}}_2 + \gamma \cdot \tilde{\mathbf{v}}_3 - C_N \cdot C_{I_i}^{-1} \cdot D[\mathbf{x}]\|^2 \tag{4.5}$$

where $C_{I_i}^{-1}$ is the backprojection matrix of image I_i.

Shape-preserving energy For each triangle of the warp mesh with vertices $(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3)$, this energy term measures its shape distortion after the warp. Ideally the triangle only undergoes a similarity transformation, resulting in a null energy value. The similarity energy is obtained by expressing one vertex of the triangle as a linear combination of the other two as in Equation 3.4:

$$E_s[T] = \|\tilde{\mathbf{v}}_3 - \big(\tilde{\mathbf{v}}_1 + a \cdot (\tilde{\mathbf{v}}_2 - \tilde{\mathbf{v}}_1) + b \cdot R_{90} \cdot (\tilde{\mathbf{v}}_2 - \tilde{\mathbf{v}}_1)\big)\|^2, \tag{4.6}$$

where a and b are the same as Equation 3.3:

$$\begin{aligned} a &= \frac{(\mathbf{v}_3 - \mathbf{v}_1)^T \cdot (\mathbf{v}_2 - \mathbf{v}_1)}{\|\mathbf{v}_2 - \mathbf{v}_1\|}, \\ b &= \frac{(\mathbf{v}_3 - \mathbf{v}_1)^T \cdot R_{90} \cdot (\mathbf{v}_2 - \mathbf{v}_1)}{\|\mathbf{v}_2 - \mathbf{v}_1\|}. \end{aligned} \tag{4.7}$$

Here, R_{90} is 90° rotation. The overall energy function for the superpixel warp is given by

$$E[S_k] = w_p \sum_{\forall \mathbf{x} \in \mathscr{D}(S_k)} E_p[\mathbf{x}] + w_s \sum_{\forall T \in \mathscr{T}(S_k)} E_s[T], \tag{4.8}$$

where $\mathscr{D}(S_k)$ is the set of all depth samples and $\mathscr{T}(S_k)$ is the set of all warp mesh triangles in superpixel S_k. We use $w_p = 4$ and $w_s = 1$ in all our experiments.

We minimize $E[S_k]$ for each superpixel by building a sparse linear system and solving it using Cholmod [Chen *et al.*, 2008] on the CPU. We solve thousands of small independent local warps in parallel, which is faster than a single global warp as in [Liu *et al.*, 2009] and silhouette-aware warp. We compare to silhouette-aware warp in Section 4.6 and also discuss the effect of the shape-preserving warp as compared to methods which reproject depth samples directly [Chen *et al.*, 2011].

4.4 Rendering

Rendering is achieved in three passes. In the first pass, we select and warp four input cameras closest to the novel camera. Next, we blend the resulting warped superpixel images to synthesize the novel view.

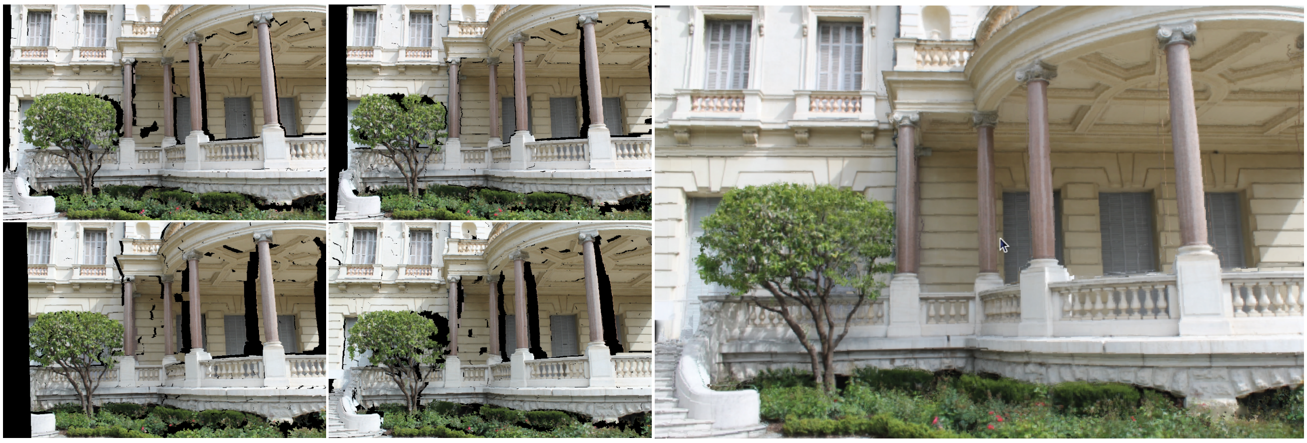


Figure 4.6: Warped superpixel images (left four) and final result after blending (right).

A final hole-filling pass completes the rendering algorithm.

4.4.1 Pass 1: Camera selection and warping

For each novel view, we select the four input cameras closest to the novel camera based on camera positions. We warp the superpixels of each of these images as described previously and render the warped superpixels of each image in a separate floating point render target with depth test enabled. We reproject the median depth of a superpixel[1] into the novel view and use it for the depth test. The warp mesh of each superpixel is rendered with an alpha matte defined by the outline of the superpixel. We use a soft alpha matte by rendering an additional 4 pixel wide zone outside the superpixel boundary if the neighboring superpixel's median depth is almost the same as the current superpixel. This fills in small cracks between warped superpixels, if any. We store the reprojected median depth and the superpixel ID of each warped superpixel in an additional render target while warping. These are used in the next pass to compute blending weights. This gives us four warped images where occluded background superpixels slide under foreground superpixels and disocclusions create holes in the warped images (see Figure 4.6).

4.4.2 Pass 2: Blending

We render a screen-size quad into the frame buffer and blend the four warped images to get the final result in the pixel shader. At runtime, each warped image contributes one candidate for blending. We also upload additional metadata for each warped image: median depth of each superpixel as well as superpixel identifier. We then compute the blending weight for each of the four candidates using an approach very similar to silhouette-aware warp (see Section 3.5).

[1] computed as median of all depth samples contained within the superpixel.

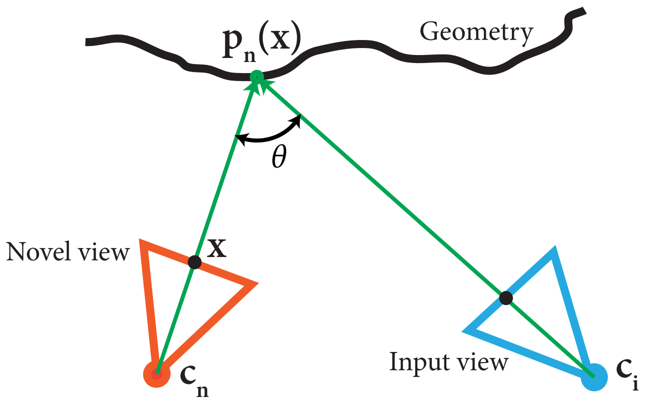


Figure 4.7: Angle θ used for computing the penalty $P_{\text{ang}}(\tilde{I}_i, \mathbf{x})$

Blending weights For each candidate, we compute the penalty and hence the blending weight. Consider pixel $\mathbf{x}$ of the novel view with center of projection $\mathbf{c}_n$ and a warped input image $\tilde{I}_i$ whose center of projection is $\mathbf{c}_i$. Let $\mathbf{p}_n(\mathbf{x})$ be the point where the ray shot from $\mathbf{c}_n$ through pixel $\mathbf{x}$ intersects the scene geometry. This point can be computed simply by backprojecting the median depth of the superpixel using inverse camera projection matrices of the image I_i. We use the angle between $(\mathbf{c}_n - \mathbf{p}_n(\mathbf{x}))$ and $(\mathbf{c}_i - \mathbf{p}_n(\mathbf{x}))$ as an angle penalty (see Figure 4.7). We also define a field-of-view penalty that checks whether the pixel lies in a disoccluded hole region of the warped image which appear as black regions in Figure 4.6(left).

$$P_{\text{ang}}(\tilde{I}_i, \mathbf{x}) = \arccos\left(\langle \mathbf{c}_n - \mathbf{p}_n(\mathbf{x}),\ \mathbf{c}_i - \mathbf{p}_n(\mathbf{x})\rangle\right) \tag{4.9}$$

$$P_{\text{fov}}(\tilde{I}_i, \mathbf{x}) = \begin{cases} \infty & \text{if } \mathbf{x} \text{ is in a disoccluded hole} \\ 0 & \text{otherwise} \end{cases} \tag{4.10}$$

The final penalty is given by the sum of the two penalty terms P_{ang} and P_{fov}. The blending weight is computed as the inverse of the penalty. We use these weights to select the best two candidates for blending, consistent with the heuristics from Section 5.2.4 that blending two images at each pixel gives the best visual quality.

Adaptive blending heuristics using superpixel correspondence We use an adaptive blending scheme by creating a superpixel correspondence graph across images. We add a correspondence edge between two superpixels from different images if they share 3D reconstructed points. Superpixels with correspondence edges are very likely to belong to the same part of the same scene object. We thus obtain a list of corresponding superpixels for each superpixel of each image and upload this additional data to the pixel shader. At runtime, if two pixels to be blended come from superpixels that have a correspondence edge, they are blended with the blending weights computed above. Since they are quite likely to belong to the same scene object, blending is unlikely to result in ghosting artifacts.

On the other hand, if the two candidates to be blended come from superpixels that do not have a correspondence edge, they are likely to belong to different scene objects in which case blending them can lead to ghosting artifacts. This can occur if one of the candidates comes from a superpixel with synthesized depth (see Section 4.2); in this case, we increase the blending weight of the other by a factor of 2.0. This is because synthesized depth is obviously less reliable than photoconsistent depth given by multiview stereo. In the other case that both candidates are from superpixels that have synthesized depth, we use the heuristic that it is better to display incorrect parallax on background regions; background parallax errors being less noticeable than those in the foreground. We therefore increase the blending weight of the pixel with the higher depth value by a factor of 2.0.

Recall that we scaled the weights of highest weighted candidate in the silhouette-aware warp (see Section 3.5) by an additional factor Ψ to reduce excessive blending. Our adaptive blending heuristics here extend that idea by making it content-sensitive. This allows our approach to avoid blending when it anticipates ghosting artifacts, using the guideline from Section 5.2.4 that ghosting artifacts are more objectionable than temporal popping. The adaptive approach inherits the advantages of blending in most regions, namely temporal coherence, but favors popping in regions where blending is expected to lead to ghosting artifacts. This results in a better tradeoff between the two types of artifacts.

Our tests showed that the above factor of 2.0 gave satisfactory results on our datasets; values higher than 4.0 effectively disable blending.

4.4.3 Pass 3: Hole filling

Moving the novel view significantly away from input cameras creates large disoccluded regions which are not captured by any of the input images. Such regions appear as holes; we use Poisson synthesis [Pérez *et al.*, 2003] for basic hole filling. We compute the divergence map from the blended result above using zero gradient value at the holes and hole boundaries. We use all pixels which are not in the holes as Dirichlet boundary conditions and solve the multigrid Poisson synthesis with 5 levels and 2 Jacobi iterations at each level. This creates blurred texture in the holes which become noticeable only when the viewpoint is moved very far away from input cameras as shown in Figure 4.14(c).

4.5 Results

We present the results of our approach on a wide variety of datasets, including scenes captured by ourselves and by others. The School dataset[2] is from Microsoft Photosynth. ChapelHill1 and ChapelHill2

[2] http://photosynth.net/view.aspx?cid=aaeb8ecf-cfef-4c03-be42-bc1ae2f896c0

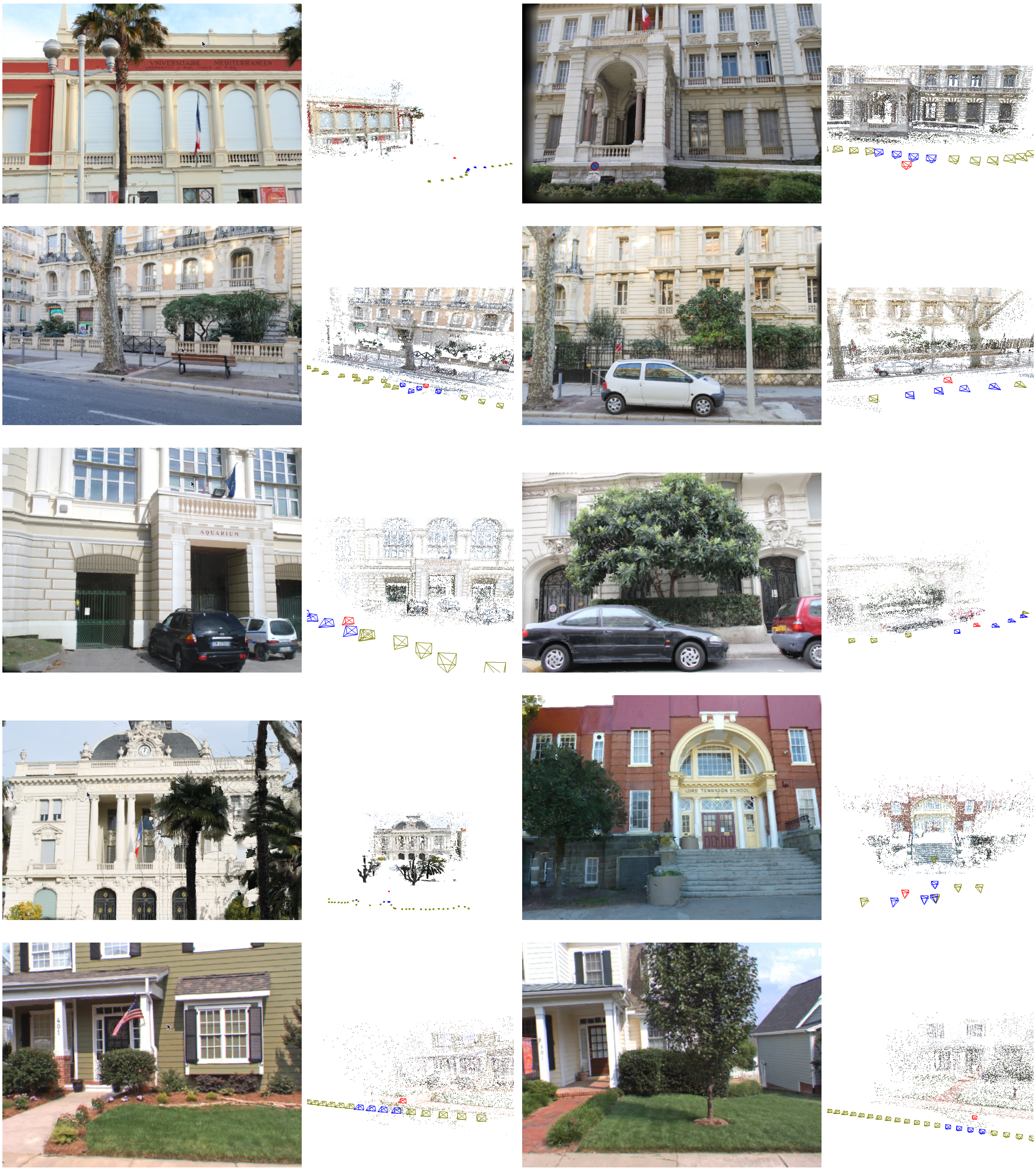


Figure 4.8: A single frame and corresponding top view of the scene for some of the datasets. From top to bottom, University, Museum2, VictorHugo1, VictorHugo2, Aquarium-20, Street-10, Commerce, School, ChapelHill1 and ChapelHill2 datasets. The top view shows the input cameras in yellow, novel camera in red and the 4 images selected for generating the novel view in blue.

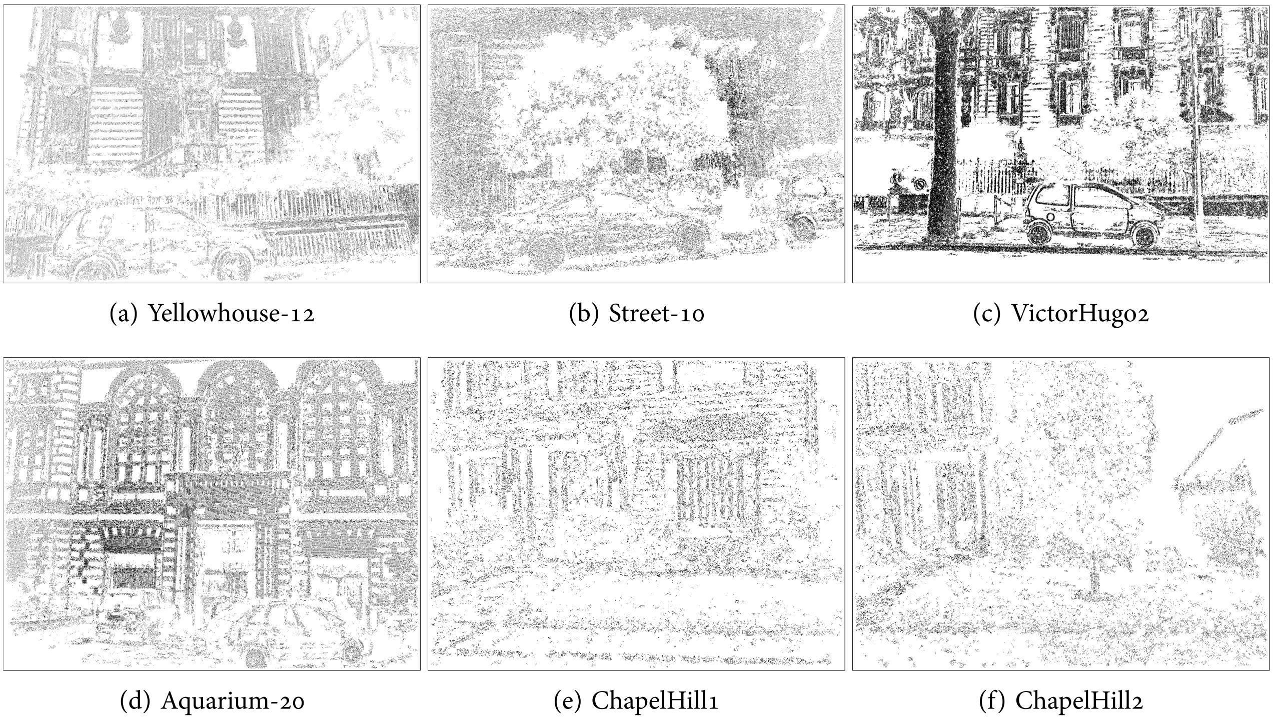
Scene	No. of images	Depth synthesis time (seconds per image)
Museum1	27	46
Museum2	30	66
University	28	75
Yellowhouse-12	12	51
ChapelHill1	25	126
ChapelHill2	30	136
Aquarium-20	20	41
Street-10	10	23
VictorHugo1	24	57
VictorHugo2	25	50
Commerce	35	152
School	36	120

Table 4.1: Depth synthesis running times

are from the street-level capture in [Pollefeys *et al.*, 2008]; we sub-sampled the video stream to simulate a sparse casual photo capture. Aquarium-20, Street-10 and Yellowhouse-12 are the same as those in silhouette-aware warp. We additionally present six new scenes: Museum1, Museum2, University, VictorHugo1, VictorHugo2 and Commerce. We show synthesized views for viewpoints which are quite far from input cameras in Figure 4.8 (see video[3]). We list the number of images and running times for depth synthesis for all the datasets in Table 4.1. Only 10 to 35 images are required for all our scenes. Depth synthesis running times are reported for an MATLAB implementation with no performance optimization which could be accelerated by an order of magnitude by running multiple images of the dataset in parallel on separate cores. Multi-view stereo including [Snavely *et al.*, 2006] and [Furukawa and Ponce, 2009] took between 30-60 minutes for all our datasets depending upon the number of images. We modified the oversegmentation source code of [Achanta *et al.*, 2012] to segment multiple images in parallel which gave running times of 1-3 minutes for all the images in any our datasets.

Rendering is real-time with an average frame rate of 53 FPS and 50 FPS at 800×600 and 1280×800 resolutions respectively on a 12-core Intel Xeon X5650 2.67 Ghz CPU with NVIDIA Quadro 6000 GPU running Fedora 16/17/19. Removing Poisson synthesis improves the frame rate by 8-10 FPS. We achieve 23 FPS and 13 FPS respectively on a laptop with a dual-core Intel 2640M 2.80 GHz CPU and NVIDIA GTX 525M GPU running Fedora 16. Our algorithm works well on a variety of different scenes, which all include challenging cases of poorly reconstructed vegetation and other foreground objects (e.g. cars). As shown in Figure 4.9, such regions get very few depth samples from multi-view stereo. Piecewise-planar techniques like [Sinha *et al.*, 2009] tend to ignore these depth samples while finding dominant planes in the scene, while [Goesele *et al.*, 2010] use “ambient point clouds” to produce a non-photorealistic effect. In contrast, our depth synthesis facilitates plausible rendering using just these few points. More often

[3] http://vimeo.com/62038845


(a) Yellowhouse-12

(b) Street-10

(c) VictorHugo2

(d) Aquarium-20

(e) ChapelHill1

(f) ChapelHill2

Figure 4.9: Original reconstructed points for one of the images from some of our datasets. Though architecture is well reconstructed, regions with vegetation or other foreground objects are very poorly reconstructed. Our approach is capable of generating plausible renderings even for such regions.

than not, urban or suburban scenes do contain trees, vegetation and cars; our method thus represents a significant step in making image-based rendering algorithms practical.

4.6 Comparisons

There exists a vast literature on image-based rendering techniques. However, only a few recent solutions target the type of datasets we focus on, i.e., scenes captured with a simple digital camera, in which large regions are very poorly reconstructed.

Overall comparison To evaluate our overall results, we compare our method to three recent approaches. We compare to Floating Textures [Eisemann *et al.*, 2008] using the author's implementation. This approach also requires a 3D model or "proxy" of the scene, which we create using [Kazhdan *et al.*, 2006] from the reconstructed point cloud. We use our own implementation for Ambient Point Clouds [Goesele *et al.*, 2010] and Silhouette-aware warp from Chapter 3. We also implemented the rendering method of [Chen *et al.*, 2011], which is an alternative warp approach based on reprojection, allowing a comparison to our shape-preserving warp.

