Modified Belief Propagation for Reconstruction of Office Environments

E. Scott Larsen

A dissertation submitted to the faculty of the University of North Carolina at Chapel Hill in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Computer Science.

Chapel Hill
2008

Abstract

E. Scott Larsen: Modified Belief Propagation for Reconstruction of Office Environments.
(Under the direction of Dr. Henry Fuchs.)

Belief Propagation (BP) is an algorithm that has found broad application in many areas of computer science. The range of these areas includes Error Correcting Codes, Kalman filters, particle filters, and – most relevantly – stereo computer vision. Many of the currently best algorithms for stereo vision benchmarks, *e.g.* the Middlebury dataset, use Belief Propagation. This dissertation describes improvements to the core algorithm to improve its applicability and usefulness for computer vision applications.

A Belief Propagation solution to a computer vision problem is commonly based on specification of a Markov Random Field that it optimizes. Both Markov Random Fields and Belief Propagation have at their core some definition of nodes and "neighborhoods" for each node. Each node has a subset of the other nodes defined to be its neighborhood. In common usages for stereo computer vision, the neighborhoods are defined as a pixel's immediate four spatial neighbors. For any given node, this neighborhood definition may or may not be correct for the specific scene. In a setting with video cameras, I expand the neighborhood definition to include corresponding nodes in temporal neighborhoods in addition to spatial neighborhoods. This amplifies the problem of erroneous neighborhood assignments. Part of this dissertation addresses the erroneous neighborhood assignment problem.

Often, no single algorithm is always the best. The Markov Random Field formulation appears amiable to integration of other algorithms: I explore that potential here by integrating priors from independent algorithms.

This dissertation makes core improvements to BP such that it is more robust to erroneous neighborhood assignments, is more robust in regions with inputs that are near-uniform, and can be biased in a sensitive manner towards higher level priors.

These core improvements are demonstrated by the presented results: application to office environments, real-world datasets, and benchmark datasets.

This work is dedicated to my wonderful wife Amy, our children who let it borrow their dad, and He who made it possible.

Acknowledgments

Thanks to my adviser, Henry Fuchs, for his patience and thought-provoking discussions; Herman Towles for his wisdom and experience, and infinite patience; Marc Pollefeys for his key insights; Philippos Mordohai for all his help, tons of it; Anselmo Lastra and Leonard McMillan for their experienced patience and guidance. Also my thanks to Janet Jones, Tammy Pike, Mike Stone, Fred Jordan, and the rest of their groups who make it all happen without getting their due share of glory. Thanks to Tim Quigg, who ranks higher in this list than his listing. Many thanks to my wife and children, and the rest of my family for their encouragement, patience, and prayers. And thanks be to God, to whom I owe this work and all else.

Table of Contents

List of Figures

Chapter 1

Introduction

Many years ago, so the legend goes, a fine distinguished professor of computer science gave his graduate student two photographs of a single static scene from slightly different viewpoints. He suggested that the student could probably, by the end of the summer, come up with a computer algorithm to build a 3D model of the scene in the photographs. This was thought to be a straightforward problem that was an appropriate difficulty for a summer project. Now, many years later, this still unsolved problem has grown into the field of computer vision, retains many unsolved problems, engages many thousands of researchers, is documented by numerous journals and conferences, and remains a very vigorous research areas in computer science.

Figure 1.1 shows a classic pair of images illustrating the problem. Figure 1.2 contains one common representation of state of the art results (Sun et al., 2003; Klaus et al., 2006). As can be seen, significant progress has been made by many researchers over the decades. Chapter 2 details some of the most common challenges of this problem area, and chronicles some important historical milestones in the progression of the state of the art through the years.

There are seminal contributors in this area. As in many areas, the work becomes regarded as seminal only years later in some cases. The goal of this dissertation is to take a step significantly forward in the field. This work would not be possible without