In Figure 4.10, we compare our view interpolation results for Yellowhouse-12 and Museum1

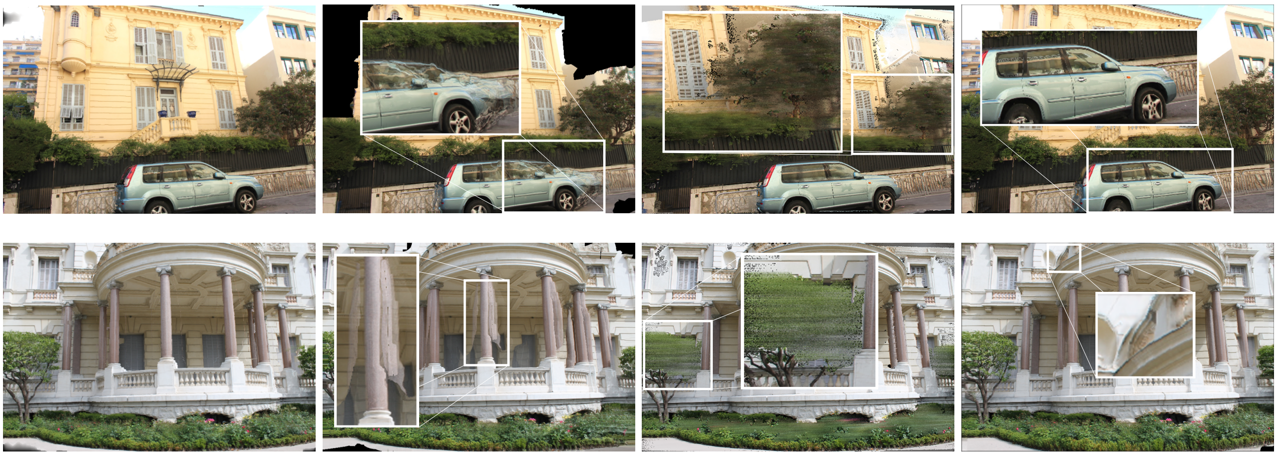
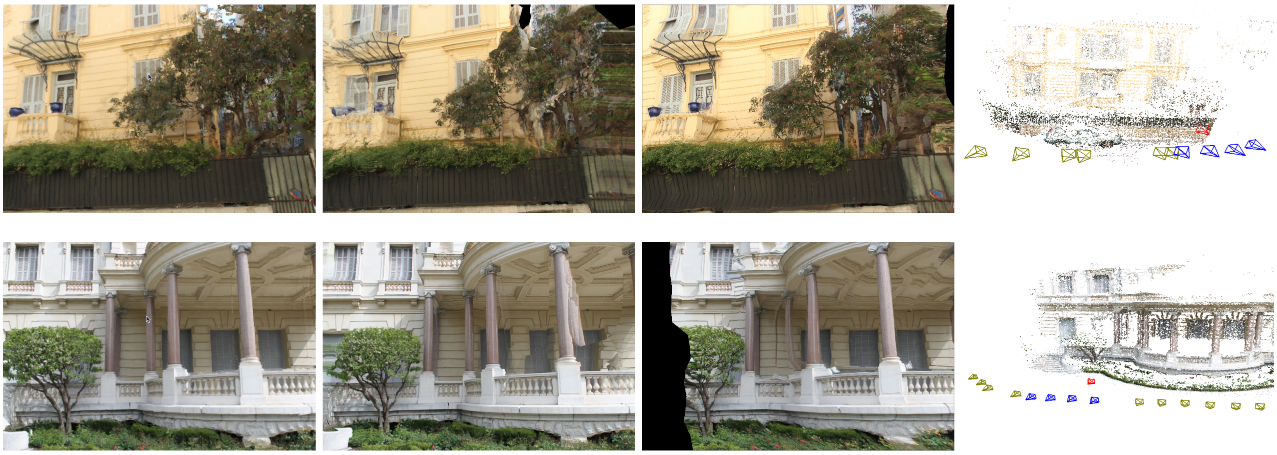

(a) Our result (b) [Eisemann *et al.*, 2008] (c) [Goesele *et al.*, 2010] (d) Silhouette-aware warp

Figure 4.10: View interpolation comparison for the Yellowhouse-12 and Museum1 datasets. [Eisemann *et al.*, 2008] depends on a 3D model and thus shows significant ghosting. In regions with very poor depth (see Figure 4.9), our method is able to create plausible results while [Goesele *et al.*, 2010] creates a smeared point cloud. Silhouette-aware warp gives results similar to ours after 1.5 hours of manual intervention to mark accurate silhouettes and add/correct depth samples, however some distortions are still visible which become much more pronounced away from view-interpolation path (see Figure 4.11).

datasets. Floating textures [Eisemann *et al.*, 2008] have ghosting artifacts because poor or wrong 3D geometry leads to texture misalignment which are too big to compensate by optical flow. [Goesele *et al.*, 2010] use a NPR effect by smearing an ambient point cloud for all poorly reconstructed regions which leads to disturbing artifacts if such regions lie on important scene objects, e.g., cars, trees etc. Our depth synthesis allows plausible novel views even for such regions. Despite the manual silhouette marking, silhouette-aware warp gives distortions in several regions which is even more pronounced if the novel camera is moved away from the view interpolation path, as shown in Figure 4.11 (see video[4]). We do not include [Goesele *et al.*, 2010] in free viewpoint image-based rendering comparison because it is designed only for view interpolation.

The results for Museum1 dataset for silhouette-aware warp in Figure 4.10 and 4.11 required 1.5 hours of manual intervention because a large number of silhouettes had to be marked and depth samples had to be added in large regions such as trees. Even then, the results show a lot of distortion because the global warp diffuses distortions caused by the slightest of depth gradients over the whole image, which become particularly severe when moving away from the view interpolation path (see Figure 4.11). Adding too many intersecting silhouettes into the conformal Delaunay triangulation of silhouette-aware warp leads to numerical issues. In contrast, our method scales to scenes with arbitrary number of silhouettes. Also, the global warp disintegrates when any depth sample of the input image lies behind the novel camera because such a depth sample behind cannot be projected into the novel camera (see Figure 4.12).

[4] http://vimeo.com/62038844



(a) Our result (b) [Eisemann *et al.*, 2008] (c) Silhouette-aware warp (d) Novel camera position

Figure 4.11: Free viewpoint navigation comparison for the Yellowhouse-12 and Museum1 datasets. Our method produces plausible results even for viewpoints quite far from the input images. In contrast, the artifacts of [Eisemann *et al.*, 2008] are clearly visible. The distortions incurred by the global warp in silhouette-aware warp are even more pronounced, despite 1.5 hours of manual intervention.

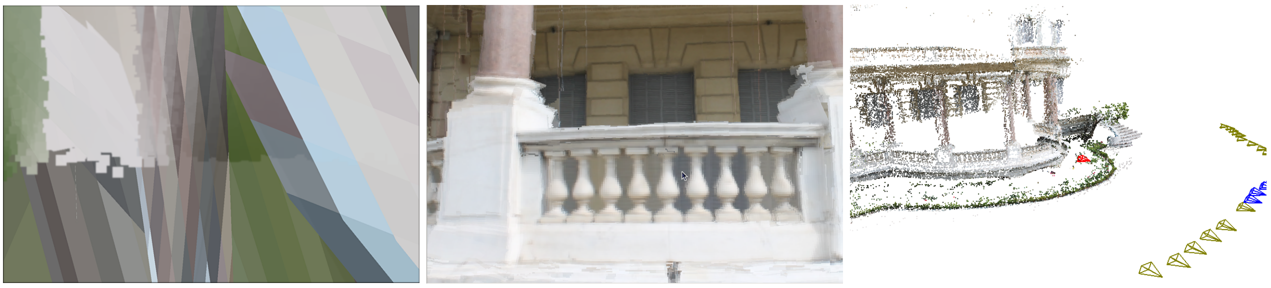


Figure 4.12: Silhouette-aware warp (left) disintegrates if any depth samples is *behind* the novel camera as shown in top view (right). This prevents the user from walking "into" the scene. Our local warp does not suffer from this limitation (middle).

Our local warp simply ignores the superpixels which contain such depth samples, while the rest of the image is warped normally. This makes our approach suitable for potential immersive applications (see Chapter 6).

Comparison with Video Mesh The warp described in Video Mesh [Chen *et al.*, 2011] triangulates and reprojects depth samples directly into the novel view. Inaccurate or outlier depth values can cause the depth sample to be reprojected at incorrect pixel coordinates, causing objectionable artifacts, most noticeable in the form of cracks. Our warp regularizes the effect of noisy depth values and outliers with the shape preserving constraint (see Section 4.3). As a consequence, our results have far fewer cracks (see Figure 4.13).

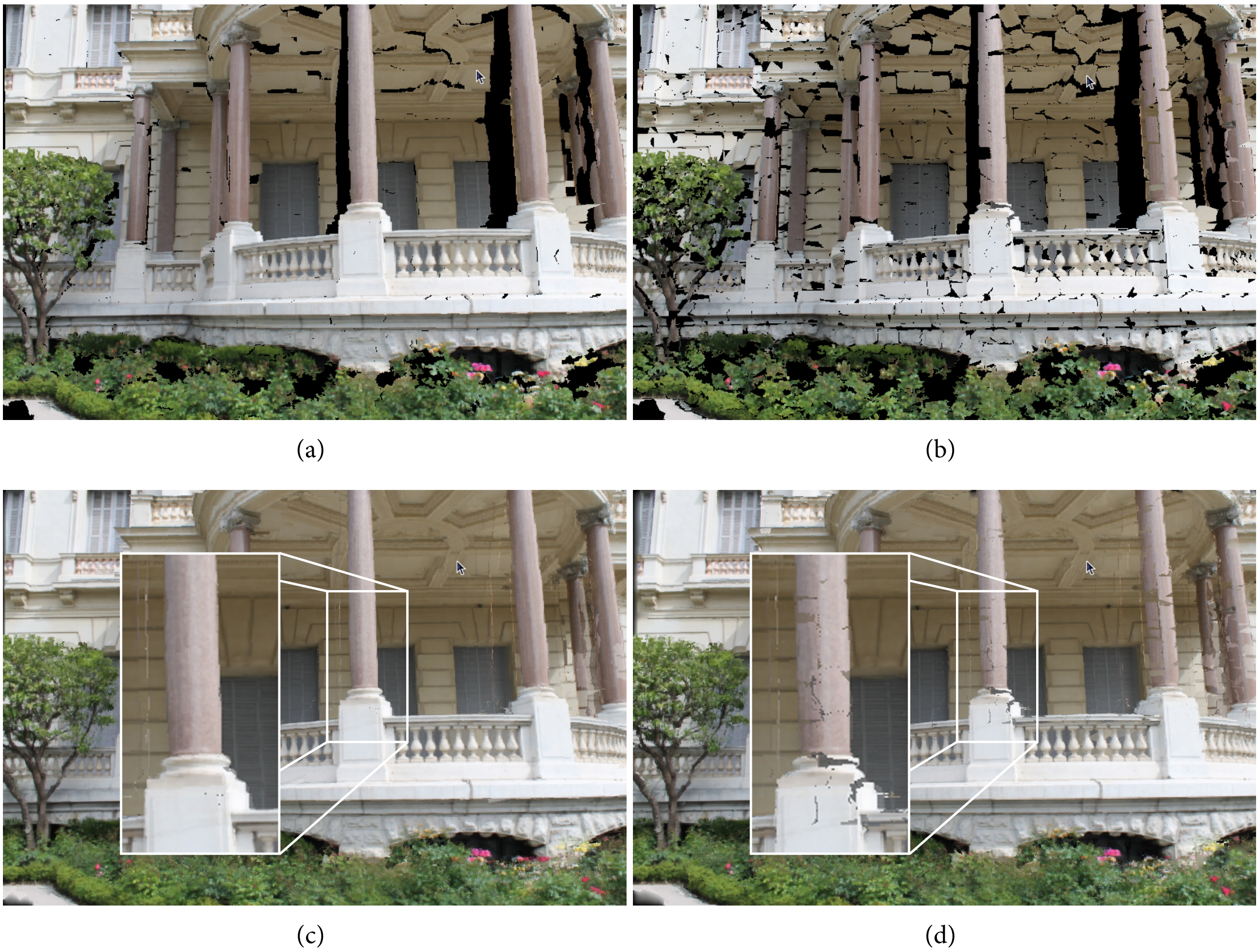


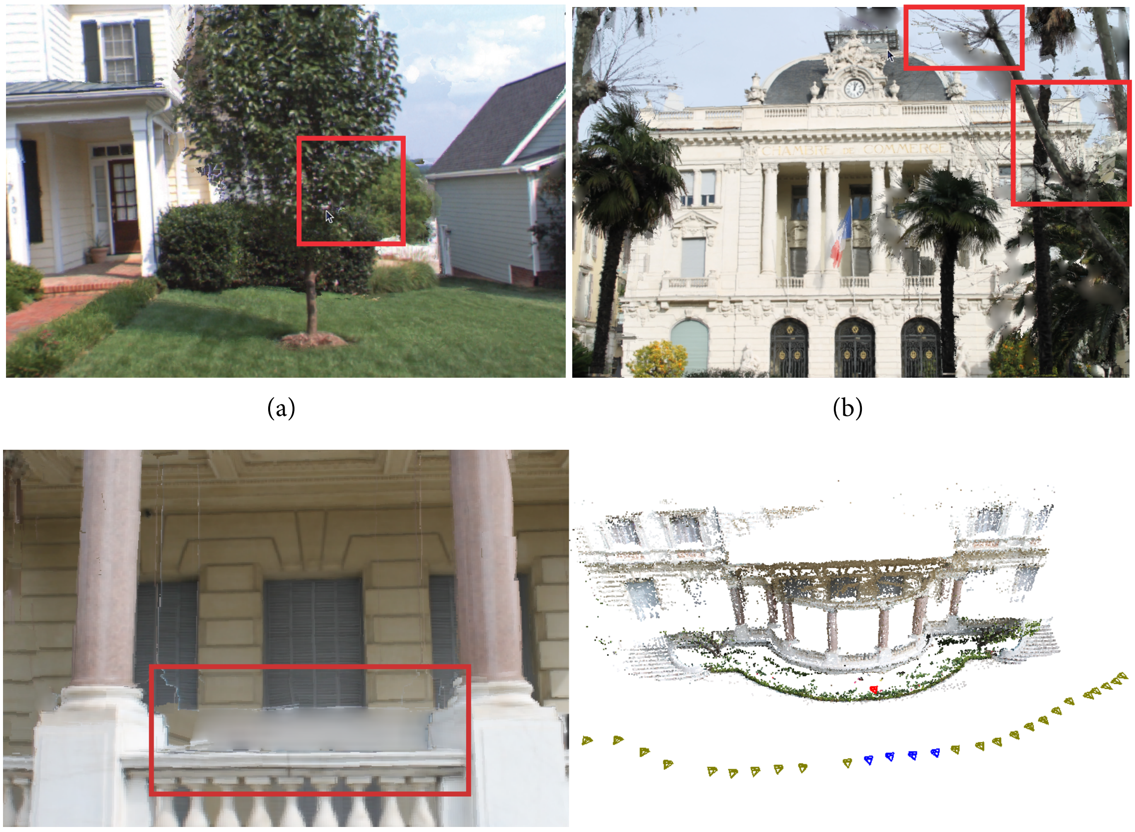
Figure 4.13: (a) Superpixels warped using our approach, (b) superpixels warped using our implementation of Video Mesh [Chen *et al.*, 2011], (c) final result generated from our warped superpixels in (a), (d) final result generated from Video Mesh style warping in (b).

4.7 Limitations

The most important limitation of the depth synthesis is that if the target superpixel corresponds to an object at a depth which does not exist elsewhere in the image, incorrect depth may be assigned from other similar objects. This is shown in Figure 4.14(a), where the background tree is not reconstructed at all and ends up being assigned depth from the foreground tree. The confounding factors are that the trees are spatial neighbors and have extremely similar color/texture to the extent that the boundary between the trees is barely discernible even to the human eye.

The shape preserving warp assumes largely fronto-parallel depth within superpixels. It does not handle surfaces with very sharp depth gradient e.g. surfaces photographed from grazing angles. Such cases are rare though.

Our approach is limited by the capabilities of the oversegmentation: very thin structures cannot be



(a)

(b)

(c)

Figure 4.14: (a) Incorrect depth assignment on the unreconstructed background tree which is barely distinguishable from the foreground tree, (b) very thin structures cannot be properly represented by superpixel and result in rendering artifacts, and (c) hole filling in disoccluded regions not captured in input images results in blurring.

captured (see Figure 4.14(b)). Our hole filling approach is very basic; we resort to blurring in disoccluded regions which are not captured in any input image (see Figure 4.14(c)). This could be replaced by more sophisticated inpainting [Criminisi *et al.*, 2003]. However, such inpainting approaches are far from real time.

4.8 Conclusions

We have presented a free viewpoint image-based rendering algorithm designed for urban environments, capable of producing high quality results in the absence of accurate 3D reconstruction. We compensate for the lack of accurate 3D reconstruction for challenging cases by synthesizing plausible depth, which is not necessarily photoconsistent. Our shape-preserving warp and rendering pipeline use the synthesized depth to produce high quality novel views. We compare our results to four recent image-based rendering methods and demonstrate that our approach extends very well to free viewpoint navigation.

This makes our approach suitable for interactive walkthrough applications such as head tracked virtual reality system, an early prototype of which is also demonstrated in Chapter 6.

Chapter 5

Evaluation of Image-based Rendering using Perceptual Studies

Quantitative analysis of image-based rendering results is very difficult because of the sheer number of variables involved: scene complexity, quality of 3D geometry, viewpoints, color/texture content of scene objects, size of scene, capture density, capture setup etc. Previous work used *leave-one-out* tests [Fitzgibbon *et al.*, 2005] where the image statistics of a simulated novel view are compared to a real photograph from the same viewpoint. Such quantitative tests are relevant in very small baseline view interpolation [Zitnick *et al.*, 2004; Fitzgibbon *et al.*, 2005; Mahajan *et al.*, 2009] where the approach is expected to produce physically correct results. However, most modern image-based rendering systems, like ours, target *plausible* or *good looking* results because the input data is far too sparse to simulate physically correct results. Moreover, different kinds of artifacts are observed simultaneously in image-based rendering results. Quantitative evaluation such as leave-one-out tests do not give any insight into the relative severity of different types of artifacts; this knowledge has a strong bearing on the design of such systems.

Perceptual studies are much more useful for evaluating plausible results. Even though image-based rendering research has advanced by leaps and bounds over the last decade, perceptual analysis has lagged behind. Psychophysical experiment design is non-trivial in this case: the effect of different variables on the final result as well as their relative importance are unclear. Moreover, there exists a wide array of accepted methodologies for conducting user studies; selecting the appropriate one is fairly hard.

In this chapter, we attempt to analyze artifacts using perceptual studies. We study the most ubiquitous artifacts in isolated settings for two cases: perspective distortions, caused when images captured from one viewpoint are projected on to 3D geometry from another viewpoint, and ghosting artifacts caused by blending pixels from multiple photographs to synthesize a novel view.

To achieve meaningful results in such perceptual studies, we need to simplify the case under ob-

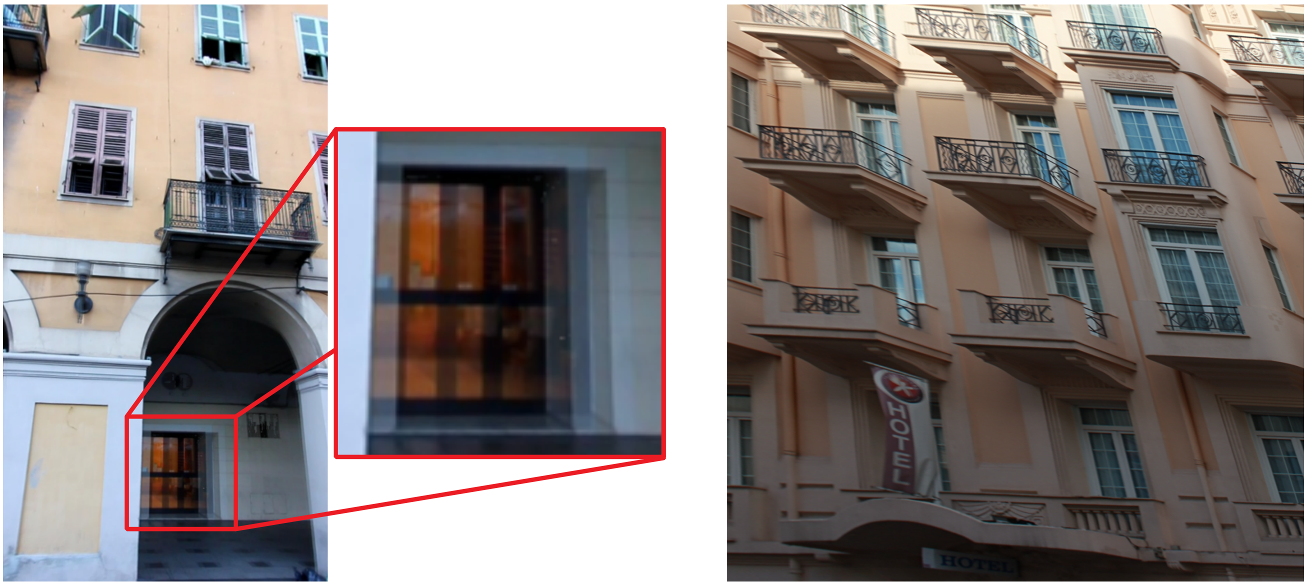


Figure 5.1: Examples of two of common artifacts in image-based rendering, namely ghosting (left) and perspective distortion (right).

servation and isolate the different factors that affect the degree and nature of artifacts. We design our perceptual studies using a rudimentary form of image-based rendering - projective texture mapping on a planar 3D geometry [Debevec *et al.*, 1996] that approximates the scene, akin to existing visualizations of street-level imagery, e.g., Google Streetview, Bing Maps, Mappy Urban Dive etc. While exact details of these systems are not always available, they appear to use panoramic images captured at discrete points along a path, and rendered using cross-fading [Sinha *et al.*, 2009] or unstructured lumigraph [Buehler *et al.*, 2001] onto a planar proxy for each façade.

We choose minimal geometry in the form of a single 3D plane and a simple rendering approach which reproduces rendering artifacts with a small number of parameters. This allows us to analyze the artifacts without introducing any bias towards parameters of the system. In practice, sophisticated systems are designed using the guidelines from simpler systems. We expect any practical image-based rendering system to be more sophisticated than the setup in our studies. Nonetheless, our studies provide useful guidelines for sophisticated systems like those in previous chapters. In their current form, our studies are directly applicable to current large scale urban visualization systems such as Google Streetview. These applications are explained in Sections 5.1.6 and 5.2.4. The main goal of the two studies presented in this chapter is to understand the perception of artifacts and provide practical guidelines that can motivate capture or rendering parameters.

Overview In the first study, we analyze perspective distortions by studying the perception of right angled protrusions on façades such as balconies. We show participants the distorted images of corners and balconies generated by image-based rendering, and ask them to specify the perceived angle in one experiment and rate how close the perceived angle is to a right angle in another experiment. The data from the first experiment allows us to map the capture and viewing parameters to the level of perceived distortions. We validate our results by means of another experiment. In the second study, we analyze artifacts caused by blending or transitioning between multiple images in image-based rendering. We ask participants to rate novel views synthesized using different levels of blending between input images. This allows us to develop guidelines for ensuring ideal levels of blending which keep spatial blurring and temporal discontinuities at acceptable levels.

Contribution For the study of perspective distortions in Section 5.1, the contribution of the thesis is in the form of:

- part of stimuli generation for the experiments (Section 5.1.1),
- validation study (Section 5.1.5), and
- applications to image-based rendering (Section 5.1.6).

The core theory based on vision science, experiment design (Section 5.1.1) and statistical analysis (Section 5.1.4) are beyond the scope of the thesis, please refer to [Vangorp *et al.*, 2013] for details. These are explained to provide context.

For the study in Section 5.2, the contribution of this thesis is in the form of:

- part of conceptual design of the experiments (Sections 5.2.2 and 5.2.3), and
- generation of real world image-based rendering stimuli for both experiments,
- final inferences from experimental data (Section 5.2.4).

The experiment user interface where multiple stimuli are shown to the participant (Figure 5.15) and statistical analysis of experimental data is beyond the scope of thesis, please refer to [Vangorp *et al.*, 2011] for details.

5.1 Perception of perspective distortions

Image-based rendering systems reproject photographs captured from one viewpoint into a novel viewpoint. The photograph captures the perspective of the scene only from the original viewpoint; reprojecting it into a novel viewpoint produces perspective errors depending upon the 3D geometry, capture and viewing parameters. Perspective distortions are always present in all image-based rendering systems. These artifacts are sometimes benign and barely noticeable, while elsewhere they can be objectionable. To avoid showing perspective distortions, image-based rendering applications tend to restrict viewing positions close to capture viewpoints. However, this is done in a rather ad-hoc manner because of the lack of principled understanding of perspective errors. A quantitative model that correlates the perception of perspective distortions with capture/display parameters can allow applications to select acceptable zones of navigation or decide the optimal capture strategy that can be expected to give a required level of perceived quality.

In this section, we devise experiments that investigate the perception of perspective distortions. We build the premise of our experiments upon well-established vision science literature which explains the perception of pictures. Vision science has long studied the perception of paintings or photographs that are captured or painted from a certain viewpoint and viewed from different viewpoints, typically in an exhibition or gallery. A key insight in this work is that this is very similar to the case of image-based rendering where photographs captured from one viewpoint are reprojected into another viewpoint. As explained in the following paragraphs, vision science hypotheses are not directly applicable in our context despite the strong intuitive analogy. We therefore extend the hypotheses from picture perception and design two psychophysical experiments using this theory. We perform statistical analysis of experimental data to develop a quantitative predictive model for perspective distortions and its applications in the context of street-level image-based rendering.

The detailed discussion of vision science theory, extension of vision science hypotheses and statistical analysis of experimental data is beyond the scope of this thesis; please refer to Vangorp *et al.* [2013] for details. We introduce the experimental setup and focus on the results, their validation and applications in image-based rendering.

Perception of pictures When a picture is viewed from the same position as the center of projection of the virtual camera that "photographed" the 3D scene, the image formed on the retina of the viewer, known as retinal image, is correct. If the position of the viewer is different from the center of projection, the retinal image is distorted: its perspective cues such as vanishing points are different from those of the original photograph.

The perception of this retinal image is explained by two competing hypotheses in vision science literature: the *scene hypothesis* and the *retinal hypothesis*. The scene hypothesis states that viewers compensate for incorrect viewing position, so the perceptual outcome is much closer to the original 3D scene than dictated by the distorted retinal image. The retinal hypothesis, on the other hand, states that viewers do not compensate for incorrect position; the perceptual outcome is dictated by the distorted retinal image. When viewers are left or right of the center of projection and view the picture and its frame with both eyes, they compensate for their incorrect viewing position, indicating that scene hypothesis might be dominant [Rosinski *et al.*, 1980; Vishwanath *et al.*, 2005]. In other situations, when the slant of a pictured object is nearly perpendicular to the picture surface or when the viewers are too close to or too far from the picture, they do not compensate for the induced image distortions and therefore perceive 3D structure incorrectly [Adams, 1972; Lumsden, 1983; Todorović, 2009; Banks *et al.*, 2009; Cooper *et al.*, 2012].

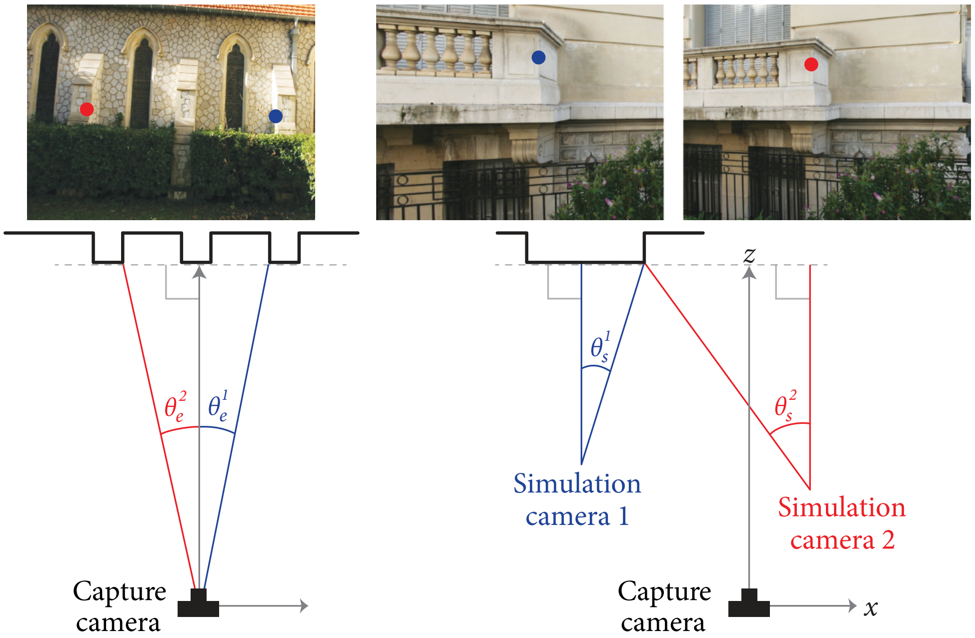


Figure 5.2: Eccentricity and simulation angles. The angles correspond to the corner marked with corresponding color in the photographs. Middle and right: the images shown as simulated view are rendered by reprojecting the input image in the simulated viewpoint.

Extended retinal hypothesis The original retinal hypothesis only applies to the perception of pictures. In the context of image-based rendering, a façade is first photographed from a certain position. This image is then projected onto simple 3D proxy geometry, which in turn is viewed from a novel camera position and then projected on to a 2D display device. The retinal hypothesis does not explain such "photograph of photograph" cases. However, it is feasible to derive the perceived angle using the same principles as the retinal hypothesis: projective geometry and vanishing points. We extend the retinal hypothesis to the context of image-based rendering such that it relates the perceived angle ζ_{ert}, simulation angle θ_s and eccentricity angle θ_e.

$$\zeta_{\text{ert}} = f\left(\theta_s, \theta_e\right) \tag{5.1}$$

The derivation of the function f is beyond the scope of the thesis, please refer to Vangorp *et al.* [2013] for details. Eccentricity angle θ_e with respect to a particular corner on the façade is the angle between the façade normal and the line joining the capture camera with the corner (see Figure 5.2). Simulation angle θ_s is defined as the angle between the façade normal and line joining the simulation camera and the corner. Clearly, simulation angle increases as the simulation camera is moved further away from the capture camera position. We assume that all photographs are captured by cameras that are frontoparallel to the façade, therefore, the façade normal is coincident with the capture camera orientation.

Goals Our first target is to study how well our extended retinal hypothesis can predict perceived angles in image-based rendering scenarios. Secondly, we want to quantify the tradeoff between the retinal and scene hypotheses and express the relationship as a function of capture and/or simulation parameters. Lastly, we want to develop an analytic model that predicts the level of perspective distortions given a set of capture and simulation parameters. To this end, we design and perform perceptual experiments to measure perceived angles and compare them to the extended retinal hypotheses. We then fit experimental data to an analytic function that predicts the level of perspective distortions and verify this model by means of a validation study. Finally, we demonstrate practical applications of our analysis, which together with the validation study, is the main contribution of this thesis in this study.

5.1.1 Experiment design

We conducted two psychophysical experiments to determine how the image distortions in typical street-level image-based rendering applications are perceived:

1. An angle-matching experiment that tested the predictions of the scene and extended retinal hypothesis by asking the participants to determine the perceived angle of corners. The results allow us to measure the relative influence of the two hypotheses on angle perception.
2. A rating experiment that determined the subjective acceptability of angle distortions. The results allow us to determine which perceived angle distortions are acceptable.

Stimuli We created synthetic 3D scenes and rendered them from known camera positions. We then created a single plane to approximate the 3D scene and textured it with the previously rendered images using projective texture mapping. This simulates the typical workflow where scenes are approximated by simple 3D geometry which is textured using photographs. Our stimuli included three façades, each with perpendicular balconies at three different protrusions, rendered from four different eccentricities with respect to the corner. The balconies or other corners of the 3D scene are absent on the 3D plane; we are interested in the perception of these angles. Synthetic stimuli give us full control over the scene; we validate our result our results on real datasets in Section 5.1.5.

Experimental procedure We use four different display sizes for our experiments: 55" television screen, 24" desktop monitor, 10" iPad and 3.5" iPhone. This is to test the effect of display size on angle perception. In both experiments, all stimuli – five simulation angles, four eccentricity angles, and three façades, each with three balcony depths, were presented twice. In addition, three particular stimuli were presented eight times to allow us to assess the consistency of responses. We repeated the process with different screens resulting in 384×4×2 stimuli, counting the four screens and two experiments. The order of stimuli presented was randomized. In the beginning, participants were given extensive in-

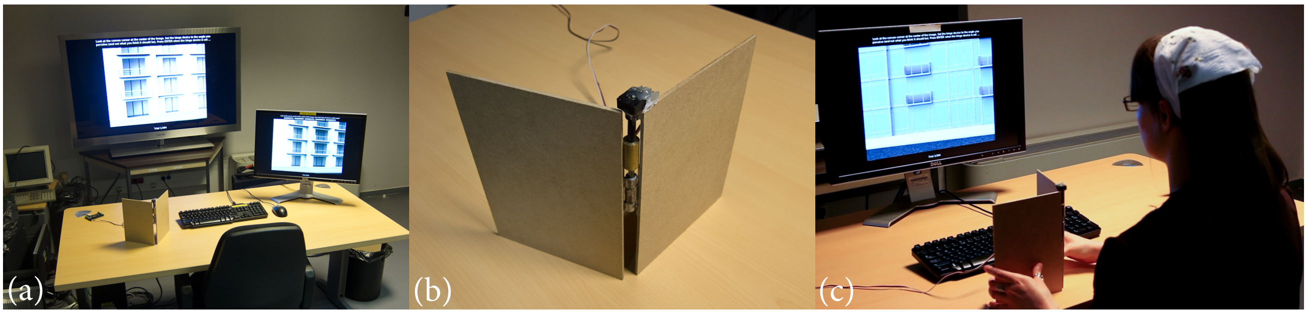
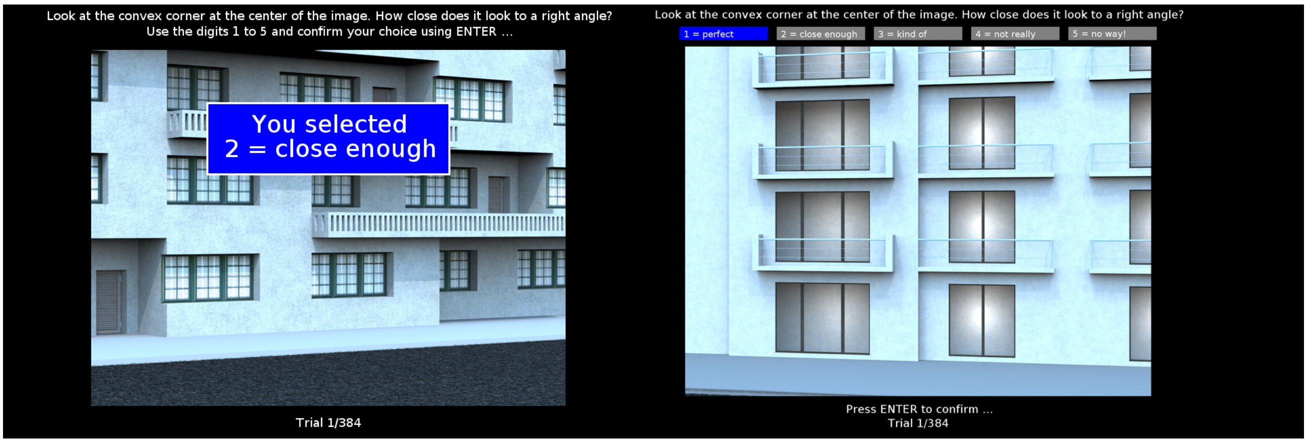


Figure 5.3: Experiment 1. (a) Experimental setup: we repeat the experiment on screens of different sizes - large size TV, computer monitor, iPad and iPhone, (b) Hinge device used by participants to specify perceived angle, and (c) a participant performing the experiment.



Look at the convex corner at the center of the image. How close does it look to a right angle?
Use the digits 1 to 5 and confirm your choice using ENTER ...

You selected
2 = close enough

Trial 1/384

Look at the convex corner at the center of the image. How close does it look to a right angle?

Press ENTER to confirm ...
Trial 1/384

Figure 5.4: Screenshots of experiment 2 on iPhone (left) and television (right).

structions and shown examples of stimuli with no or extreme angle distortions. It is common practice in vision science to use a relatively small number of participants who are tested extensively [Ernst and Banks, 2002]. We follow this practice by testing six paid participants extensively (7.5 hours on average for a total of 3072 measurements each).

5.1.2 Experiment 1: Hinge angle matching

In the first experiment, we determine how a 90° corner is perceived after it undergoes distortion due to the position of the simulation camera. Participants were asked to "Look at the convex corner at the center of the image. Set the hinge device to the angle you perceive (and not what you think it should be)" (see Figure 5.3(b) and video[1]). We used a real hinge instead of a virtual one, because the virtual hinge will itself undergo perceptual distortion when displayed to the participant.

Participants were shown images from our pool of stimuli in random order. The intended corner was always in the center of the image and briefly indicated by a blinking red dot. Participants adjusted the

[1] http://vimeo.com/64144141

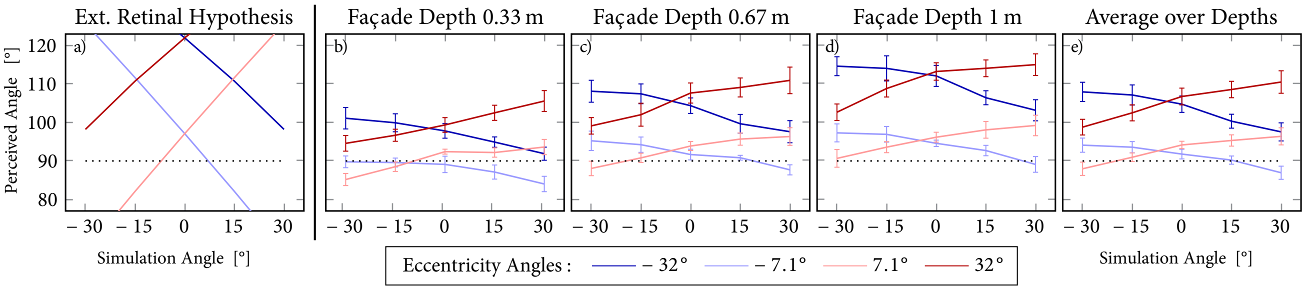


Figure 5.5: Perceived angle predictions by the extended retinal hypothesis (a), compared to the angle-matching results for different façade depths (b–d) and averaged over all depths (e). Error bars indicate the between-participant agreement. The dotted line at 90° represents the scene hypothesis.

hinge device until the hinge angle matched their perception of the corner angle. We recorded the hinge angle when the participant advanced to the next stimulus.

Results From a qualitative point of view, we observe that the trend of perceived angles is similar to the extended retinal hypothesis as shown in Figure 5.5; the variation can be attributed to the tradeoff between the retinal and scene hypotheses which in turn is determined by façade depth. This experimental data allows us to model the perceived angle as an analytic function that interpolates the two hypotheses in Section 5.1.4. Note that these results are averaged over all participants and display devices because we observed these to be statistically insignificant parameters.

5.1.3 Experiment 2: Angle rating

In the second experiment, we study the acceptability of various angle distortions. We asked participants to indicate how acceptable a given corner was as a simulation of a 90° corner. Participants were shown the same images as in the previous experiment in random order. They rated how close the indicated corner in each image looked to a right angle on a 5-point scale where 1 to 5 corresponded to "perfect", "close enough", "kind of", "not really", and "no way!". Participants entered each rating using a numerical keypad and confirmed the entry by pressing "Enter" (see Figure 5.4 and video[2]).

Results We use interpolated medians [Revelle, 2008] to summarize the ratings data accumulated from different participants over different façades and display devices (see Figure 5.6). Clearly, the most unacceptable stimuli are in the lower left and upper right corners of these plots, which correspond to large simulation and eccentricity angles of the same sign. The most acceptable stimuli are in the middle of the plot where the simulation and eccentricity angles are small in magnitude – and the upper left and

[2] http://vimeo.com/64144141

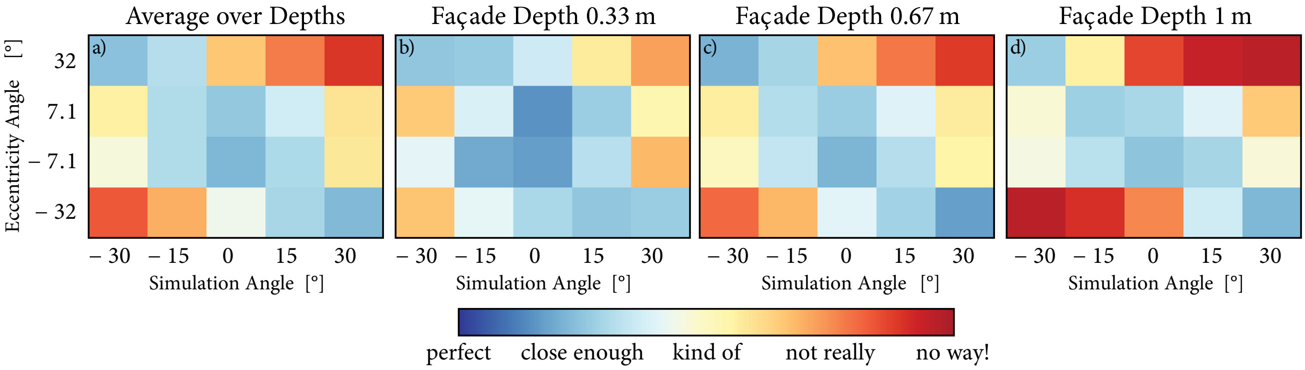


Figure 5.6: Rating results showing interpolated medians across all participants: (a) across all façade depths, and (b–d) ratings for different façade depths.

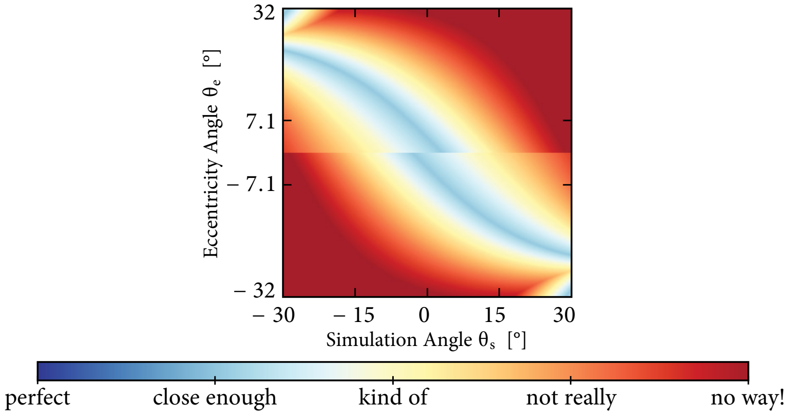


Figure 5.7: Predictive model for perspective distortions for façade depth of 10 meters. We fit the experimental data to develop the predictive model which quantifies the level of perspective distortions for different eccentricity and simulation angles.

lower right where the large simulation and eccentricity angles are opposite in sign. A comparison with Figure 5.5 shows that the most unacceptable cases are those when perceived angle is very different from 90°. We formalize the relationship between the two experiments by developing a predictive model for the level of perspective distortions in the next section.

5.1.4 Inferences: Predictive model for perspective distortions

Our extended retinal hypothesis (see Equation 5.1) gives the perceived angle ζ_{ert} as a function of eccentricity and simulation angles, whereas the scene hypothesis gives the perceived angles as the true angle i.e., 90°, in all cases. In practice, the actual perceived angle is always between the two hypotheses (see Figure 5.5). We fit the experimental data to an analytic function that interpolates the two hypotheses and gives the perceived angle as follows:

$$\zeta_{\text{perceived}} = \zeta_{\text{ert}} \cdot g(d) + 90^\circ \cdot \big(1 - g(d)\big) \tag{5.2}$$

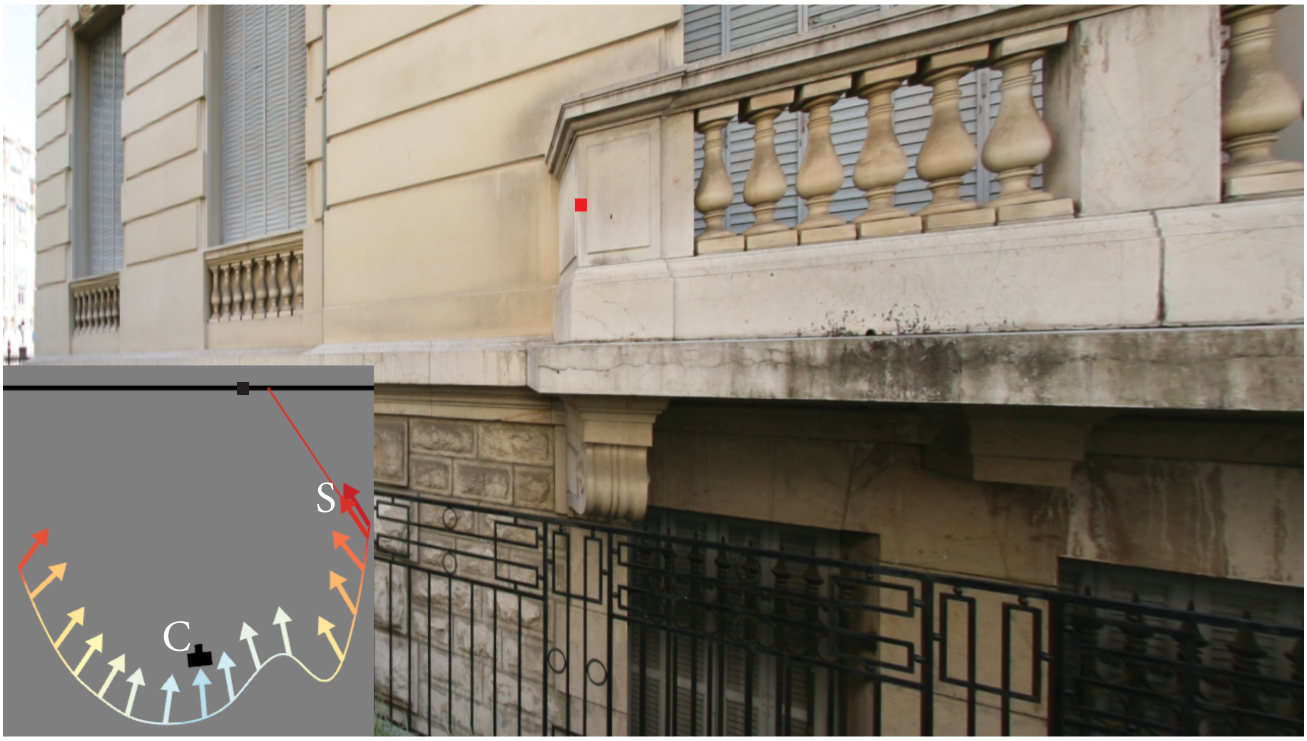


Figure 5.8: Simulated views along a navigation path (see inset). The heat map of predicted ratings of Figure 5.7 is used as the color scheme to visualize predicted quality along the path. The view shown here is predicted to be heavily distorted, as indicated by the red line-of-sight in the inset (labeled "S" in inset). The capture camera is indicated by "C", façade by the black line and the balcony corner by the black dot on the black line in inset.

where, d is the depth of the façade and g is the result of data fitting. We then fit the rating experiment data to develop an analytic relationship between the perceived angle and rating value as follows:

$$\text{Rating} \xleftarrow{h} \zeta_{\text{perceived}} \tag{5.3}$$

The details of the interpolating function g in Equation 5.2 and the mapping function h in Equation 5.3 are beyond the scope of this thesis, please see Vangorp *et al.* [2013] for more details. The combination of Equations 5.1, 5.2 and 5.3 gives the final predictive model which maps the capture simulation parameters, namely façade depth, eccentricity angle and simulation angle to a rating level of perspective distortions as shown in Figure 5.7.

5.1.5 Validation of experimental results

We present a prototype interface for street-level image-based rendering which we use to visualize the quality of a simulated view path, and perform a study to validate our predictive model.

Visualization of predicted ratings Our implementation reads a set of cameras calibrated using structure from motion [Snavely *et al.*, 2006]. We assume that the cameras are fronto-parallel to the façade which corresponds to a side camera of a commercial capture car for Google Streetview-like applications. We fit a single 3D plane to the reconstructed point cloud of the scene generated using multi-view stereo [Furukawa and Ponce, 2009]. The plane serves as the proxy for the façade, much like street-level image-based rendering applications. We finally develop a simple projective texturing application which projects a particular input image onto the proxy plane and visualizes it from different simulated camera positions (see Figure 5.8). The rendered view is synthesized from a single image, hence it can only have perspective distortions. Rendering artifacts like popping and ghosting are not present since we use the same input image for generating all simulated views.

We use the above application as a design tool where we can design simulated camera paths. The top view of the 3D scene is shown in the inset in Figure 5.8. The top view shows the façade and the corner as the black line and the black dot. The abscissa is the horizontal position of the simulation camera relative to the capture camera and the ordinate is the distance of the simulation camera to the façade. We use the same visualization in Figures 5.10, 5.11(a) and 5.11(b). We can mark several keypoints with orientations which serve as simulated camera positions (shown as arrows in inset in Figure 5.8). We then fit a cubic spline to these control points which gives a full simulated camera path with positions and orientations. For any point on this path, we can use the simulated camera position and orientation, input camera camera position and orientation to predict the level of perspective distortion as per the predictive model in the previous section. The predicted distortion level is indicated using the heat map of Figure 5.7.

While using our interface we noticed one significant temporal effect: motions along a path on which the perceived angle changes quickly are quite disconcerting. We can predict such paths as those which cross many different rating levels and design paths with the desired temporal variation.