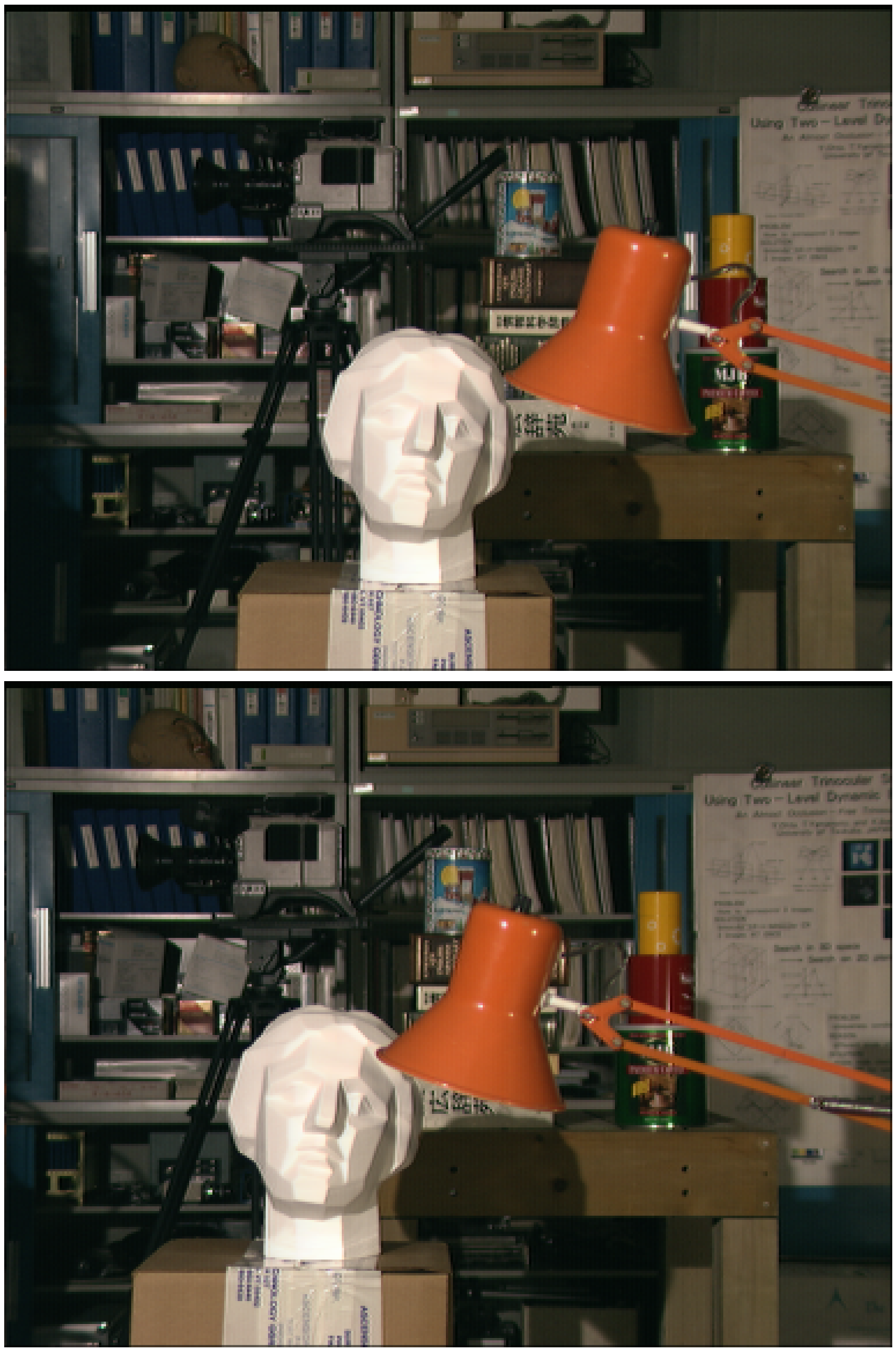


Figure 1.1: A classic pair of images of the same scene (Nakamura et al., 1996), from slightly different viewpoints.