Validation user study To evaluate our predictive model, we use the above street-level image-based rendering prototype in a user user study. The goal of the study is to determine how well our predictions agree with user observations in a situation quite similar to street-level image-based rendering: a navigation path with real stimuli. For each of the three datasets used for this study, we provide three different paths: one path was predicted to be low quality (rating of 3-3.5), one medium quality (2.5-3), and one high quality (2-2.5). We designed these paths with the above visualization tool. We created pre-recorded sequences of these paths and presented them to participants on a webpage. Participants were instructed to look at a specific corner when it appeared in the middle of the screen (indicated by a red dot). They rated the perceived distortion in the same manner as for Experiment 2: i.e., “Look at the corner indicated by the red square. How close does it look to a right angle when the red square is

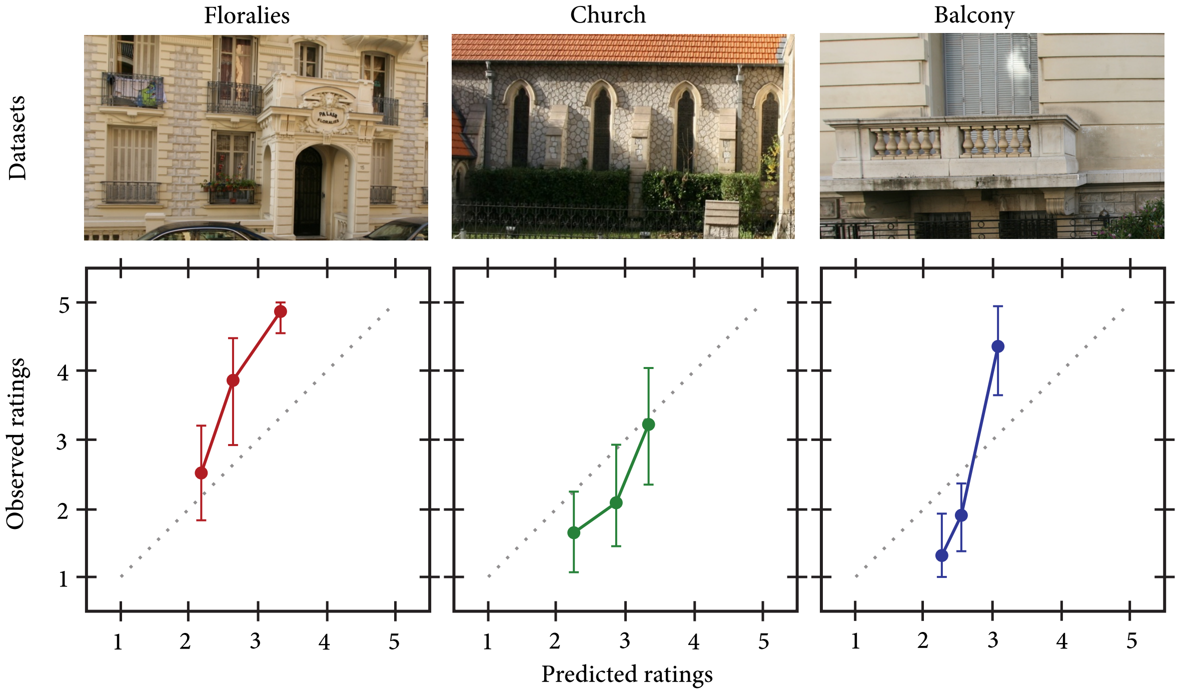


Figure 5.9: Observed ratings (colored lines) for the three scenes compared to predicted ratings (dotted lines) given by our predictive model for perspective distortions (see Section 5.1.4).

visible?" (see video[3]). They chose a value on the same five-point scale (see Section 5.1.3) for a total of nine paths (three paths for three scenes). We presented the three videos of each dataset on a single page, and instructed participants to adjust their relative ratings between the three videos. A total of 91 participants performed the study on their own computer screens. We summarize the results in Figure 5.9, which plots observed ratings as a function of predicted ratings separately for the three scenes. The correlation between predicted and observed ratings is moderate ($r > 0.5$) for the first two scenes and strong ($r > 0.8$) for the third. Thus, the predictions are reasonably good despite the many differences between the experiments used to generate the predictions (static scenes with well controlled conditions) and this user study (unstructured dynamic scenes).

5.1.6 Applications to street-level image-based rendering

Restricting free viewpoint navigation We use the above rendering application and perspective distortion visualization to develop an interactive application that exploits the predictive model for perspective distortions (see Figure 5.10). The interface shows the simulated view and a top view of the scenario in the inset. The user starts viewing at a particular position, and then translates and/or rotates. If the user is translating (Figure 5.10(a)), the inset shows predicted ratings for all camera positions while keeping the same camera orientation. Figure 5.11(a) shows a similar visualization for three particular simulation

[3] http://vimeo.com/64144141

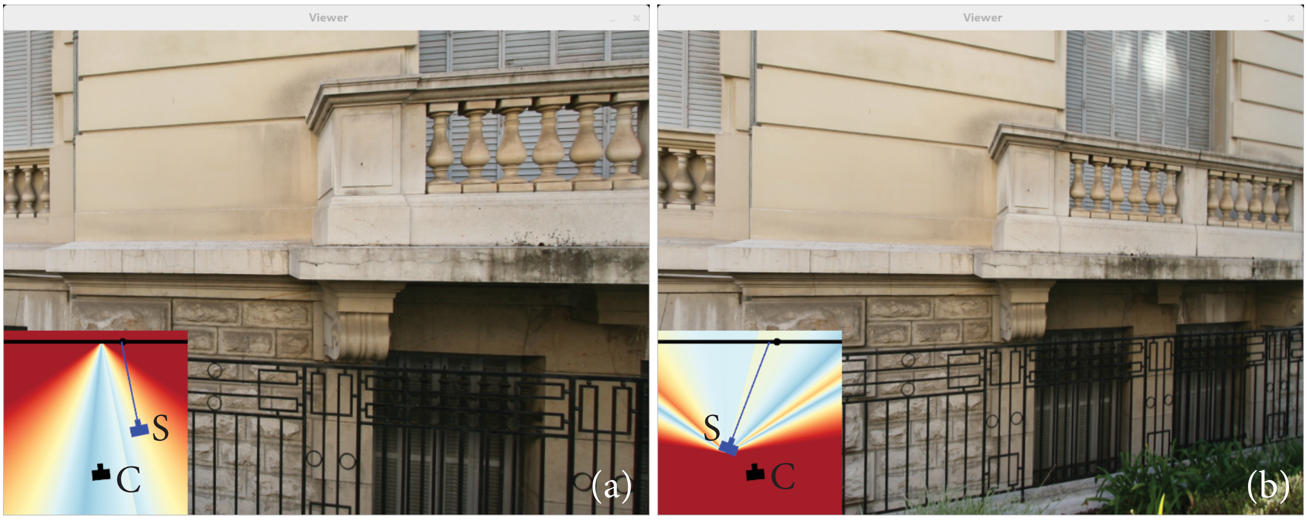


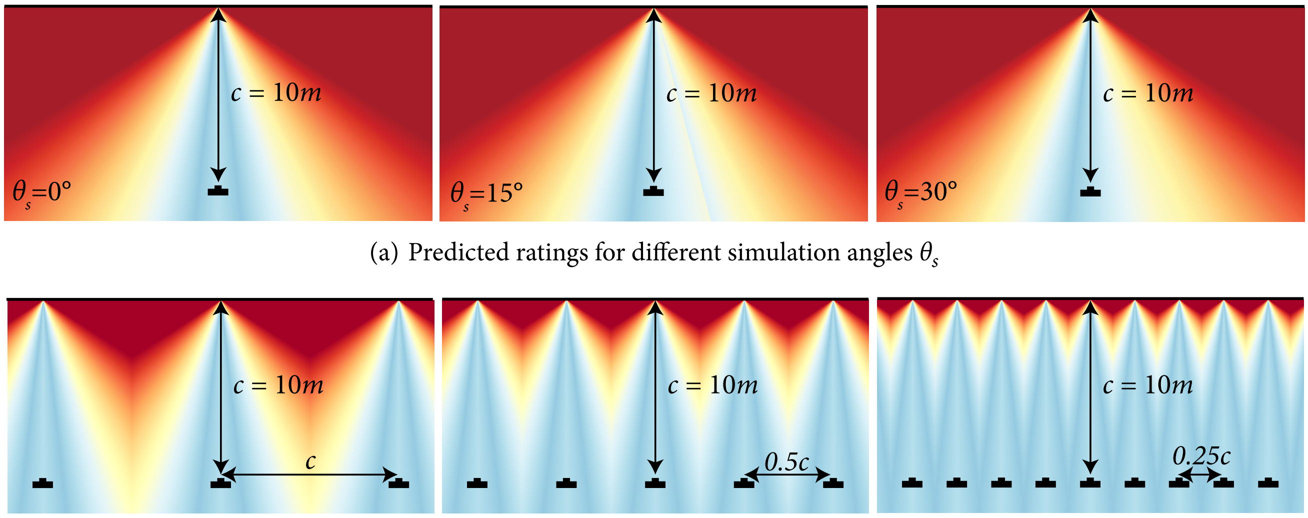
Figure 5.10: Interactive navigation tool. (a) When the user translates, the inset shows predicted ratings for all camera positions keeping the orientation fixed. (b) When the user turns, the inset shows ratings for all possible camera orientations keeping the position fixed. The application restricts the user navigation to regions with acceptable predicted quality. Capture camera is indicated by "C", simulation camera by "S", façade by a black line and corner by a black dot on the black line in the insets.

angles. When the user turns (Figure 5.10(b)), the visualization shows ratings for all camera orientations keeping the camera position fixed. The user can translate and turn as they wish as long as they stay within the zone of acceptable quality; we use a rating value of 3 as the threshold. The application prevents the user from reaching a camera position or orientation that corresponds to a predicted rating higher than a threshold, and instead shows a blinking camera icon at the current camera position (see video[4]).

Capture guidelines We also use our predictive model to provide capture density guidelines. Figure 5.11(a) shows that each capture camera induces a region of acceptable perspective distortions for any simulation angle, represented by blue regions of the heat map from Figure 5.7. Any novel view can be synthesized using the input camera that gives the best possible rating for the particular simulation position. In other words, additional capture cameras induce identical regions of acceptable perspective distortions; the rating for any novel viewpoint can be calculated as the best of the rating values induced by different capture cameras as shown in Figure 5.11(b). We empirically observe that the ratings are acceptable everywhere if the displacement between capture cameras is at most one-fourth the distance of capture cameras from the façade. The level of perspective distortions is expected to be perceptually acceptable for such a capture. For baselines greater than this threshold, some yellow-orange-red regions are observed (see Figure 5.11(b)), indicating potentially severe perspective distortions.

The results shown in Chapters 3 and 4 do not show significant perspective distortions even though the shape-preserving warp used to synthesize novel views is an approximation to the true affine trans-

[4] http://vimeo.com/64144141



(a) Predicted ratings for different simulation angles θ_s

(b) Predicted ratings for different capture densities with $\theta_s = 0°$

Figure 5.11: Capture density guidelines. Increasing the number of capture cameras induces larger areas of acceptable distortion (shown as blue). We observe empirically that a baseline of one-fourth the distance of the capture camera from the façade results in largely acceptable ratings everywhere. The capture camera position is shown in black and the façade is indicated by the black line on top.

formation. This is because the baseline between input cameras is always 2–5 meters while the distance of capture cameras from scene geometry is typically 15–20 meters.

Given the navigation requirements of an image-based rendering system, the above analysis allows us to compute the required capture density as well as capture positions. It is important for many applications especially those operating at city scale, to plan the capture beforehand, since brute force captures in the form of videos are prohibitively expensive.

5.2 Perception of ghosting artifacts

The previous study investigates perspective distortions incurred when a single input image is reprojected into a novel view. This is only the first step of most practical image-based rendering systems which reproject multiple input images in the novel view and assemble visual content from each of these reprojected images. This is done typically to create smooth transitions and approximate parallax effects as the novel view transitions in the 3D scene. Blending is the most popular methodology for the above because of its ease of implementation, real time performance and acceptable visual quality. The earliest image-based rendering frameworks advocate heavy blending between images [Buehler *et al.*, 2001] which gives smooth temporal transitions. However, synthesized novel views are often blurred, lacking in crisp details and high frequencies present in the input images. Ghosting artifacts ensue when the underlying 3D geometry is lacking in details.

On the other hand, reducing the degree of blending between multiple images leads to pronounced transitions between images as the novel camera is translated or rotated. We refer to these disturbing temporal discontinuities as popping artifacts. Most image-based rendering approaches resolve this tradeoff by selecting blending weights that give the best results for the datasets being tested without any intuition about the generalizability of the blending scheme. Other approaches bypass this issue by avoiding blending altogether; they composite pixels from multiple images using graph cut [Mahajan *et al.*, 2009] to synthesize the novel view. The results are impressive but these approaches are far from real time and are unsuitable for our goal of image-based rendering as an interactive visualization tool. Since blending cannot be avoided, it is important to develop a principled study of blending artifacts.

In this section, we develop psychophysical experiments to study the tradeoff between ghosting and popping artifacts. We create a number of image-based rendered stimuli using different capture and rendering parameters and show these to participants as compared to ground truth i.e., video of the same scene. Such comparisons with ground truth have been shown before for still images [Fitzgibbon *et al.*, 2005] but our work is the first to approach the problem from a perceptual point of view; previous work has focused on comparing image statistics which can be incomplete because the relationship between image statistics and perceived rendering quality is not necessarily well defined.

In the discussion of the results of our study, we provide guidelines to facilitate optimal capture, as well as motivate algorithmic choices used in image-based rendering systems.

5.2.1 Experiment overview

We study image-based rendering in two of its most common forms: unstructured lumigraph [Buehler *et al.*, 2001] and cross-fading [Sinha *et al.*, 2009].

Unstructured lumigraph Many current image-based rendering systems use 3D geometry of the scene and texture it using pixels from multiple input images blended with appropriate per-pixel weights. If the 3D geometry is not perfect, ghosting artifacts are observed if multiple images are blended. In contrast, abrupt temporal popping artifacts are observed if a single image is used to synthesize any output pixel.

The two parameters that control the severity of artifacts for a given level of geometric reconstruction, are the *coverage* between input images, and the *number of images blended*, at any given pixel. Coverage is a way to measure capture image density, and thus defines the total number of images used to generate the final result. We define coverage in a canonical fronto-parallel viewing condition, as the number of images covering a given point on the planar proxy on average. Low coverage causes *slow popping* with infrequent but long jumps; a dense set of input images causes *fast popping* with frequent but short jumps.

Blending more images per-pixel (e.g. 2, 3 or more) increases ghosting artifacts; low coverage causes well-separated ghosts while high coverage leads to smaller displacement between the ghosts.

Cross-fading Another approach for synthesizing novel views in the context of view interpolation is cross-fading [Sinha *et al.*, 2009; Kemelmacher-Shlizerman *et al.*, 2011]. This appears to be the base for commercial image-based rendering systems like Google Streetview. Here the same two input images are blended at each pixel and the blending weight of each image is same for all the pixels of the novel view. Unstructured lumigraph is a generalization of cross-fading because it can select a different set of input images, each with its own blending weight, to synthesize each pixel of the novel view independently. It is therefore capable of free viewpoint image-based rendering while cross-fading is restricted to view interpolation. However, cross-fading may be advantageous in certain cases where it presents a different tradeoff between ghosting and popping artifacts. While transitioning from image A to B, we can use only A for a fraction of the path, then cross-fade between A and B for another fraction and then use B for the rest of the path. The duration of the cross-fade determines the severity of the two artifacts. Cross-fading for the entire duration of the path results in maximum ghosting with very smooth temporal effects while cross-fading for a very small duration (e.g. $< 5\%$) results in a visible abrupt temporal transition.

We conducted two psychophysical experiments to formally study the tradeoff between ghosting and popping for the above two image-based rendering setups:

1. The first experiment compared ghosting and popping artifacts in unstructured lumigraph using simple planar geometry as a function of the coverage and number of images blended.
2. The second experiment studied the same tradeoff in cross-fading, and further compared this to unstructured lumigraph with different parameters.

Detailed discussion of the statistical analysis of experimental results is beyond the scope of this thesis, please refer to [Vangorp *et al.*, 2011] for more details.

5.2.2 Experiment 1: Artifact analysis in Unstructured Lumigraph

The purpose of this experiment is to measure how ghosting and popping artifacts affect the perceived quality of unstructured lumigraph of real façades. From the discussion in previous section, it is clear that ghosting and popping artifacts are the result of different levels of blending between input images. The specific questions we seek to answer are as follows:

- Under which conditions do the artifacts become objectionable?
- Which type of artifact is worse?
- What is the optimal display strategy when there are restrictions on the number of images that can be captured or stored?

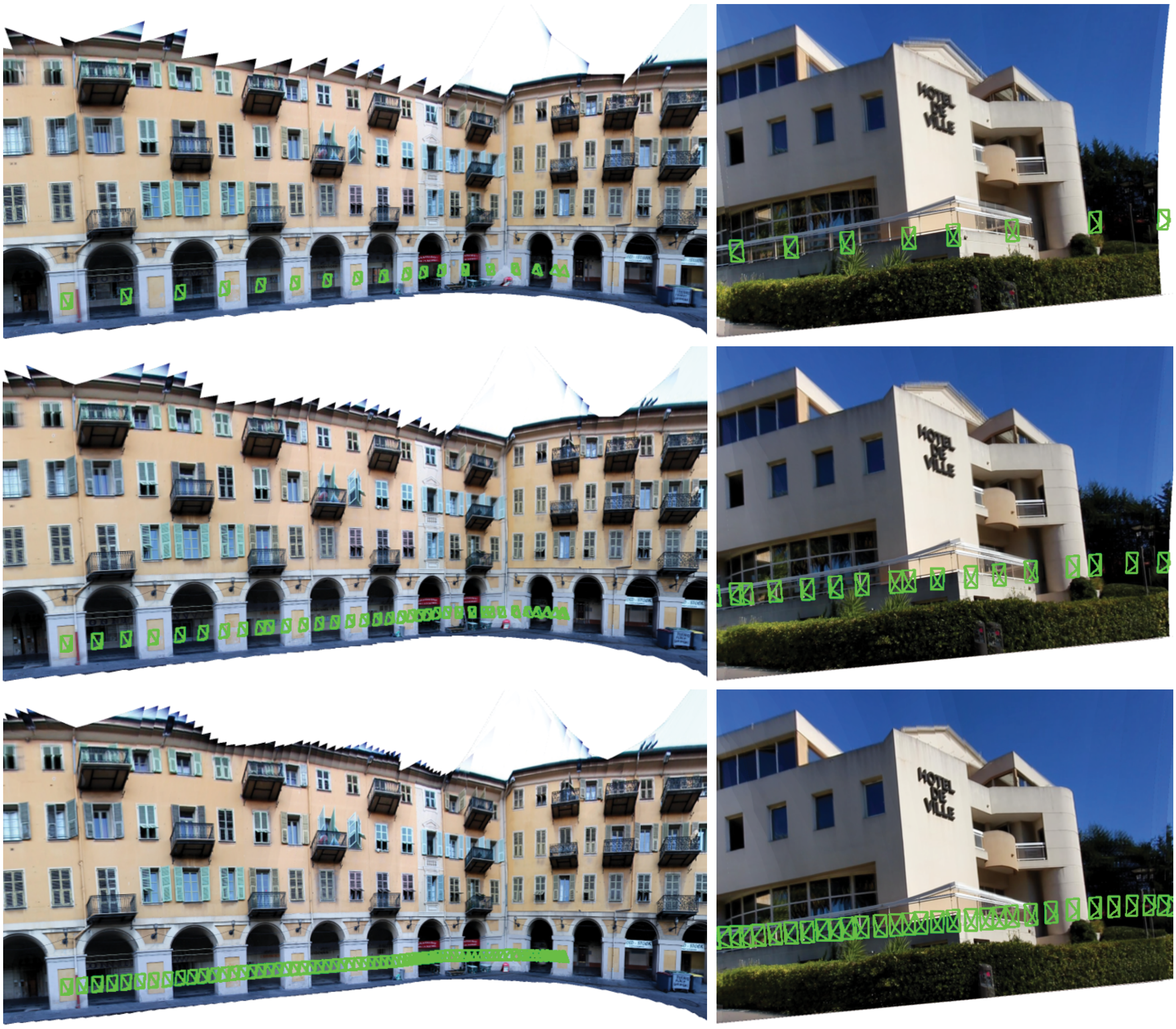
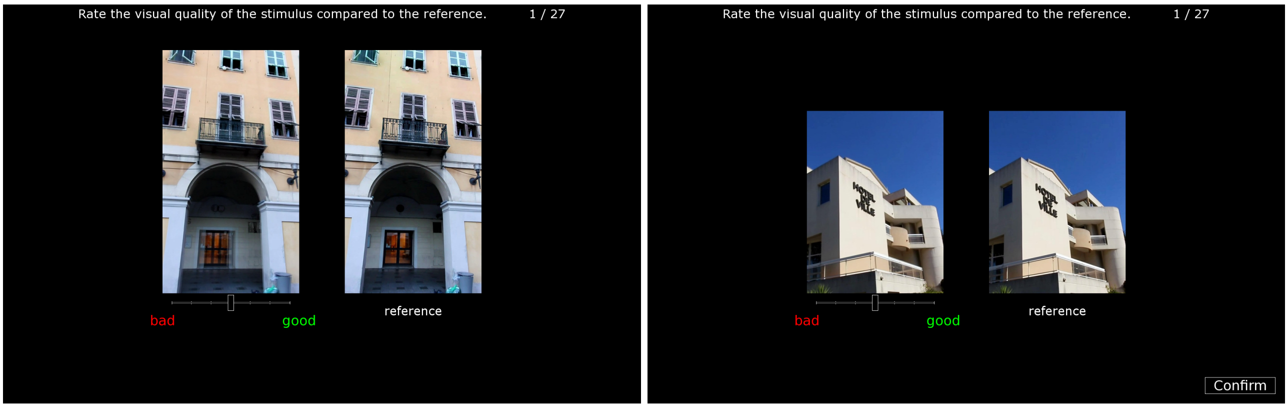


Figure 5.12: Corner scene (left) and Town Hall scene (right) used for experiment 1. The capture camera viewpoints (shown in green) in the three rows demonstrate the three capture densities or coverage used in the experiment.



Rate the visual quality of the stimulus compared to the reference.

Rate the visual quality of the stimulus compared to the reference.

Figure 5.13: Experimental interface for the visual quality rating experiment for Corner scene (left) and Town Hall scene (right).

Stimuli We study the effect of coverage and number of images blended on artifacts; in order to eliminate scene complexity and quality of 3D geometry as variables, we selected scenes with two depth levels – façade with convex (balconies) or concave (arches) features and used a few planes and boxes as 3D geometry. We captured steady video sequences of a Corner of a large city square and of a Town Hall (see Figure 5.12), which allows us to make direct comparisons between image-based renderings and real video. We then extract a regular subsampling of frames from the video and use structure from motion [Snavely *et al.*, 2006] to calibrate the cameras and generate a sparse 3D point set, which we use to create a piecewise planar proxy geometry. We then used unstructured lumigraph [Buehler *et al.*, 2001] as the image-based rendering algorithm to generate the stimuli. The video serves as the ground truth for a view interpolation generated unstructured lumigraph.

Experimental procedure The parameters we vary for the approximate renderings are (1) coverage, and (2) number of images blended for any given pixel. For coverage, we use low (`lo`), medium (`me`) and high (`hi`) values corresponding to approximately 3, 6 and 12 images covering any point of the scene geometry. We need 18, 36 and 65 (Town Hall) or 69 (Corner) input images to achieve these values of coverage (see Figure 5.12). For the number of images blended per pixel, we use values of 1, 2 and 3, as commonly used for this class of image-based rendering techniques [Buehler *et al.*, 2001; Eisemann *et al.*, 2008].

The participants were presented with a pair of videos: an image-based rendering and the corresponding video reference. The videos play in a loop of approximately 16 seconds with the camera moving to and fro along the path. The participants were asked to "rate the visual quality of the approximation with respect to the reference" using a continuous slider (see Figure 5.13). This provides a direct measure of quality. Each of the 3×3 stimuli is repeated 3 times in random order, in separate blocks for both scenes

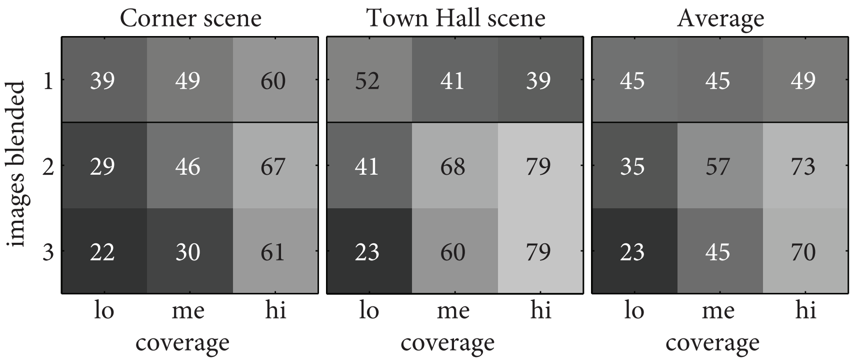


Figure 5.14: Average visual quality ratings for Experiment 1, ranging from the worst quality (0%, black) to the best (100%, white). Higher values means the sequence looked better, i.e., fewer artifacts. Top row corresponds to popping artifacts while the other rows correspond to ghosting artifacts.

(see video[5]).

Results In what follows, we report visual quality levels as percentages depending upon the position of the slider as marked by the participants. We report differences in visual quality levels as percentage points (*pp*). Intuitively, we would expect a monotonic progression of quality as we increase the number of images used overall. The key question is how this is affected by popping and blending artifacts.

Popping artifacts The top rows of Figure 5.14 refer to popping, since only a single image is being used at any given pixel. For this case, the overall visual quality appears to depend on the severity of the artifacts which varies from scene to scene. This dependence on the scene is revealed by linear regression of the quality as a function of coverage. There is a significant preference for faster popping in the Corner scene (significantly positive slope of 10.38*pp* per approximate doubling of coverage). A more surprising outcome is the preference for *slower* popping in the Town Hall scene (significantly negative slope of -6.53*pp* per approximate doubling of coverage). This result is of interest since it means that it is not necessarily advantageous to use a larger number of images.

Ghosting artifacts In contrast, for ghosting artifacts (see Figure 5.14, bottom rows), linear regression confirms our expectation that the overall visual quality improves as the coverage grows (significantly positive slope of 21.45*pp* per approximate doubling of coverage). With a sparser set of input images, the images blended are further from the output camera position on average and therefore result in larger feature misalignment when projected onto the planar geometric proxy.

We might expect that blending more images together at every pixel improves appearance by smoothing out transitions. Interestingly, however, we find that perceived visual quality tends to improve when fewer images are blended per pixel. The average perceived quality increases by 9.14*pp* from 3 to 2 images

[5] http://www.youtube.com/watch?v=akaWUeOmum8

blended per pixel. When geometry is not sufficiently accurate, blending fewer images at any given pixel reduces blurring or the number and spatial extent of ghost images.

Comparison between artifacts It is interesting to study whether there is a clear difference in quality between popping (using 1 image per pixel) or blending 2 images per pixel. We find that the relative unpleasantness of popping and blending artifacts depends on the preference for fast or slow popping in the scene. However, in both scenes there is a crossover point; popping is preferred for low coverage, and blending 2 images which results in limited ghosting, is preferred for high coverage. Blending 3 images results in more pronounced ghosting which consistently ranks less preferable to the other two options.

5.2.3 Experiment 2: Artifact analysis in Cross Fading

The goal of this experiment is to address the following questions:

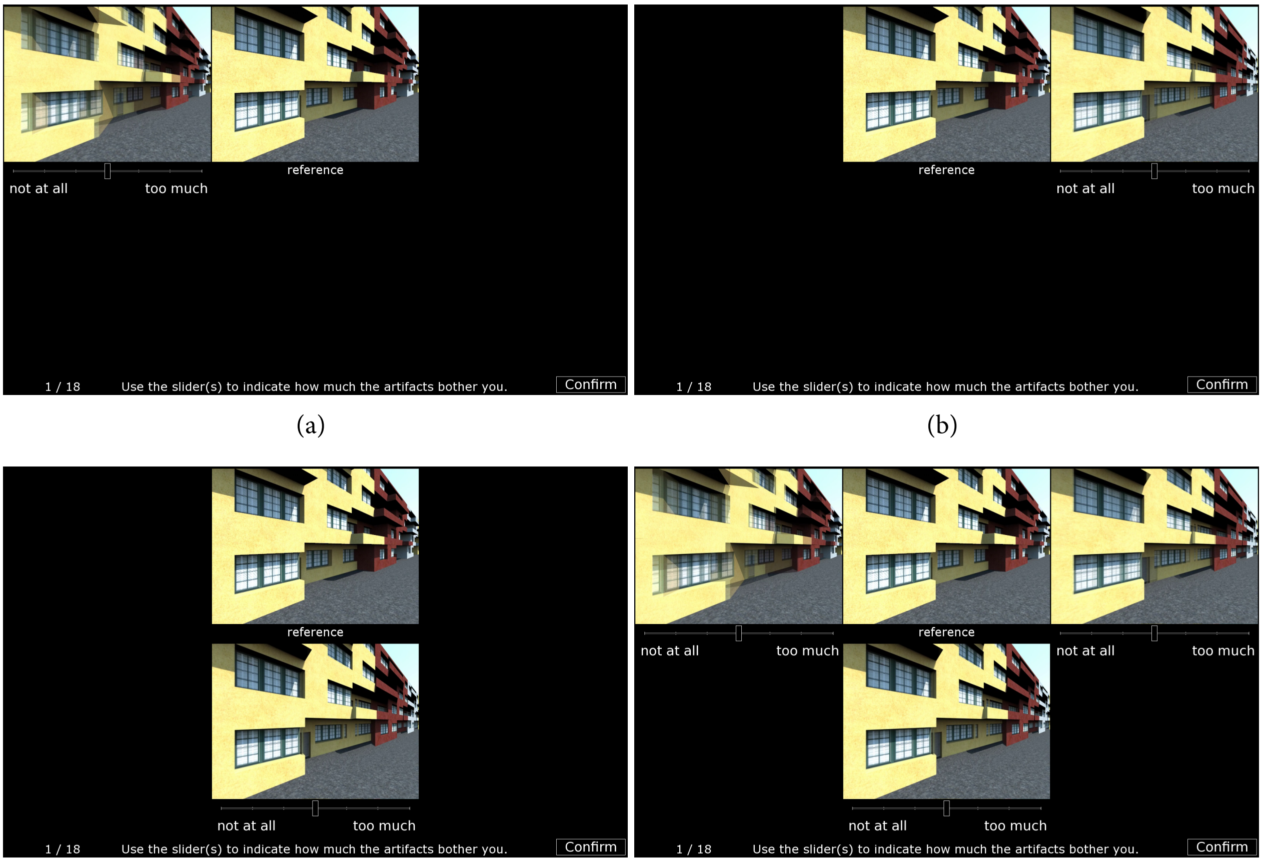
- How does cross-fading compare to unstructured lumigraph [Buehler *et al.*, 2001] in terms of artifacts?
- Should transitions be fast (potentially too abrupt), or slow (potentially causing misalignment artifacts to be visible for longer duration)?

Stimuli We performed this experiment first with artificial scenes which allow precise control over experimental conditions and then investigated how the results generalize to a real scenes even though the control over experimental conditions is necessarily less precise. We performed the study with artificial stimuli because cross-fading with wide-angle imagery leads to perspective distortions (see Figure 5.1(right)) which intuitively seem to depend upon scene and capture characteristics such as depth range of façade and novel viewpoint's position and orientation (see Section 5.1). Artificial stimuli allow us to fix these variables.

We created an artificial façade used with a fixed depth range and viewing angle 45° (see Figure 5.15 and video[6]). We then rendered the façade from two end points with a wide field of view; these serve as input images for cross-fading. We generated the stimuli by projecting these images onto the planar proxy and blending them using linear interpolation weights over the full output camera path or over the middle 40% or 10%. Before and after this blending transition only a single image was used to synthesize the result. We created the reference video by rendering the entire path using physically-based rendering. While the internal details of commercial systems like Google Streetview are unknown, our stimuli resemble their renderings.

We also generated stimuli for unstructured lumigraph rendering to compare to cross-fading. To this end, we render the façade from evenly spaced cameras at a density equivalent to the densest set of the

[6] http://www.youtube.com/watch?v=akaWUeOmum8



Use the slider(s) to indicate how much the artifacts bother you.

(a)

Use the slider(s) to indicate how much the artifacts bother you.

(b)

Use the slider(s) to indicate how much the artifacts bother you.

(c)

Use the slider(s) to indicate how much the artifacts bother you.

(d)

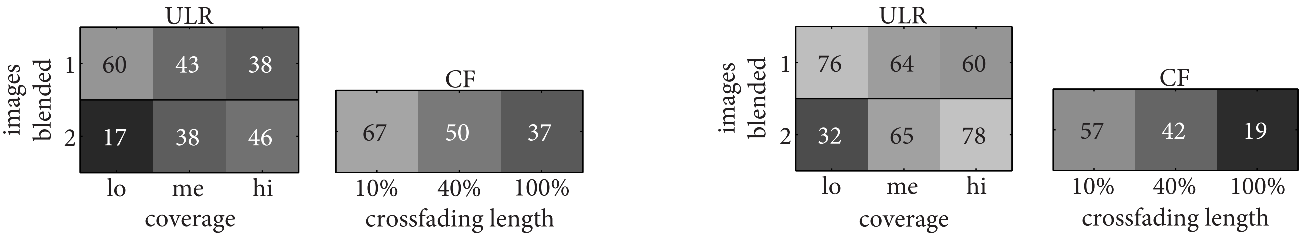
Figure 5.15: Interface for the cross-fading experiment. The participant is presented first with (a–c) one of three image-based rendering result and the reference, and then with (d) all three renderings and the reference.

Corner scene of Experiment 1. We only generated the stimuli for blending 1 or 2 images with varying coverage; we abandoned blending 3 images at each pixel because it clearly does not improve the visual quality as shown by results of previous experiment.

We generated the stimuli for real scenes using the same procedure, with real photographs replacing snapshots of the synthetic scene. The ground truth was generated by rendering a sequence of images for the synthetic case. For the real scene, we did not record a ground truth.

Experimental procedure For the cross-fading stimuli, we tested three different durations of cross-fading. We created the stimuli with 100%, 40% and 10% of the path under cross-fading for both artificial and real stimuli. For the unstructured lumigraph stimuli, we varied the coverage and number of images blended per-pixel the same way as in Experiment 1, the only difference being we did not test the case for blending 3 images per-pixel because the results of the previous experiment indicate that increasing the number of images from 2 to 3 did not improve visual quality in any case (see Figure 5.14).

In case of the synthetic scenes, participants were presented with the stimuli and asked to "rate how



(a) Artificial stimuli

(b) Real stimuli

Figure 5.16: Average visual quality ratings for Experiment 2 using artificial and real stimuli, ranging from the worst quality (0%, black) to the best (100%, white). The left figure labeled "ULR" shows the results of unstructured lumigraph and the right figure labeled "CF" shows cross-fading results. The top row of unstructured lumigraph result corresponds to popping artifacts (using 1 image per pixel), and the bottom row corresponds to ghosting artifacts (blending 2 images per pixel).

much the artifacts bothered them" by adjusting a continuous slider as in the previous experiment. Participants were first presented with the identified reference stimulus in the center of the screen, with one additional stimulus corresponding to one of blending, popping or cross-fading. These were presented in randomized order, to the left, right and below the reference (see Figure 5.15). Blending and popping in a given trial use the same total number of images. The participant rated each stimulus with respect to the reference. After the three stimuli have been rated, the participant was presented with all three stimuli, sliders and the reference, and may adjust the relative ratings (see video[7]). We repeated the same experiment for the real scenes but without the reference stimulus which amounts to no-reference comparative study between cross-fading and blending.

Results Figure 5.16(a) summarizes the visual quality for the cross-fading experiment with artificial stimuli, averaged over 10 participants. Short cross-fading is given the highest quality rating overall, while longer cross-fading received very low ratings, demonstrating a preference for shorter cross-fading (significantly negative slope of -3.20*pp* per 10% increase in cross-fading length). Short cross-fading results in stronger parallax artifacts towards the middle of the path, but less prolonged ghosting artifacts during the transition. This suggests that the parallax distortions are less objectionable than the blending artifacts in these stimuli.

Figure 5.16(b) summarizes the visual quality for the cross-fading experiment with the real scene, averaged over 8 participants. This confirms the trends within each technique. There is again a clear preference for shorter cross-fading (significantly negative slope of -4.13*pp* per 10% increase in cross-fading length). There is a slight preference for slow popping in the case of unstructured lumigraph (significantly negative slope of -8.36*pp* per approximate doubling of coverage) and for blending with denser coverage.

[7] http://www.youtube.com/watch?v=akaWUeOmum8

Validation of unstructured lumigraph results The design of the experiment allows us to revisit the question of whether popping or blending artifacts are preferable in unstructured lumigraph. In contrast to the Corner scene in previous experiment, slow popping is preferred (significantly negative slope of -11.23*pp* per doubling of coverage). These results also confirm the results from the previous experiment that perceived quality of ghosting artifacts in unstructured lumigraph improves with higher coverage (see Figure 5.14).

The results for artificial scenes show a higher relative quality of cross-fading compared to unstructured lumigraph. We believe this is most likely caused by the lack of detail and complexity in the artificial façade and by the high accuracy of its geometric proxy and camera positions. In real scenes, misalignment between the two input images at the far ends of a façade as used by cross-fading are typically larger and more noticeable than between consecutive input images as used for unstructured lumigraph. Therefore we hypothesize that the results with real stimuli are more suitable as a basis for guidelines which will generalize to other real world scenes.

5.2.4 Guidelines for current image-based rendering systems

We now summarize the results of the experiments and present guidelines for capture and display.

Unstructured lumigraph guidelines: ghosting versus popping The results from the unstructured lumigraph experiment (see Figure 5.14) show a systematic ranking of popping and ghosting. When coverage is low, popping is clearly preferable to ghosting artifacts. This seems to be the case because synthesized views are as crisp as the input images and the artifacts are visible only at transitions, presenting a plausible image for longer duration. Dense captures lead to more frequent popping which is perceived as worse than slow popping with sparser captures. Ghosting causes salient image content such as text, architectural features etc. to become illegible or unrecognizable, which is perceived as more disturbing compared to sudden transitions. The best overall result is achieved when coverage is high and 2 images are blended per pixel. This gives the best tradeoff; however it might be impractical because dense captures can result in prohibitive acquisition and storage costs.

Cross fading guidelines Our experiment indicates an interesting way to improve image-based navigation applications based on cross-fading, such as Google Streetview which currently appear to use a technique akin to long cross-fading. Our results show that by switching to shorter cross-fading perceived quality would be enhanced, despite perspective distortions.

Cross fading versus unstructured lumigraph The comparison between cross-fading and unstructured lumigraph on real scenes (see Figure 5.16(b)) show that the latter gives better quality by (a) using single image to synthesize each pixel under low coverage i.e. slow popping, or (b) blending 2 images per-pixel under dense coverage.

Ghosting artifacts versus perspective distortions The data collected from cross-fading experiment with both artificial and real stimuli suggest that cross-fading of short transition is preferred. In the short cross-fade condition, perspective distortions become acute towards the middle of the path; despite this, the condition is ranked as highest quality among cross-fading stimuli. This indirectly indicates that perspective distortions are more tolerable than ghosting artifacts caused by long transition cross-fades. The shape-preserving warp in Chapters 3 and 4 exploits this observation. The warp produces perspective errors, however, the overall perceived quality is high because both the approaches minimize the perceptually more important ghosting artifacts.

Conclusion It is clear that heavy ghosting artifacts are always perceived as unacceptable even if they lead to smooth temporal transitions. Observers seem to prefer crisp images even if there are parallax errors or temporal jumps. This fundamental guideline is the basis of our blending strategy in both the image-based rendering approaches of Chapters 3 and 4. Both the blending strategies use at most 2 images at each pixel and even then favor one candidate heavily, resulting in minimal ghosting.

The shape-preserving warp used in both the image-based rendering approaches is again motivated by the fact that perspective distortions are less noticeable than ghosting artifacts. The shape-preserving warp serves as a 2D approximation for the true affine transformation and thus leads to perspective errors, but they are hardly perceivable, even more so for the local warps in Chapter 4 because the perspective error is localized to much smaller regions.

5.3 Discussion

The main limitation of both the perceptual studies is that they analyze a simple form of image-based rendering restricted to axis-aligned geometry typical of façades. In Section 5.2, we analyze the tradeoff between ghosting and popping artifacts and give guidelines for resolving it, but our guidelines do not account for the complexity of the scene. Our analysis is also restricted to unstructured lumigraph [Buehler *et al.*, 2001] and cross-fading [Sinha *et al.*, 2009] while modern image-based rendering approaches are more sophisticated. Again, our analysis of perspective distortions in Section 5.1 is limited to very simple scene geometry. Although our results are very relevant to current commercial systems such as Google Streetview, they cannot evaluate the quality of state of the art image-based rendering approaches such as those presented in Chapters 3 and 4. A full fledged perceptual evaluation framework must extend to sophisticated image-based rendering algorithms and take into account the complexity of the scene and the quality of reconstructed 3D geometry. Psychophysical experiment design for such complex scerarios can be fairly non-trivial. For such cases, it may be simpler to study implicit visual process by measuring brain response directly [Mustafa *et al.*, 2012b]. Such studies can be of much utility, provided the relationship between conscious visual cognitive processes and implicit processes is well understood.

5.4 Conclusion

We analyze different kinds of distortions and give guidelines for designing systems that maximize perceived quality. Our analysis of blending and its associated artifacts reveals that applications should avoid blending excessively. We develop this guideline in a simple image-based rendering system, but this is a powerful result which applies to more sophisticated systems such as those in Chapters 3 and 4. Perspective distortions are the most important artifacts because they will always be present. However, our studies show that the human visual system is quite tolerant towards such distortions which leaves a lot of scope for image-based rendering approaches to experiment with algorithms that can work with sparse depth maps even though they incur distortions to some extent. We experimented with shape preserving warps in this spirit; there is promise for even better results in this direction of research.

Chapter 6

Virtual Reality using Image-based Rendering

Virtual reality (VR) uses computer graphics to immerse users in virtual environments. VR applications produce life size renderings of virtual environments around the user rather than on a computer screen as is common in a typical computer graphics application. This is generally achieved by specialized hardware such as multi-screen projection systems, head mounted displays etc. This is an exciting area of research with a lot of potential for gaming, training simulations and health applications such as cognitive therapy.

The bulk of research in VR focuses on hardware - multi-screen projection systems, stereo mechanism, tracking and haptics. Most of the software related research focuses on human-computer interaction where the target is to develop more intuitive mechanisms for interacting with the virtual environment using the same or enhanced hardware e.g. virtual navigation interfaces [Cirio *et al.*, 2012]. There is little research on the core rendering algorithm for virtual environments. This is rather surprising because VR requires large and possibly animated virtual environments be rendered at interactive rates in stereo. Therefore, current VR application use very trivial Phong shading or texture mapping. It is hard to use even rough approximations to global illumination, e.g. screen space ambient occlusion, simply because the frame rate drops significantly on large stereo displays.

The use of traditional graphics poses two main problems for VR systems. Firstly, large virtual scenes have to be modeled manually which can be time consuming and expensive. Secondly, large multi-screen stereo rendering lends a powerful sense of immersion but the rendering quality is often poor. We present the first image-based rendered (IBR) immersive system that alleviates the above problems. IBR systems use photographs: this makes capturing large environments fairly trivial compared to manual 3D modeling. More often than not, VR applications try to model real scenes, IBR can be a very convenient tool for such cases. At the same time, IBR gives photorealistic results which can potentially increase the sense of immersion in VR applications.

While a large number of IBR approaches are available for experimentation in VR systems, it is easy to see that approaches restricted to view interpolation [Zitnick *et al.*, 2004; Goesele *et al.*, 2010] and non-

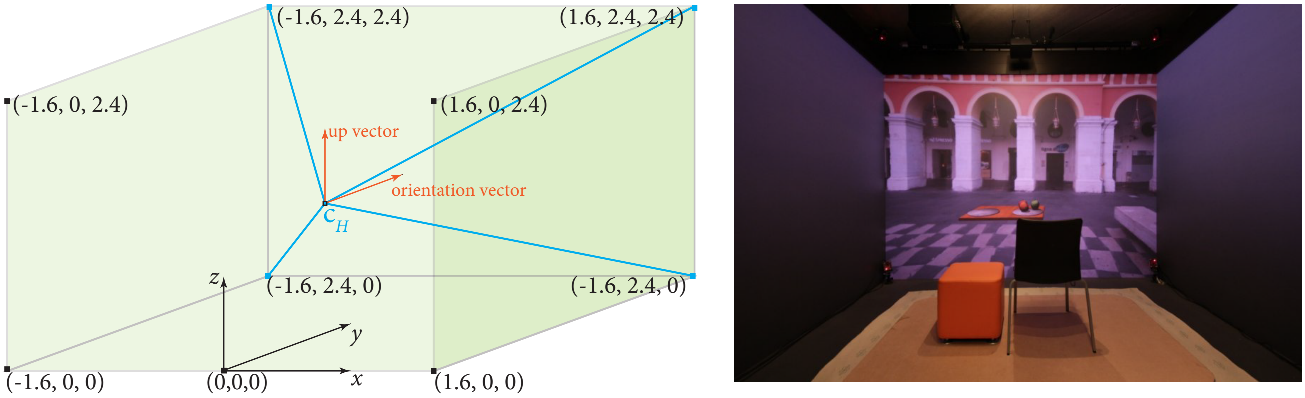


Figure 6.1: Immersive space setup. (a) View of the immersive space along with dimensions and the coordinate system used for computing the viewing frustum from any head position H inside the immersive space. (b) Photograph of the immersive space with image-based rendering result on the front screen. The chair and carpet are used in the Reminiscence Therapy experiment as shown in Figure 6.6.

interactive or offline approaches [Fitzgibbon *et al.*, 2005; Mahajan *et al.*, 2009] are clearly not suitable. After all, VR requires interactive navigation in the virtual environment. Other approaches based on 3D geometry [Buehler *et al.*, 2001; Eisemann *et al.*, 2008] require accurate 3D models which can be prohibitively large for virtual scenes that span tens of meters of urban imagery. As shown in Chapters 3 and 4, these approaches give a lot of artifacts if the 3D geometry is not accurate.

The above discussion shows that our image-based rendering approach is the first which can potentially be ported to a VR setup. Our approach allows free viewpoint navigation at interactive rates which is critical for VR. In this chapter, we describe the changes to the superpixel warp (see Chapter 4) in order to use it in a VR setup. We then briefly demonstrate the use of the our system in a cognitive therapy application. Finally, we discuss the current limitations of our system.

The contribution of the thesis towards this project is the development of a VR system that is capable of using free-viewpoint image-based rendering. This includes engineering issues associated with immersive space hardware setups as well as algorithmic challenges associated with developing an image-based rendering solution suitable for head tracked navigation. The cognitive therapy experiment described briefly in Section 6.5 that uses this setup is beyond the scope of thesis, please refer to [Chapoulie *et al.*, 2014] for more details.

6.1 Immersive space hardware setup

We use the superpixel warp for image-based rendering (see Chapter 4) which is designed for a single desktop screen. Consequently, we only use the front screen of a BARCO iSpace[1] (see Figure 6.1) for

[1] http://www.barco.com/en/Products-Solutions/Visual-display-systems/3D-video-walls/Multi-walled-stereoscopic-environment.aspx

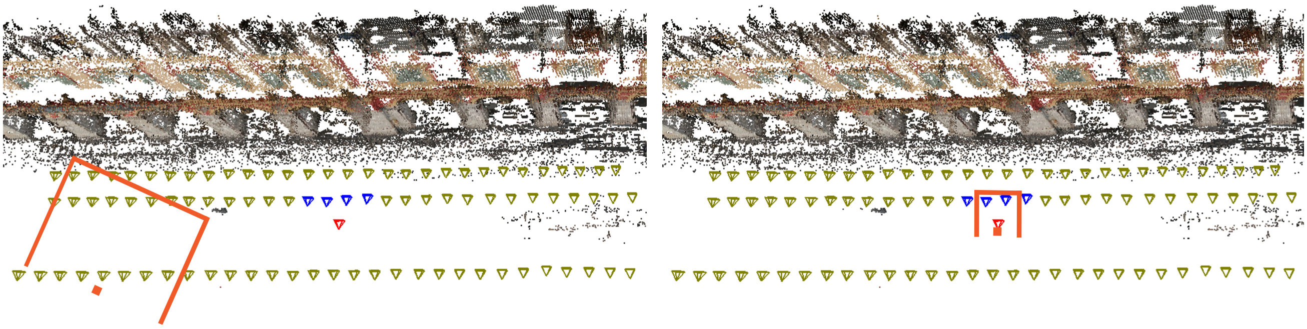


Figure 6.2: Left: visualization of capture cameras for one of the datasets. Such a capture allows navigation in a significantly large portion of the scene ($75m{\times}30m$ shown here as top view). The reconstructed scene and the immersive space (shown in orange) can be in totally different coordinate systems. Right: we register the immersive space (shown in orange) and the reconstructed scene by aligning one of the input camera positions with a reference point in the immersive space.

displaying the virtual environment. The problems with using all screens is discussed later in Section 6.6. The size of the screen is $3.2m{\times}2.4m$ with a display resolution of 1600×1200 pixels. The projectors use passive Infitec stereo glasses[2], which are tracked using the ART tracking system[3].

6.2 Capture and dataset preparation

We capture the scene using photographs as shown in Figure 6.2. The figure shows the scene captured using less than 100 photographs. This exemplifies the utility of IBR in such systems; large scenes such as the one shown can require a lot of man hours for 3D modeling. Capturing the whole scene using photographs required just 30 minutes in this case.

After capture, we preprocess the photographs similar to Chapter 4 by running 3D reconstruction [Snavely *et al.*, 2006; Furukawa and Ponce, 2009], oversegmentation [Achanta *et al.*, 2012] and depth synthesis in poorly reconstructed regions (see Section 4.2).

6.2.1 Registration of 3D scene and immersive space

The scale of the 3D scene and pose of input cameras are determined in an arbitrary coordinate system at by 3D reconstruction as shown in Figure 6.2. We first compute the scale factor between the actual scene and its reconstructed version as the ratio of the actual distance between two known points of the scene and the distance between their 3D reconstructed versions. This negates any scale variations between real world dimensions and reconstructed dimensions.

The 3D scene has to be displayed in the immersive space where the display camera matrices are

[2] http://infitec.net/index.php/home/glasses/infitec-premium-glasses
[3] http://www.ar-tracking.com

constructed using an absolute coordinate system fixed to the immersive space as shown in Figure 6.1. We therefore estimate a 3×3 rotation matrix R_{pre} and translation vector $\mathbf{t}_{\text{pre}}$ between the reconstructed scene and immersive space. The net transformation between the two coordinate systems is given by

$$T_{\text{pre}} = \begin{pmatrix} R_{\text{pre}} & \mathbf{t}_{\text{pre}} \\ \mathbf{0} & \mathbf{1} \end{pmatrix} \tag{6.1}$$

In order to estimate the rotation and translation, we select any input camera and align its center of projection and rotation matrix with a fixed reference position and rotation matrix in the immersive space respectively. Let the selected input camera's rotation matrix be R_i and the reference rotation matrix in the immersive space be R_{ref}. The projection of any 3D point in the rotation matrix of the input camera must be the same as the projection of the same point after applying the rotation R_{ref} in the reference rotation matrix. This invariance gives the following equation:

$$\begin{aligned} & R_i \cdot \mathbf{v} && = R_{\text{ref}} \cdot R_{\text{pre}} \cdot \mathbf{v} \\ \Rightarrow \quad & R_{\text{pre}} && = R_{\text{ref}}^T \cdot R_i \end{aligned} \tag{6.2}$$

Note that the inverse of a rotation matrix is the same as its transpose. Next, assume that the input camera's center of projection is $\mathbf{c}_i$ and the reference position is $\mathbf{c}_{\text{ref}}$. Applying the rotation and translation we get:

$$\begin{aligned} & R_{\text{pre}} \cdot \mathbf{c}_i + \mathbf{t}_{\text{pre}} && = \mathbf{c}_{\text{ref}} \\ \Rightarrow \quad & \mathbf{t}_{\text{pre}} && = \mathbf{c}_{\text{ref}} - R_{\text{pre}} \cdot \mathbf{c}_i \\ \Rightarrow \quad & \mathbf{t}_{\text{pre}} && = \mathbf{c}_{\text{ref}} - R_i^T \cdot R_{\text{ref}} \cdot \mathbf{c}_i \end{aligned} \tag{6.3}$$

We transform the whole 3D scene including the input cameras by the rigid transform T_{pre} computed from Equations 6.1, 6.3 and 6.3. We apply T_{pre} on each 3D depth sample of the scene and transform the input cameras by transforming their centers of projection as well as the modelview matrices as explained in Appendix C. This step is an additional part of preprocessing which needs to be performed once to register the 3D scene with the immersive space (as shown in Figure 6.2).

6.3 Modification of IBR for immersive space

We use the same image-based rendering approach as in Chapter 4 except for the following modifications to handle head tracking, scene navigation and stereo.

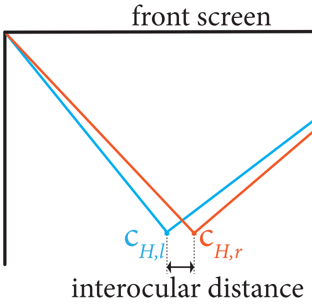


Figure 6.3: Top view of immersive space to show the frusta for left and right eye. The modelview matrices and frusta for left and right eye are computed using the head positions $\mathbf{c}_{H,l}$ and $\mathbf{c}_{H,r}$, which are computed by perturbing the head position returned by the head tracker in left and right directions parallel to the screen.

6.3.1 Head tracking

The approach in Chapter 4 is developed for desktop applications where the user navigates using a mouse. In the immersive space setup, the novel camera position $\mathbf{c}_H$ is provided by a head tracking device in real time. The captured 3D scene is already registered with the immersive space such that one of the input cameras is aligned with a known reference point in the immersive space.

The virtual camera used to display the front screen has orientation $(0, 1, 0)$ and up vector $(0, 0, 1)$ as shown in Figure 6.1. These vectors remain the same irrespective of the position and orientation of the head. Thus, in OpenGL terminology, the modelview matrix is given by:

$$M_{f,H} = \texttt{gluLookAt}\,(\mathbf{c}_H, \mathbf{c}_H + (0, 1, 0), (0, 0, 1)) \tag{6.4}$$

where the subscript f denotes the front screen. We select input images whose modelview matrices are most similar to that of the novel camera. Thus, to synthesize the final image, we select the input images whose center of projection and orientation are closest to $\mathbf{c}_H$ and $(0, 1, 0)$ respectively, in the coordinate system of the immersive space. We compute the perspective matrix $P_{\mathrm{f,H}}$ using the standard approach of joining the head position $\mathbf{c}_H$ with the four corners of the screen [Cruz-Neira *et al.*, 1993]. Once we pre-select the input images, we warp them using the overall projection matrix given by $P_{f,H} \cdot M_{f,H}$. We blend the warped images as described in Section 4.4.

6.3.2 Stereo rendering

For stereo rendering, we render two frames into the OpenGL left and right buffers, each computed independently using the same image-based rendering pipeline. For each head position $\mathbf{c}_H$, we create two head positions $\mathbf{c}_{H,l}$ and $\mathbf{c}_{H,r}$ separated by interocular distance of 6-8 cm such that the line joining $\mathbf{c}_{H,l}$ and $\mathbf{c}_{H,r}$ is parallel to the screen (see Figure 6.3). At each frame, we compute an IBR result corresponding to each of $\mathbf{c}_{H,l}$ and $\mathbf{c}_{H,r}$ and display them using quad-buffered stereo.

6.3.3 Navigation

The above approach seamlessly handles head movement within the immersive space. To allow long range navigation, we use the *wand* device which has a joystick and a orientation sensor. The hardware returns the current orientation of the wand and movement of the joystick. We interpret the joystick as positive or negative translation in the direction given by the orientation of the wand. This gives an overall transformation T for the position of the user in the virtual world which can be achieved by transforming the head position by T. However, recall that the head position is updated asynchronously by the head tracker. Therefore, instead of transforming the head position by T, we transform the scene and input cameras by T^{-1} with equivalent effect. To transform the scene, we apply T^{-1} on each 3D depth sample used for IBR. We transform the input cameras by transforming the center of projection as well as the modelview or extrinsic matrix. We explain this derivation in Appendix C.

While our system provides very realistic images for free navigation, like any image-based method it is restricted to representing content which actually exists in some of the input images. When the user leaves this region, visual artifacts appear. To avoid this, we limit navigation to the zone where artifacts are very small. This still leaves sufficient room for free navigation by actual movement within the immersive or long range translation using the wand.

6.3.4 Rendering synthetic objects with IBR

Capturing and modeling a real scene can provide a very realistic 3D environment. However, it may not contain all the desired scene elements required for a VR application. This is especially true for training or simulation applications where it may be critical to have additional objects for interaction (see Figure 6.6(right) for example). Such objects have to be modeled or captured separately and added to the virtual environment. To allow this, we modify the IBR pipeline of Chapter 4 to add other objects. Recall that we store the median depth of each superpixel as metadata while warping it to the novel view (see Section 4.4.1). During the second pass for blending, we reproject this median depth for each candidate superpixel to the novel view using the following:

$$d_{\mathrm{f,novel}} = C_{\mathrm{f,novel}} \cdot C_{\mathrm{input}}^{-1} \cdot d \tag{6.5}$$

where d is the median depth of the superpixel, C_{input} and C_{novel} are the projection matrices of the input and novel camera respectively. While blending the candidate superpixels, we write the reprojected depth of the highest weighed candidate as the final depth of the pixel.

Having rendered the full scene using the image-based pipeline, we then render synthetic objects. These objects are typically created in 3D modeling software such as Autodesk Maya or Blender etc. We therefore use Phong shading with texture mapping to render these objects. We render them in the same

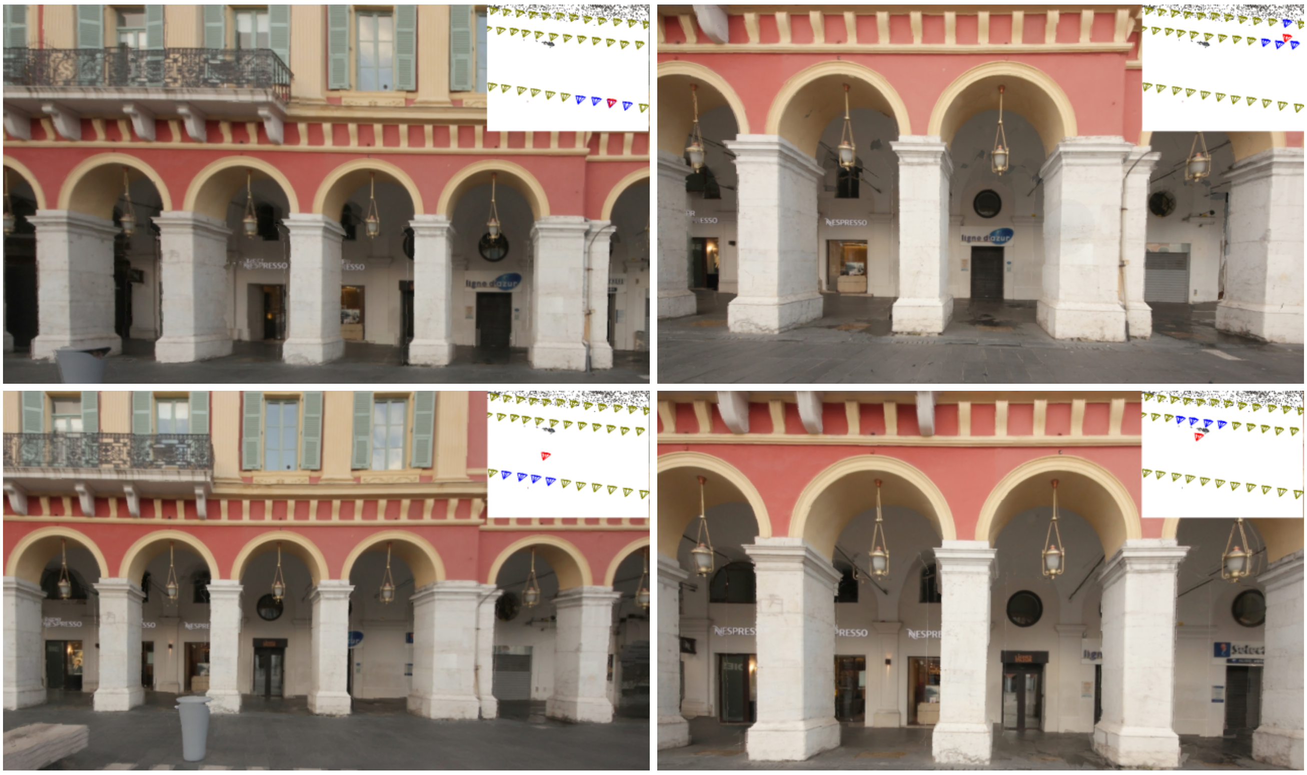


Figure 6.4: Renderings generated by our VR system. These screenshots are generated by mirroring the VR renderings on a desktop monitor. The inset is not displayed in the VR renderings. It is shown here to illustrate the head position in the 3D scene during interactive navigation.

render target as the IBR result with depth test enabled. This automatically places the objects at the correct depth in the scene giving correct (dis)occlusion effects. Thus, our VR system can be cleanly converted into an augmented reality system where the most of the scene is rendered using IBR and synthetic objects are rendered using the traditional graphics pipeline.

6.4 Results

We now present the results of our image-based rendering system in the immersive space in Figure 6.4 and 6.5. The overall frame-rate of the implementation is 15 FPS; higher resolution of the immersive

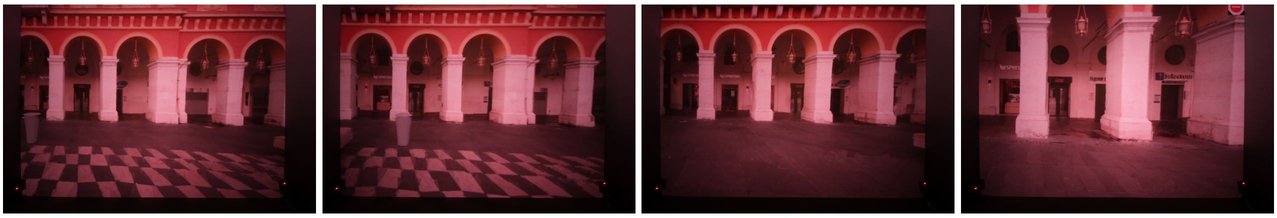


Figure 6.5: Photographs of our VR system running in the immersive space.

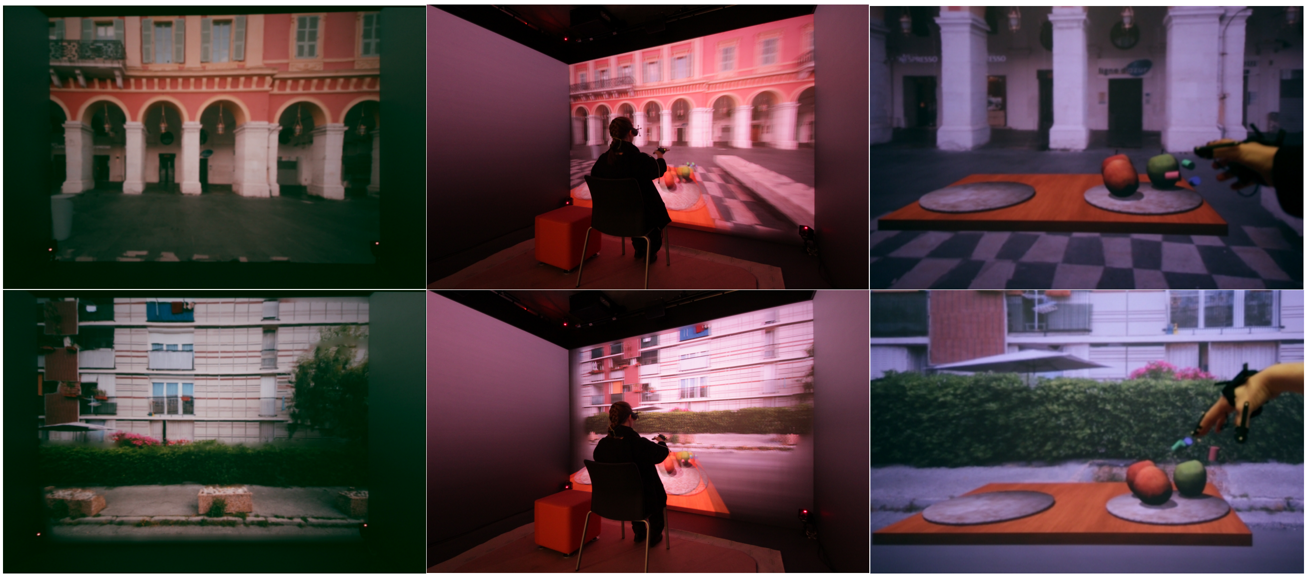


Figure 6.6: Reminiscence Therapy experiments. Left: different datasets used as known and unknown landmarks of the city. Middle: user immersed in our image-based VR system. Participants are seated to accommodate possible mobility restrictions. Right: Our VR system enhanced with finger gestures and synthetic objects.

space and stereo rendering account for the decreased frame-rate as compared to Section 4.5.

6.5 Application: Reminiscence Therapy

We demonstrate the VR prototype by using it in a Reminiscence Therapy (RT) experiment. In Reminiscence Therapy, patients are presented with familiar environments, e.g. their neighborhoods, prominent landmarks of their city etc. Traditional RT accomplishes this by means of photographs. VR can be used for the same purpose by modeling the environments and allowing the patient to interact with the environments virtually. This is based on recent work that show the utility of VR for memory treatments [Brooks and Rose, 2003; Gonneaud *et al.*, 2012]. However, creating a realistic 3D models of a patient's familiar environments using traditional manual modeling is far too expensive both in time and resources. We hypothesize that the realistic renderings using IBR are equally or more powerful than photographs or synthetic scenes while also making it much easier to capture such environments. We develop experiments where familiar and unfamiliar landmarks of the city are presented to elderly participants, their memory response measured by means of a questionnaire and compared to the case where still photographs are used as stimulus. Some photographs of the experiments can be seen in Figure 6.6.

This experiment including the design, analysis and inferences are beyond the scope of the thesis. We give a brief overview of the experiment and results to demonstrate the utility of our VR system. Please refer to [Chapoulie *et al.*, 2014] for more details.

Experiment The participants were presented with different stimuli for 2 minutes and asked to generate as many memories as possible related to the environment, using short sentences. The verbal reponses were recorded and analyzed by a speech linguist who classified them as concious or vague recollections. We only count concious recollections for our purpose. The different stimuli included (a) image-based VR rendering of a known city landmark, (b) image-based VR rendering of unknown city landmark, (c) static photograph of a familiar landmark and (d) grey image which represents no visual stimulus. The number of recollections in each case is expected to indicate the utility of the four settings for Reminiscence Therapy. We performed the experiment on elderly participants and not patients since this project is still in its early stages and the immersive space is not an authorized clinical laboratory.

Results The most important result of the study is that the number of recollections was highest for image-based VR renderings of a familiar landmark followed by static photograph of familiar landmark. This indicates that realistic immersive visualization of 3D environments surpasses static photographs for trigger conscious recollections of autobiographical memory, which is the main purpose of RT. Our system can be easily adapted to environments familiar to the patient; this holds particular promise from a clinical perspective, since it makes VR a viable clinical procedure. The responses to our questionnaires indicate the ability of IBR to convey a sense of "being there". We also got conclusive evidence that immersion in familiar landmarks of the city generated more memories, which indicates that the level of realism in our system indeed makes a virtual environment more recognizable.

Besides, our studies confirm the acceptability of sophisticated VR technology by elderly participants. The responses to the questionnaires indicate that the technological setup is well tolerated by the participants. In fact, being a novel technology, it is more likely to engage the participants than traditional approaches. One of the great challenges for RT is whether different technologies maintain the user's motivation when confronted them with a repetitive series of training challenges. The interactivity and realism seem to improve motivation and engagement.

These results show that image-based techniques offer great promise for RT, and for VR in general. Our system has numerous advantages over traditional 3D assets used in VR. The fact that only a few casual photographs are required to create a scene that can be used for VR is an advantage with very significant consequences. The level of realism obtained by the imagery, despite some residual artifacts, is at least as good as that produced at great cost with manual modeling. This is an experimental setup with many technical challenges, the most important being a multi-screen IBR solution (see Section 6.6). Overcoming these difficulties will further increase the utility of such systems for clinical procedures.

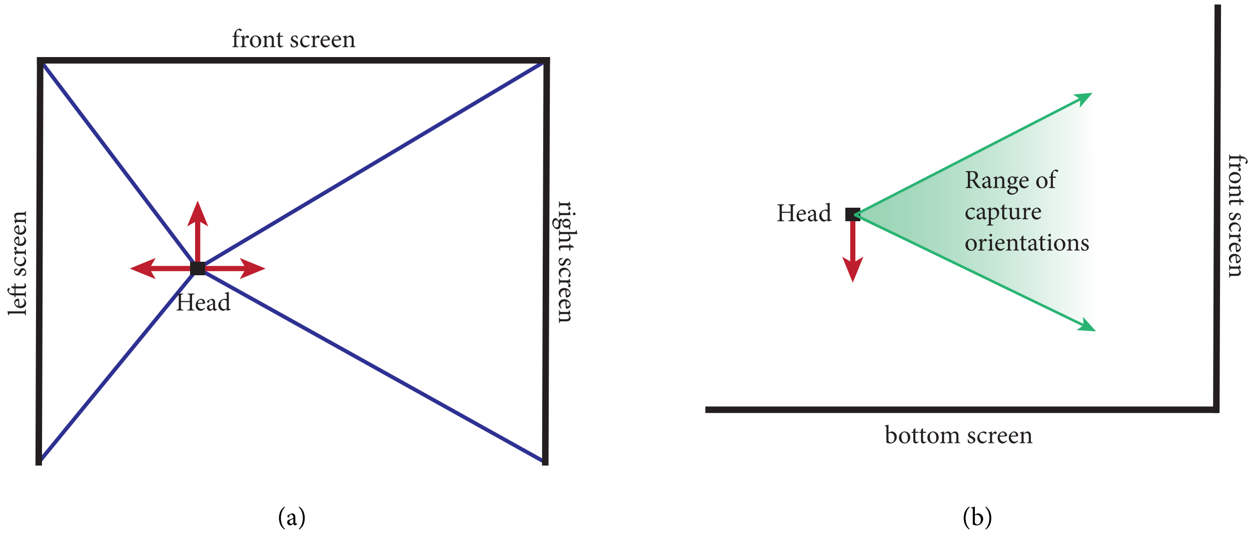


Figure 6.7: (a) Immersive space top view: orientation vectors of camera used to display front, left and screens are shown in red. Pre-selecting input images using these orthogonal orientations leads to inconsistencies at screen boundaries because different images are selected for different screens. (b) Immersive space side view: the orientation for bottom screen display camera (red) is almost orthogonal to the capture cameras orientations (green). The scene floor is captured at grazing angles and warping these superpixels over a large angle between capture cameras and bottom screen camera leads to artifacts.

6.6 Current technical issues and possible solutions

The most pressing direction of future work is the improvement of the underlying IBR algorithm. This includes the development of a solution for multiple screens, thus providing full immersion.

Limitation to single screen The image-based rendering approach of Chapter 4 pre-selects a subset of images to warp. The modelview matrices of the cameras used to display the front, left, right and bottom screens in the immersive space have orthogonal orientation vectors as shown in Figure 6.7(a). This results in different images being pre-selected for different screens. This in turn leads to discontinuities at the boundaries of the screens. A potential solution is to select a set of superpixels (not images) depending upon head orientation and not the orientation of display camera for each screen.

Special case for scene floor The shape-preserving warp of Chapter 4 allows only minor depth gradients within each superpixel. In order to display the scene floor on the bottom screen, the floor superpixels have to be warped to the camera for bottom screen. As shown in Figure 6.7, the floor is captured at grazing angles by horizontal capture cameras (shown in green). This leads to warping artifacts (see Section 4.7) because these superpixels have a high depth gradient and they have to be warped over the wide angle, determined by the angle between capture orientation and bottom camera orientation (shown in red). A potential solution is to relax the priors of the IBR approach or simply use depth sample reprojection [Chen *et al.*, 2011] instead of shape-preserving warp (see Section 4.3) for the floor of the scene.

Chapter 7

Conclusions and Future Work

7.1 Conclusions

We have presented a set of new approaches to provide plausible image-based rendering for navigation in casually captured complex scenes, which are difficult to handle using state of the art multi-view stereo and image-based rendering.

The most important insight obtained from this thesis is that 2D image-based constraints can be used compensate for 3D reconstruction. It is not always possible to obtain high quality geometry with state of the art approaches. Even with improvements in multi-view stereo some cases can still be expected to be problematic. We have demonstrated that in cases when photoconsistency can be very hard to enforce, it is still possible to obtain plausible image synthesis by using image-based constraints. It is arguably a better strategy to store view dependent representations (e.g., view dependent depth maps and warp systems populated with 2D constraints), for some scene objects rather than a viewpoint agnostic static representation like 3D models, especially if 3D models are hard to obtain or very complex.

In terms of specific contributions, we believe that the introduction of discontinuous variational warp in Chapter 3 breaks new ground. In particular, we have demonstrated how to introduce content preserving discontinuities in a smooth variational warp. This will hopefully have applications in other image related tasks. The depth synthesis in Chapter 4 is the first approach which synthesizes plausible depth in a non-local manner; it is capable of finding plausible depth from relatively distant image regions while previous approaches have only tried local depth propagation [Yang *et al.*, 2007; Goesele *et al.*, 2010].

We have also developed a shape-preserving warp that ensures distortion free image synthesis even though the guiding constraints in the form of depth samples are sometimes approximate. The warp applied independently to each superpixel preserves the structures within superpixels, across superpixels as well as across time (during an interactive walkthrough). No global constraints are necessary to ensure spatial or temporal coherence. Similar approaches have been used before for warping images [Liu *et al.*, 2009, 2013]; our work is the first to demonstrate these warps applied to local regions instead of the full image, and to combine them with occlusion handling, parallax effects and view synthesis by blending multiple warped images. Moreover, our solutions are also robust for situations where input images have to be warped over significant baselines of nearly 10 meters.

We have evaluated common artifacts relevant to image-based rendering using perceptual analysis. Our studies reveal that spatial ghosting artifacts are more objectionable compared to sporadic temporal discontinuities, which indicates that observers prefer crisp renderings rather than smooth transitions. We develop a quantitative model that correlates the perceived level of perspective distortions with capture and rendering parameters, and also show that observers are more tolerant towards perspective distortions as compared to ghosting artifacts.

Lastly, this thesis is the first research work in image-based rendering to target free viewpoint navigation; previous work has largely focused on view interpolation. We believe that future image-based rendering approaches, in an attempt to improve upon the results of this thesis, will consider this an indispensable goal.

7.1.1 Research impact and deployment

The results of this thesis are being used in the EU IP project VERVE[1] for developing virtual reality applications for Reminiscence Therapy (see Chap. 6). In this context, the newly built *Centres Mémoire de Ressources et de Recherche*[2] (memory center) in Nice has installed a VR setup which will use our system in a clinical setting. The results also inspired the EU IP project CR-PLAY[3] which targets image-based rendering as a tool for content generation for lightweight games. The companies involved in this project are persuaded that improved versions of our solutions will result in significant savings to their production costs. The rendering aspect of the French project ANR SEMAPOLIS[4] is also based on our ideas. Besides, major industrial players have expressed interest in evaluation of techniques presented in this thesis.

7.2 Future work

The limitations of our image-based rendering approaches provide new directions for research. Our shape preserving warp assumes largely fronto-parallel depth which results in artifacts on surfaces captured at grazing angles. This could be resolved by formulating the warp so as to account for local depth variation.

[1] http://www.verveconsortium.eu/
[2] http://www.cmrr-nice.fr/
[3] http://cordis.europa.eu/fp7/ict/creativity/creativity-projects-fp7_en.html
[4] https://project.inria.fr/semapolis/

Computer animation and geometry manipulation literature may provide insights.

Current approaches, including ours, have problems handling very thin objects such as lamp posts, gratings, railings etc. This is also the case for non-lambertian surfaces such as reflections, transparencies etc., although reflections have been handled in restricted settings [Sinha *et al.*, 2012]. Significant algorithmic changes would be required to provide robust solutions to these cases.

We use basic Poisson synthesis to fill holes in the final result, there is need for more sophisticated algorithms for this purpose. Inpainting [Criminisi *et al.*, 2003], combined with recent acceleration techniques e.g., PatchMatch [Barnes *et al.*, 2009], could provide a basis for such a solution.

Our depth synthesis and blending approaches can benefit from object cosegmentaion [Kowdle *et al.*, 2012] which simultaneously segment all images of a multi-view dataset while also building a correspondence graph of segments across different images.

Our depth synthesis uses a graph based approach inspired by Geodesic methods in computer graphics [Criminisi *et al.*, 2010]. Alternatively, MRF based formulations [Zitnick and Kang, 2007] might improve the results. An important challenge in depth synthesis is to disambiguate between superpixels based on visual content. We compare color histograms which can be problematic in case of shadows or structured repetitive textures such as checkerboard pattern on a façade. More robust results can be obtained by computing the intrinsic decomposition of the image [Bousseau *et al.*, 2009; Laffont *et al.*, 2013] and comparing the albedo.

Although real time, our approach is still more computationally demanding than simple projective texture mapping of piecewise-planar reconstructions [Sinha *et al.*, 2009; Gallup *et al.*, 2010] which can be used very effectively for simpler scenes. Hybrid approaches can be developed which use planar reconstruction wherever possible and switch to more complex approaches such as ours for non-planar or irregular regions. This might also alleviate some of the limitations of our approach such as the problem with surfaces captured at grazing angles.

Most modern image-based rendering systems are real time, however they consume a large amount of CPU memory for 3D models or point clouds and GPU memory for textures. These requirements make it difficult to port these applications on mobile devices. Fortunately, multi-view scenes contain a lot of redundant data because the same visual content is present in a large number of images. Depending upon the memory budget, this redundancy can be exploited to compress multi-view scenes into texture atlases.

Our perceptual studies of visual artifacts (Chapter 5) are limited to simple image-based rendering setups on simple scenes. The immediate next step is to generalize these studies to more complex settings. In our experiments, we have shown an indirect comparison between perspective distortions and ghosting artifacts; it would be interesting to quantify this comparison.

7.2.1 Long term research directions

Current image-based rendering approaches restrict themselves to using 3D reconstruction and image segmentation as input. Machine learning approaches can provide much more data about the scene content from photographs, which has never been utilized for the purpose of image-based rendering. In recent work, Xiao [2012] learn the semantic layouts of common scenes and use this to improve 3D reconstruction. Scene understanding can also be used to improve the results of different stages of image-based rendering, e.g., computing superpixels of the ideal size, providing strong priors for our depth synthesis and better blending heuristics to improve the tradeoff between ghosting and temporal artifacts.

Another recent development has been the popularity of RGBD cameras, especially Microsoft Kinect Fusion which is capable of producing high quality 3D models for indoor scenes [Chen *et al.*, 2013; Nießner *et al.*, 2013; Zhou *et al.*, 2013]. While this does not necessarily help our context of urban imagery, related depth sensing technologies like LIDAR can be of much utility. The additional data can be used to push image-based rendering to new levels of sophistication such as reducing the number of input images from 10–20 in our approach to 2–3 for each building, handling dynamic scenes, using varying illumination datasets such as community photo collections etc.

Relighting [Laffont *et al.*, 2012] and computational photography [Shih *et al.*, 2013] techniques can drastically manipulate the appearance of photographs using other exemplars. Basic forms of relighting will help remove micro-variations in illumination between photographs captured during a single session. The combination of relighting and viewpoint manipulation will allow seamless transitions between different lighting conditions during a free viewpoint walkthrough, without the need for capturing the full scene under different lighting conditions. This will eliminate one of the most significant limitations of image-based rendering which is the fact that the lighting of a captured scene is fixed; identical captures are required under different lighting conditions to change the appearance of the walkthrough.

The underlying principle of relighting described in [Laffont *et al.*, 2012, 2013] is to use 2D image-based constraints that directly target novel view synthesis instead of relying on accurate intermediate representations such as a 3D mesh. Since our approach is based on a principle similar in spirit, it seems feasible to unify our viewpoint manipulation with appearance manipulation in a single framework, effectively adding time-lapse capabilities to image-based rendering.

All image-based rendering approaches model a single scene at a time, i.e., they expect all images of a multi-view dataset to be of the same 3D scene. It is possible to combine several scenes into a single dataset by aligning their coordinate systems appropriately and rendering everything simultaneously. Our treatment of synthetic objects in image-based rendering setups (see Chapter 6) is the first step in this direction. However, this involves significant challenges related to coherent illumination across scenes, seamless transitioning between scenes, guided capture process etc. This could be a useful tool for 3D modeling of large scenes, where some assets can be traditional 3D models with baked textures and others can be acquired 3D scenes rendered using image-based rendering.

Multi-view techniques require a minimal set of images captured in a particular manner to produce high quality results. It is impractical to expect a naive user to understand these guidelines. Internet photo collections can be of much utility for augmenting personal photographs, the combined set can be used for very high quality results. Our emphasis on casual captures using only handheld cameras is inspired by the ultimate goal of using arbitrarily unstructured internet photo collections. Appearance manipulation techniques are indispensable for this goal since photographs are expected to have varying appearances.

Concluding remarks This thesis demonstrates the utility of image-based rendering by pushing it to new levels of sophistication such as wide baseline imagery, free viewpoint navigation, robustness towards scene complexity, and perceptually based guidelines. Future developments, coupled with the increasing involvement of the industry and ever expanding capabilities of related commercial systems like Microsoft Photosynth, Bing Maps, Google Streetview etc., promise exciting new applications for image-based rendering.

Appendices

Appendix A

Limitations of High-level Image Segmentation

As stated in Section 3.3.1, we use manual intervention instead of automatic image segmentation to extract silhouette polylines for the silhouette-aware warp. In this section, we present results of experiments with object classification and image segmentation and highlight their limitations, which in turn justify manual intervention for the purposes of Chapter 3. The limitations of high level image segmentation presented here inspired the use of low level image oversegmentation in Chapter 4, which successfully eliminated the need for manual intervention.

An automatic approach for extracting silhouettes polylines would consist of two steps, (a) extract irregular contours at depth discontinuities using image segmentation, and (b) convert the irregular contours into polylines. We present the results of our experiments with both the steps.

Contour extraction Most image segmentation algorithms extract pixel contours or edge maps. Since we need polylines for our silhouette-aware warp (Chapter 3), we first extract irregular contours and then convert them into polylines.

A comparison between different segmentation algorithms is shown in Figure A.1, along with the ideal silhouette polylines required by the approach of Chapter 3, generated using manual intervention. The first row shows one of the input images of our datasets. Second row shows the required silhouettes marked manually. The third row shows the final result of occlusion boundary extraction from single image [Hoiem *et al.*, 2007b]. The fourth and fifth rows show the soft and binary edge maps using a combination of [Arbelaez, 2006] and [Maire *et al.*, 2008]. The last row shows hierarchical segmentation [Arbelaez *et al.*, 2011] using results from the fifth row.

Clearly, all these approaches perform fairly well but none of them is 100% accurate. The difference between the required silhouettes (Figure A.1(second row)) and results of all other approaches is quite large in some cases. Hoiem *et al.* [2007b] works the best in most cases, but the results often miss some silhouettes. This is a common problem with techniques based on machine learning – their results are impressive but not 100% accurate for any single input image.

Note that semi-automatic user interfaces like Adobe Photoshop's Magic Wand or Lasso tools can also be used to extract irregular contours at depth discontinuities. This is an alternative to segmentation algorithms which guarantees accuracy. However, as we demonstrate in the following section, it is very hard to convert these contours into polylines.

Contour to polyline conversion The edge maps or contours extracted from the previous step using segmentation approaches or semi-automatic methods can be converted into polylines using contour tracing [Teh and Chin, 1989] and polygon approximation [Douglas and Peucker, 1972]. However, contour tracing becomes ambiguous in the presence of many intersecting contours and the result has many doubled line segments and noise (see Figure A.2(c)).

In comparison, the manual silhouette annotation used in Chapter 3 took less than 40 seconds for each image, which is better than the options described above. Hence, instead of the above options, we use manual silhouettes in Chapter 3 and then automate it using image oversegmentation in Chapter 4.

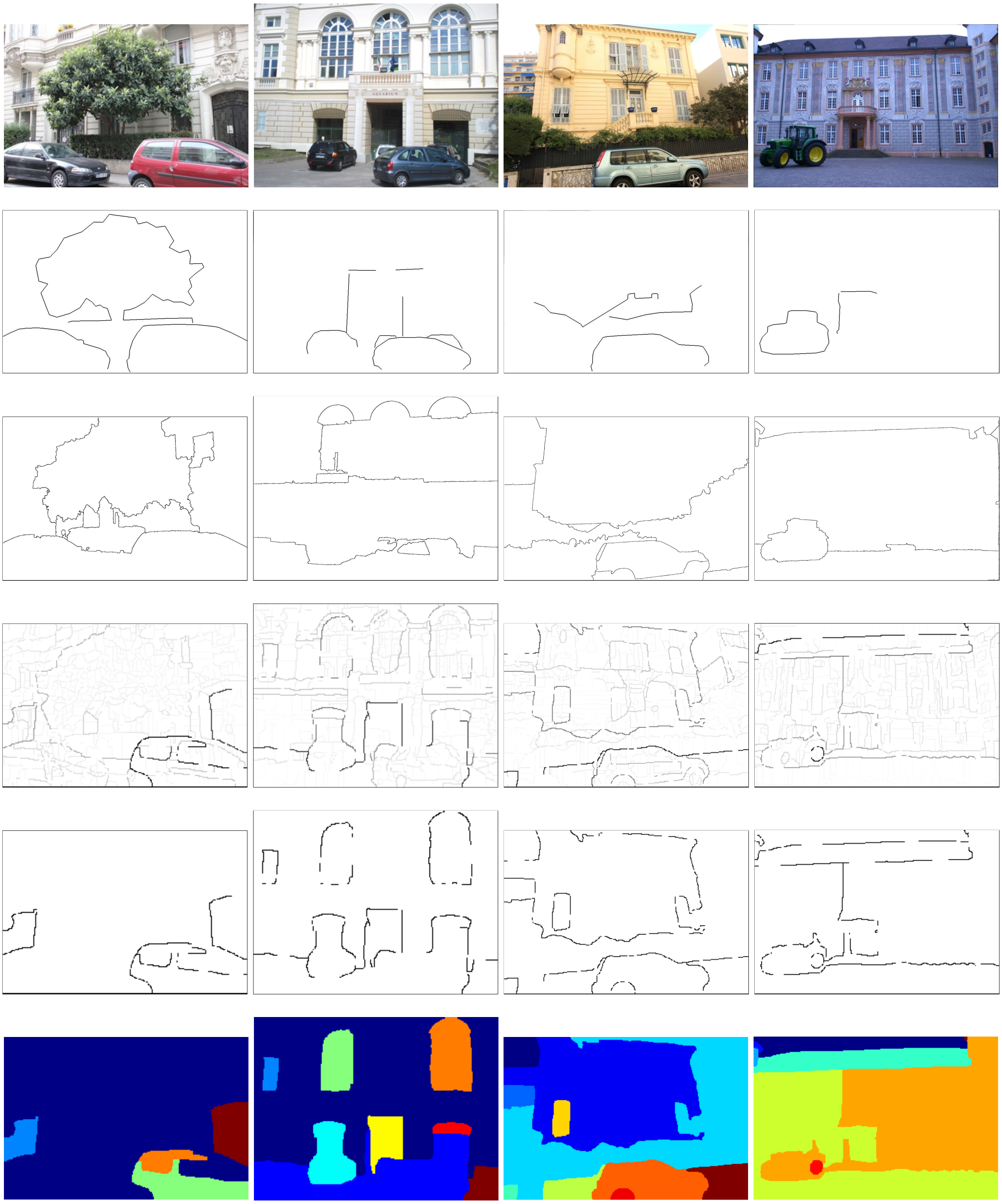
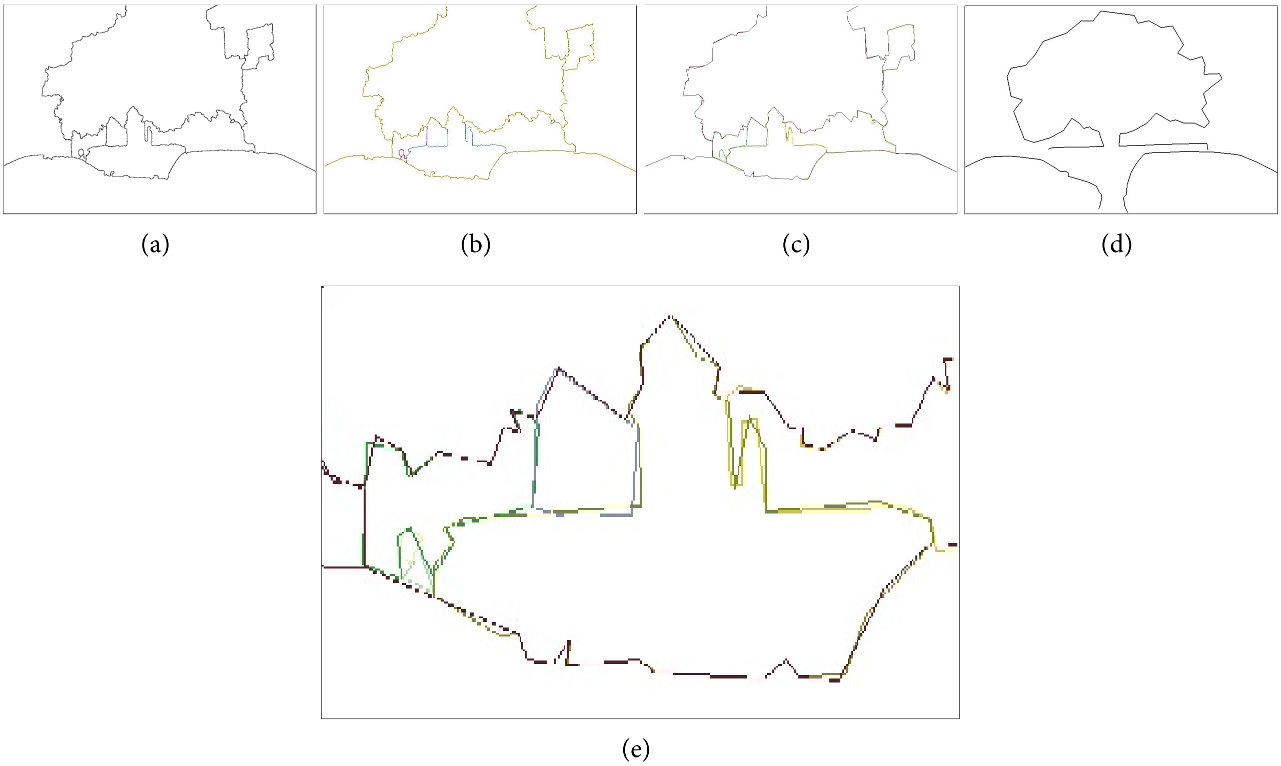


Figure A.1: Automatic silhouette extraction results. Top row: input image. The second row: manually annotated silhouettes. Third row: final output of [Hoiem *et al.*, 2007b]. Fourth row: soft edge maps generated using [Arbelaez, 2006] + [Maire *et al.*, 2008]. Fifth row: binary edge map obtained by thresholding the result in the fourth row. Sixth row: Hierarchical segmentation [Arbelaez *et al.*, 2011]. Note that the result of automatic segmentation are similar to manual annotation only for first and last dataset, and even then the localization is not very good.



(a) (b) (c) (d)

(e)

Figure A.2: (a) Binary edge map using [Hoiem *et al.*, 2007b], (b) contour tracing with each contour shown in a different color, (c) polygon approximation of the contours (d) manual silhouettes, and (e) zoomed in view of line segments in (c). Note the jagged contours and double line segment for each contour in (c), which make this unsuitable for silhouette-aware warp (see Section 3.4.1).

Appendix B

Depth Synthesis for Sky Regions

The depth synthesis (see Chapter 4.2) adds plausible depth to regions which have very few or no depth samples. In this section, we describe the special case of depth synthesis for image which have significant sky regions, which was needed for University and ChapelHill2 datasets in our experiments.

Our depth synthesis approach can synthesize depth values on objects which have *some* though sparse depth samples. Large regions of sky typically have no depth samples at all. We identify such sky regions in the image using a graph-cut. We assume that the images are captured upright and sky pixels are close to the top border. We create a graph with all the pixels of the image as nodes and add edges between adjacent pixels. The label costs for the graph cut are given in the following table. We keep a very high penalty of 10^6 for having neighboring pixels with different labels, except at superpixel boundaries where we relax it to 100. After computing the graph cut using [Kolmogorov and Zabih, 2004], we mark the pixels labeled 0 as sky and assign them 99th percentile depth of the image. Note that [Hoiem *et al.*, 2007a] may be used to identify sky regions; we resort to this approach because it is sufficient and much faster.

Pixel	Label 0 cost	Label 1 cost
Pixels along top border contained in superpixels with no depth samples	0	10^6
All other pixels contained in a superpixel with no depth samples	1	0
All other pixels	10^6	0

Appendix C

Transformation of Camera Matrices to Immersive Space

In Section 6.2, we transform the entire 3D scene to the coordinate system of the immersive space. To this end, we apply a rigid transformation to the whole scene. The transformation of all 3D reconstructed points is trivial. In this section, we describe the transformation of the extrinsic or modelview matrices of the input cameras to the immersive space.

Assume the entire scene including the input cameras, has to be transformed by the matrix M which comprises a uniform scale s, rotation R_M and translation T_M. Any 3D point $\mathbf{x}$ in homogeneous coordinates can be transformed by applying M.

$$\bar{\mathbf{x}} = M \cdot \mathbf{x} = sR_M \cdot \mathbf{x} + T_M \tag{C.1}$$

Clearly, M is invertible because the scale, rotation and translation are invertible.

Consider an input camera with original perspective matrix (or frustum) F, rotation matrix R and center of projection $\mathbf{v}$. The camera extrinsic or modelview matrix is given by:

$$\begin{pmatrix} R & -R \cdot \mathbf{v} \\ 0 & 1 \end{pmatrix} \tag{C.2}$$

The projection of any point $\mathbf{x}$ in this camera is given by

$$\begin{aligned} \mathbf{y} &= F \cdot \begin{pmatrix} R & -R \cdot \mathbf{v} \\ 0 & 1 \end{pmatrix} \cdot \mathbf{x} \\ &= F \cdot (R \cdot \mathbf{x} - R \cdot \mathbf{v}) \end{aligned} \tag{C.3}$$

While transforming the entire scene, the camera's rotation matrix and center of projection change but the frustum remains the same because it is an intrinsic property of the camera. Let the new camera position and rotation matrix be $\bar{\mathbf{v}}$ and $\bar{R}$ respectively. The projection of a scene point $\mathbf{x}$ using the original camera should be the same (up to constant factor) as that of the transformed point $\bar{\mathbf{x}}$ using the transformed camera.

$$\begin{aligned} F(R \cdot \mathbf{x} - R \cdot \mathbf{v}) &\sim F\left(\bar{R} \cdot \bar{\mathbf{x}} - \bar{R} \cdot \bar{\mathbf{v}}\right) \\ &\sim F \cdot \left(\bar{R} \cdot M \cdot \mathbf{x} - \bar{R} \cdot \bar{\mathbf{v}}\right) \end{aligned} \tag{C.4}$$

This gives the following equations

$$R \cdot \mathbf{x} \sim \bar{R} \cdot M \cdot \mathbf{x}, \quad R \cdot \mathbf{v} \sim \bar{R} \cdot \bar{\mathbf{v}} \tag{C.5}$$

Solving these two, we get the rotation matrix and center of projection of the transformed camera.

$$\bar{R} \sim R \cdot M^{-1}, \quad \bar{\mathbf{v}} \sim M \cdot \mathbf{v} \tag{C.6}$$

Bibliography

Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., and Süsstrunk, S., 2012. SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(11):2274 –2282. ISSN 0162-8828.

Adams, K.R., 1972. Perspective and the viewpoint. *Leonardo*, 5(3):209–217.

Agarwal, S., Furukawa, Y., Snavely, N., Curless, B., Seitz, S.M., and Szeliski, R., 2010. Reconstructing rome. *IEEE Computer*, 43(6):40 –47. ISSN 0018-9162.

Agarwal, S., Snavely, N., Simon, I., Seitz, S.M., and Szeliski, R., 2009. Building rome in a day. In Proc. ICCV, 72–79.

Alexa, M., Behr, J., Cohen-Or, D., Fleishman, S., Levin, D., and Silva, C.T., 2001. Point set surfaces. In Proc. IEEE Visualization, 21–28. ISBN 0-7803-7200-X.

Aliaga, D., Funkhouser, T., Yanovsky, D., and Carlbom, I., 2003a. Sea of images: A dense sampling approach for rendering large indoor environments. *IEEE Computer Graphics & Applications*, 23(6).

Aliaga, D.G., Yanovsky, D., Funkhouser, T.A., and Carlbom, I., 2003b. Interactive image-based rendering using feature globalization. In Proc. I3D, 163–170.

Andreetto, M., Zelnik-Manor, L., and Perona, P., 2008. Unsupervised learning of categorical segments in image collections. In Proc. CVPR Workshops CVPRW '08, 1–8.

Arbelaez, P., 2006. Boundary extraction in natural images using ultrametric contour maps. In IEEE CVPR Workshop, 182–182.

Arbelaez, P., Maire, M., Fowlkes, C., and Malik, J., 2011. Contour detection and hierarchical image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(5):898–916.

Avidan, S. and Shamir, A., 2007. Seam carving for content-aware image resizing. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 26(3). ISSN 0730-0301.

Ballan, L., Brostow, G.J., Puwein, J., and Pollefeys, M., 2010. Unstructured video-based rendering: Interactive exploration of casually captured videos. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 29:87:1–87:11. ISSN 0730-0301.

Banks, M.S., Held, R.T., and Girshick, A.R., 2009. Perception of 3-D layout in stereo displays. *Information Display*, 25(1):12–16.

Barnes, C., Shechtman, E., Finkelstein, A., and Goldman, D.B., 2009. Patchmatch: a randomized correspondence algorithm for structural image editing. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 28(3):24:1–24:11. ISSN 0730-0301.

Berger, K., Lipski, C., Linz, C., Sellent, A., and Magnor, M., 2009. A ghosting artifact detector for interpolated image quality assessment. In Proc. APGV, 128–128. ISBN 978-1-60558-743-1.

Bhat, P., Zitnick, C.L., Snavely, N., Agarwala, A., Agrawala, M., Cohen, M., Curless, B., and Kang, S.B., 2007. Using photographs to enhance videos of a static scene. In Proc. EGSR, 327–338. ISBN 978-3-905673-52-4.

Bousseau, A., Paris, S., and Durand, F., 2009. User-assisted intrinsic images. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)*, 28(5):130:1–130:10. ISSN 0730-0301.

Brooks, B. and Rose, F., 2003. The use of virtual reality in memory rehabilitation: current findings and future directions. *NeuroRehabilitation*, 18(2):147–157.

Buehler, C., Bosse, M., McMillan, L., Gortler, S., and Cohen, M., 2001. Unstructured lumigraph rendering. In Proc. SIGGRAPH, 425–432. ISBN 1-58113-374-X.

Carroll, R., Agarwala, A., and Agrawala, M., 2010. Image warps for artistic perspective manipulation. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 29:127:1–127:9. ISSN 0730-0301.

Carroll, R., Agrawala, M., and Agarwala, A., 2009. Optimizing content-preserving projections for wide-angle images. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 28(3).

Chang, C.H. and Chuang, Y.Y., 2012. A line-structure-preserving approach to image resizing. In Proc. CVPR, 1075–1082. ISSN 1063-6919.

Chapoulie, E., Guerchouche, R., Petit, P.D., Chaurasia, G., Robert, P., and Drettakis, G., 2014. Reminiscence therapy using image-based rendering in VR. In Proc. IEEE Virtual Reality (short paper). To appear.

Chen, J., Bautembach, D., and Izadi, S., 2013. Scalable real-time volumetric surface reconstruction. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 32(4):113:1–113:16. ISSN 0730-0301.

CHEN, J., PARIS, S., WANG, J., MATUSIK, W., COHEN, M., and DURAND, F., 2011. The video mesh: A data structure for image-based three-dimensional video editing. In ICCP, 1–8.

CHEN, S.E. and WILLIAMS, L., 1993. View interpolation for image synthesis. In Proc. SIGGRAPH, 279–288.

CHEN, Y., DAVIS, T.A., HAGER, W.W., and RAJAMANICKAM, S., 2008. Algorithm 887: Cholmod, supernodal sparse cholesky factorization and update/downdate. *ACM Trans. Math. Softw.*, 35(3):22:1–22:14. ISSN 0098-3500.

CIGLA, C., ZABULIS, X., and ALATAN, A., 2007. Region-based dense depth extraction from multi-view video. In ICIP. ISSN 1522-4880.

CIRIO, G., VANGORP, P., CHAPOULIE, E., MARCHAL, M., LECUYER, A., and DRETTAKIS, G., 2012. Walking in a cube: Novel metaphors for safely navigating large virtual environments in restricted real workspaces. *IEEE Transactions on Visualization and Computer Graphics (Proc. IEEE Virtual Reality)*, 18(4):546–554. ISSN 1077-2626.

COOPER, E.A., PIAZZA, E.A., and BANKS, M.S., 2012. The perceptual basis of common photographic practice. *Journal of Vision*, 12(5).

CRIMINISI, A., PÉREZ, P., and TOYAMA, K., 2003. Object removal by exemplar-based inpainting. In Proc. CVPR, 721–728.

CRIMINISI, A., SHARP, T., ROTHER, C., and P'EREZ, P., 2010. Geodesic image and video editing. *ACM Trans. Graph.*, 29(5):134:1–134:15. ISSN 0730-0301.

CRUZ-NEIRA, C., SANDIN, D.J., and DEFANTI, T.A., 1993. Surround-screen projection-based virtual reality: The design and implementation of the cave. In Proc. SIGGRAPH, 135–142. ISBN 0-89791-601-8.

DEBEVEC, P., 1998. Rendering synthetic objects into real scenes: Bridging traditional and image-based graphics with global illumination and high dynamic range photography. In Proc. SIGGRAPH, 189–198. ISBN 0-89791-999-8.

DEBEVEC, P.E., TAYLOR, C.J., and MALIK, J., 1996. Modeling and rendering architecture from photographs: a hybrid geometry- and image-based approach. In Proc. SIGGRAPH, 11–20. ISBN 0-89791-746-4.

DIDYK, P., RITSCHEL, T., EISEMANN, E., MYSZKOWSKI, K., and SEIDEL, H.P., 2010. Adaptive image-space stereo view synthesis. In VMV, 299–306.

Douglas, D. and Peucker, T., 1972. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *The Canadian Cartographer*, 10(2).

Eisemann, M., Decker, B.D., Magnor, M., Bekaert, P., de Aguiar, E., Ahmed, N., Theobalt, C., and Sellent, A., 2008. Floating textures. *Comput. Graph. Forum (Proc. Eurographics)*, 27(2):409–418.

Ernst, M.O. and Banks, M.S., 2002. Humans integrate visual and haptic information in a statistically optimal fashion. *Nature*, 415(6870):429–433.

Felzenszwalb, P.F. and Huttenlocher, D.P., 2004. Efficient graph-based image segmentation. *Int. J. Comput. Vision*, 59:167–181. ISSN 0920-5691.

Fitzgibbon, A.W., Wexler, Y., and Zisserman, A., 2005. Image-based rendering using image-based priors. *Int. J. Comput. Vision*, 63(2):141–151.

Fuhrmann, S. and Goesele, M., 2011. Fusion of depth maps with multiple scales. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)*, 30(6):148:1–148:8. ISSN 0730-0301.

Furukawa, Y., Curless, B., Seitz, S.M., and Szeliski, R., 2009. Manhattan-world stereo. In Proc. CVPR, 1422–1429. ISSN 1063-6919.

Furukawa, Y. and Ponce, J., 2009. Accurate, dense, and robust multi-view stereopsis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(8):1362–1376.

Gal, R., Sorkine, O., and Cohen-Or, D., 2006. Feature-aware texturing. In Proc. EGSR, 297–303. ISBN 3-905673-35-5.

Gallup, D., Frahm, J.M., and Pollefeys, M., 2010. Piecewise planar and non-planar stereo for urban scene reconstruction. In Proc. CVPR, 1418–1425. ISSN 1063-6919.

Goesele, M., Ackermann, J., Fuhrmann, S., Haubold, C., and Klowsky, R., 2010. Ambient point clouds for view interpolation. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 29:95:1–95:6. ISSN 0730-0301.

Goesele, M., Curless, B., and Seitz, S.M., 2006. Multi-view stereo revisited. In Proc. CVPR, 2402–2409. ISBN 0-7695-2597-0.

Goesele, M., Snavely, N., Curless, B., Hoppe, H., and Seitz, S.M., 2007. Multi-view stereo for community photo collections. In Proc. ICCV, 1–8.

Gonneaud, J., Piolino, P., Lecouvey, G., Madeleine, S., Orriols, E., Fleury, P., Eustache, F., and Desgranges, B., 2012. Assessing prospective memory in young healthy adults using virtual reality. In Disability, Virtual Reality, and Associated Technologies, Proc. of the 9th Int. Conf., 211–218.

Gortler, S.J., Grzeszczuk, R., Szeliski, R., and Cohen, M.F., 1996. The Lumigraph. In Proc. SIGGRAPH, 43–54. ISBN 0-89791-746-4.

Grundmann, M., Kwatra, V., Han, M., and Essa, I., 2010. Efficient hierarchical graph based video segmentation. In Proc. CVPR.

Gupta, A., Bhat, P., Dontcheva, M., Curless, B., Deussen, O., and Cohen, M., 2009. Enhancing and experiencing spacetime resolution with videos and stills. In Proc. ICCP.

Hauswiesner, S., Straka, M., and Reitmayr, G., 2011. Coherent image-based rendering of real-world objects. In Symposium on Interactive 3D Graphics and Games, I3D '11, 183–190. ISBN 978-1-4503-0565-5.

He, K., Chang, H., and Sun, J., 2013. Rectangling panoramic images via warping. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 32(4):79:1–79:10. ISSN 0730-0301.

He, X. and Yuille, A., 2010. Occlusion boundary detection using pseudo-depth. In Proc. ECCV, 539–552. ISBN 3-642-15560-X, 978-3-642-15560-4.

Heigl, B., Koch, R., Pollefeys, M., Denzler, J., and Gool, L., 1999. Plenoptic modeling and rendering from image sequences taken by a hand-held camera. In Mustererkennung 1999, Informatik aktuell, 94–101. Springer Berlin Heidelberg. ISBN 978-3-540-66381-2.

Hiep, V.H., Keriven, R., Labatut, P., and Pons, J.P., 2009. Towards high-resolution large-scale multi-view stereo. In Proc. CVPR, 1430–1437. ISSN 1063-6919.

Hoiem, D., Efros, A.A., and Hebert, M., 2007a. Recovering surface layout from an image. *Int. J. Comput. Vision*, 75(1):151–172. ISSN 0920-5691.

Hoiem, D., Stein, A.N., Efros, A.A., and Hebert, M., 2007b. Recovering occlusion boundaries from a single image. *Proc. ICCV*, 1–8.

Hoppe, H., DeRose, T., Duchamp, T., McDonald, J., and Stuetzle, W., 1992. Surface reconstruction from unorganized points. In Proc. SIGGRAPH, 71–78. ISBN 0-89791-479-1.

Hornung, A. and Kobbelt, L., 2009. Interactive pixel-accurate free viewpoint rendering from images with silhouette aware sampling. *Comput. Graph. Forum*, 28(8):2090–2103. ISSN 1467-8659.

Igarashi, T., Moscovich, T., and Hughes, J.F., 2005. As-rigid-as-possible shape manipulation. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 24(3):1134–1141. ISSN 0730-0301.

Kazhdan, M., Bolitho, M., and Hoppe, H., 2006. Poisson surface reconstruction. In Proc. Symposium on Geometry Processing, 61–70. ISBN 30905673-36-3.

Kemelmacher-Shlizerman, I., Shechtman, E., Garg, R., and Seitz, S.M., 2011. Exploring photobios. *ACM Trans. Graph.*, 30(4):61:1–61:10. ISSN 0730-0301.

Kolmogorov, V. and Zabih, R., 2004. What energy functions can be minimized via graph cuts? *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(2):147–159.

Kopf, J., Chen, B., Szeliski, R., and Cohen, M., 2010. Street slide: browsing street level imagery. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 29(4):96:1–96:8. ISSN 0730-0301.

Kopf, J., Langguth, F., Scharstein, D., Szeliski, R., and Goesele, M., 2013. Image-based rendering in the gradient domain. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)*, 32(6):to appear.

Kowdle, A., Sinha, S.N., and Szeliski, R., 2012. Multiple view object cosegmentation using appearance and stereo cues. In Proc. ECCV. ISBN 978-3-642-33714-7.

Krähenbühl, P., Lang, M., Hornung, A., and Gross, M., 2009. A system for retargeting of streaming video. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)*, 28(5):126:1–126:10. ISSN 0730-0301.

Labatut, P., Pons, J.P., and Keriven, R., 2007. Efficient multi-view reconstruction of large-scale scenes using interest points, delaunay triangulation and graph cuts. In Proc. ICCV, 1–8. ISSN 1550-5499.

Laffont, P.Y., Bousseau, A., and Drettakis, G., 2013. Rich intrinsic image decomposition of outdoor scenes from multiple views. *IEEE Transactions on Visualization and Computer Graphics*, 19(2):210–224. ISSN 1077-2626.

Laffont, P.Y., Bousseau, A., Paris, S., Durand, F., and Drettakis, G., 2012. Coherent intrinsic images from photo collections. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)*, 31(6):202.

Levin, D., 1998. The approximation power of moving least-squares. *Mathematics of Computation*, 67(224):1517–1531. ISSN 0025-5718.

Levinshtein, A., Stere, A., Kutulakos, K.N., Fleet, D.J., Dickinson, S.J., and Siddiqi, K., 2009. Turbopixels: Fast superpixels using geometric flows. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(12):2290–2297. ISSN 0162-8828.

Levoy, M. and Hanrahan, P., 1996. Light field rendering. In Proc. SIGGRAPH, 31–42. ISBN 0-89791-746-4.

Lhuillier, M. and Quan, L., 1999. Image interpolation by joint view triangulation. In Proc. CVPR, volume 2.

Lhuillier, M. and Quan, L., 2000. Edge-constrained joint view triangulation for image interpolation. *Proc. CVPR*, 2:2218. ISSN 1063-6919.

Lipski, C., Linz, C., Berger, K., Sellent, A., and Magnor, M., 2010. Virtual video camera: Image-based viewpoint navigation through space and time. *Comput. Graph. Forum*, 29(8):2555–2568.

Liu, F., Gleicher, M., Jin, H., and Agarwala, A., 2009. Content-preserving warps for 3D video stabilization. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 28:44:1–44:9. ISSN 0730-0301.

Liu, S., Yuan, L., Tan, P., and Sun, J., 2013. Bundled camera paths for video stabilization. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 32(4):78:1–78:10. ISSN 0730-0301.

Liverman, M., 2004. The Animator's Motion Capture Guide: Organizing, Managing, and Editing.

Lumsden, E.A., 1983. Perception of radial distance as a function of magnification and truncation of depicted spatial layout. *Attention, Perception, & Psychophysics*, 33(2):177–182.

Mahajan, D., Huang, F.C., Matusik, W., Ramamoorthi, R., and Belhumeur, P., 2009. Moving gradients: a path-based method for plausible image interpolation. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 28:42:1–42:11. ISSN 0730-0301.

Maire, M., Arbelaez, P., Fowlkes, C., and Malik, J., 2008. Using contours to detect and localize junctions in natural images. *Prpc. CVPR*, 1–8.

McMillan, L. and Bishop, G., 1995. Plenoptic modeling: an image-based rendering system. In Proc. SIGGRAPH, 39–46. ISBN 0-89791-701-4.

Mičušík, B. and Košecká, J., 2009. Piecewise planar city 3D modeling from street view panoramic sequences. In Proc. CVPR, 2906–2912. ISBN 978-1-4244-3992-8.

Mičušík, B. and Košecká, J., 2010. Multi-view superpixel stereo in urban environments. *Int. J. Comput. Vision*, 89(1):106–119. ISSN 0920-5691.

Mori, G., 2005. Guiding model search using segmentation. In Proc. ICCV, volume 2, 1417–1423. ISSN 1550-5499.

Morvan, Y. and O'Sullivan, C., 2009a. Handling occluders in transitions from panoramic images: A perceptual study. *ACM Trans. Appl. Percept.*, 6:25:1–25:15. ISSN 1544-3558.

Morvan, Y. and O'Sullivan, C., 2009b. A perceptual approach to trimming and tuning unstructured lumigraphs. *ACM Trans. Appl. Percept.*, 5:19:1–19:24. ISSN 1544-3558.

Mustafa, M., Guthe, S., and Magnor, M., 2012a. Single-trial eeg classification of artifacts in videos. *ACM Trans. Appl. Percept.*, 9(3):12:1–12:15. ISSN 1544-3558.

Mustafa, M., Lindemann, L., and Magnor, M., 2012b. Eeg analysis of implicit human visual perception. In Proc. SIGCHI, 513–516. ISBN 978-1-4503-1015-4.

Niessner, M., Zollhöfer, M., Izadi, S., and Stamminger, M., 2013. Real-time 3D reconstruction at scale using voxel hashing. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)*, 32(6):169:1–169:11. ISSN 0730-0301.

Panozzo, D., Weber, O., and Sorkine, O., 2012. Robust image retargeting via axis-aligned deformation. *Comp. Graph. Forum*, 31(2pt1):229–236. ISSN 0167-7055.

Pérez, P., Gangnet, M., and Blake, A., 2003. Poisson image editing. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 22(3):313–318. ISSN 0730-0301.

Pollefeys, M., Nistér, D., Frahm, J.M., Akbarzadeh, A., Mordohai, P., Clipp, B., Engels, C., Gallup, D., Kim, S.J., Merrell, P., Salmi, C., Sinha, S., Talton, B., Wang, L., Yang, Q., Stewénius, H., Yang, R., Welch, G., and Towles, H., 2008. Detailed real-time urban 3D reconstruction from video. *Int. J. Comput. Vision*, 78(2-3):143–167. ISSN 0920-5691.

Ren, X. and Malik, J., 2003. Learning a classification model for segmentation. In Proc. ICCV, 10–17 vol.1.

Revelle, W., 2008. psych: Procedures for Psychological, Psychometric, and Personality Research. R package version 1.0-42+.

Rineau, L., 2010. 2D conforming triangulations and meshes. In CGAL User and Ref. Manual. CGAL Ed. Bd., 3.7 edition.

Rosinski, R.R., Mulholland, T., Degelman, D., and Farber, J., 1980. Picture perception: An analysis of visual compensation. *Attention, Perception, & Psychophysics*, 28(6):521–526.

Rubinstein, M., Gutierrez, D., Sorkine, O., and Shamir, A., 2010. A comparative study of image retargeting. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)*, 29(6):160:1–160:10. ISSN 0730-0301.

Schaefer, S., McPhail, T., and Warren, J., 2006. Image deformation using moving least squares. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 25(3):533–540. ISSN 0730-0301.

Schirmacher, H., Heidrich, W., and Seidel, H.P., 2000. High-quality interactive lumigraph rendering through warping. In Proc. Graphics Interface, 87–94. ISBN 0-9695338-9-6.

Schwarz, M. and Stamminger, M., 2009. On predicting visual popping in dynamic scenes. In Proc. APGV, 93–100. ISBN 978-1-60558-743-1.

Sedgwick, H.A., 1991. The effects of viewpoint on the virtual space of pictures. In S.R. Ellis, editor, Pictorial Communication in Virtual and Real Environments, 460–479. Taylor & Francis.

Seitz, S.M., Curless, B., Diebel, J., Scharstein, D., and Szeliski, R., 2006. A comparison and evaluation of multi-view stereo reconstruction algorithms. In Proc. CVPR. ISSN 1063-6919.

Seitz, S.M. and Dyer, C.R., 1996. View morphing. In Proc. SIGGRAPH, 21–30.

Shi, J. and Malik, J., 1997. Normalized cuts and image segmentation. In Proc. CVPR, 731. ISSN 1063-6919.

Shih, Y., Paris, S., Durand, F., and Freeman, W.T., 2013. Data-driven hallucination of different times of day from a single outdoor photo. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)*, 32(6):200:1–200:11. ISSN 0730-0301.

Shum, H. and Kang, S., 2000. A review of image-based rendering techniques. In VCIP00, 2–13.

Shum, H.Y., Chan, S.C., and Kang, S.B., 2006. Image-based rendering, volume 2. Springer.

Sillion, F., Drettakis, G., and Bodelet, B., 1997. Efficient impostor manipulation for real-time visualization of urban scenery. *Comput. Graph. Forum (Proc. Eurographics)*, 16:207–218.

Sinha, S.N., Kopf, J., Goesele, M., Scharstein, D., and Szeliski, R., 2012. Image-based rendering for scenes with reflections. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 31(4):100:1–100:10. ISSN 0730-0301.

Sinha, S.N., Mordohai, P., and Pollefeys, M., 2007. Multi-view stereo via graph cuts on the dual of an adaptive tetrahedral mesh. In Proc. ICCV, 1–8.

Sinha, S.N., Steedly, D., and Szeliski, R., 2009. Piecewise planar stereo for image-based rendering. In Proc. ICCV, 1881–1888.

Smith, P.C. and Smith, O.W., 1961. Ball throwing responses to photographically portrayed targets. *Journal of Experimental Psychology*, 62(3):223.

Snavely, N., Seitz, S.M., and Szeliski, R., 2006. Photo tourism: exploring photo collections in 3D. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 25(3):835–846. ISSN 0730-0301.

STEIN, A.N. and HEBERT, M., 2009. Occlusion boundaries from motion: Low-level detection and mid-level reasoning. *Int. J. Comput. Vision*, 82:325–357. ISSN 0920-5691.

STICH, T., LINZ, C., WALLRAVEN, C., CUNNINGHAM, D., and MAGNOR, M., 2011. Perception-motivated interpolation of image sequences. *ACM Trans. Appl. Percept.*, 8(2):11:1–11:25. ISSN 1544-3558.

STRECHA, C., VON HANSEN, W., GOOL, L.J.V., FUA, P., and THOENNESSEN, U., 2008. On benchmarking camera calibration and multi-view stereo for high resolution imagery. In Proc. CVPR.

TEH, C.H. and CHIN, R., 1989. On the detection of dominant points on digital curves. *IEEE Trans.Pattern Anal. Mach. Intell.*, 11(8). ISSN 0162-8828.

TODOROVIĆ, D., 2009. The effect of the observer vantage point on perceived distortions in linear perspective images. *Attention, Perception, & Psychophysics*, 71(1):183–193.

TOLEDO, S., 2003. TAUCS: A Library of Sparse Linear Solvers, version 2.2. Tel-Aviv University.

TOMPKIN, J., KIM, K.I., KAUTZ, J., and THEOBALT, C., 2012. Videoscapes: Exploring sparse, unstructured video collections. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 31(4):68:1–68:12. ISSN 0730-0301.

TOMPKIN, J., KIM, M.H., KIM, K.I., KAUTZ, J., and THEOBALT, C., 2013. Preference and artifact analysis for video transitions of places. *ACM Trans. Appl. Percept.*, 10(3):13:1–13:19. ISSN 1544-3558.

VANGORP, P., CHAURASIA, G., LAFFONT, P.Y., FLEMING, R.W., and DRETTAKIS, G., 2011. Perception of visual artifacts in image-based rendering of façades. *Comput. Graph. Forum (Proc. EGSR)*, 30(4):1241–1250.

VANGORP, P., RICHARDT, C., COOPER, E.A., CHAURASIA, G., BANKS, M.S., and DRETTAKIS, G., 2013. Perception of perspective distortions in image-based rendering. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 32(4):58:1–58:12. ISSN 0730-0301.

VEDALDI, A. and SOATTO, S., 2008. Quick shift and kernel methods for mode seeking. In D.A. Forsyth, P.H.S. Torr, and A. Zisserman, editors, Proc. ECCV, 705–718. ISBN 978-3-540-88692-1.

VISHWANATH, D., GIRSHICK, A.R., and BANKS, M.S., 2005. Why pictures look right when viewed from the wrong place. *Nature Neuroscience*, 8(10):1401–1410.

WANG, Y.S., LIN, H.C., SORKINE, O., and LEE, T.Y., 2010. Motion-based video retargeting with optimized crop-and-warp. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 29(4):90:1–90:9. ISSN 0730-0301.

WANG, Y.S., TAI, C.L., SORKINE, O., and LEE, T.Y., 2008. Optimized scale-and-stretch for image resizing. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)*, 27(5):118:1–118:8. ISSN 0730-0301.

Wolf, L., Guttmann, M., and Cohen-Or, D., 2007. Non-homogeneous content-driven video-retargeting. In Proc. ICCV, 1–6. ISSN 1550-5499.

Xiao, J., 2012. A 2D + 3D Rich Data Approach to Scene Understanding. Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA.

Yang, Q., Yang, R., Davis, J., and Nistér, D., 2007. Spatial-depth super resolution for range images. In Proc. CVPR.

Zhang, G.X., Cheng, M.M., Hu, S.M., and Martin, R.R., 2009. A shape-preserving approach to image resizing. *Comput. Graph. Forum*, 28(7):1897–1906. ISSN 1467-8659.

Zhou, Q.Y., Miller, S., and Koltun, V., 2013. Elastic fragments for dense scene reconstruction. In Proc. ICCV.

Zitnick, C.L. and Kang, S.B., 2007. Stereo for image-based rendering using image over-segmentation. *Int. J. Comput. Vision*, 75(1):49–65. ISSN 0920-5691.

Zitnick, C.L., Kang, S.B., Uyttendaele, M., Winder, S., and Szeliski, R., 2004. High-quality video view interpolation using a layered representation. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 23(3):600–608. ISSN 0730-0301.