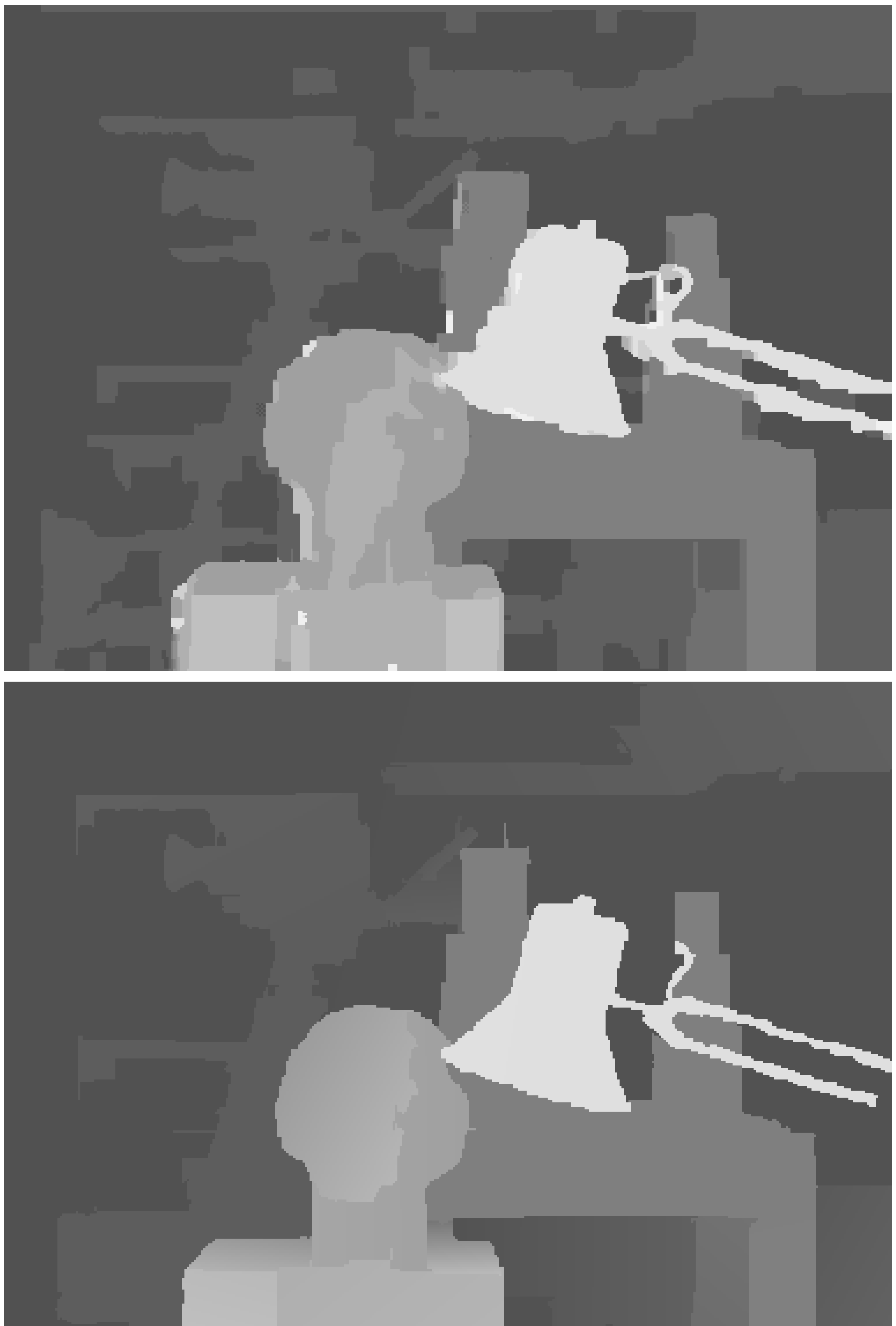


Figure 1.2: Two "depth maps:" images with the color value at each pixel encoding that pixel's depth – distance from the camera to the object in the scene. The top is from (Sun et al., 2003) in 2003, while the bottom is from (Klaus et al., 2006) in 2006.

the “standing on the shoulders of giants” which makes it all possible. This dissertation presents my results, discusses them, and aims at producing a solid foundation upon which the next researchers can stand even a little higher.

Section 1.2 of this chapter details the specific application domain that I am targeting. Section 1.1, further detailed in Chapter 2, briefly discusses computer vision. I have chosen as a foundation for my research the methods of Belief Propagation (Pearl, 1998). Details, including my reasons for this decision, are overviewed in Section 1.3 and detailed in Chapter 3. A summary statement of the aims of this research is presented in Section 1.4. Section 1.2.4 presents the datasets that I, and many others, use for experimentation in attempts to measure the effectiveness of research in these areas.

1.1 Introduction to Computer Vision

This section introduces computer vision, Chapter 2 provides fuller discussion and details.

Computer vision as a field has grown considerably in the last few decades. The sub-field of focus here is the extraction/estimation of depth for every pixel in an image. “Depth” here refers to the distance between the camera’s focal center and the small patch of physical surface corresponding to the pixel in the image. Figure 1.3 illustrates. In the figure, the camera’s focal center is at location C. There is a surface in the scene with a small patch on it labeled P. The camera images this patch at pixel x. The distance from C to P is the desired *“depth”*. As we only have information from the image and the camera, the true location of P is unknown. The objective is to reconstruct all imaged surfaces in the scene. An estimation of the depth for each pixel is a representation of the imaged surfaces, as illustrated in 2D in Figure 1.5. Thus, the objective is to obtain good depth estimates for every pixel in the image taken by the camera.

There are two aids I will use: additional information in “space” and additional information in “time”. The first is very commonly used, the second is less common because

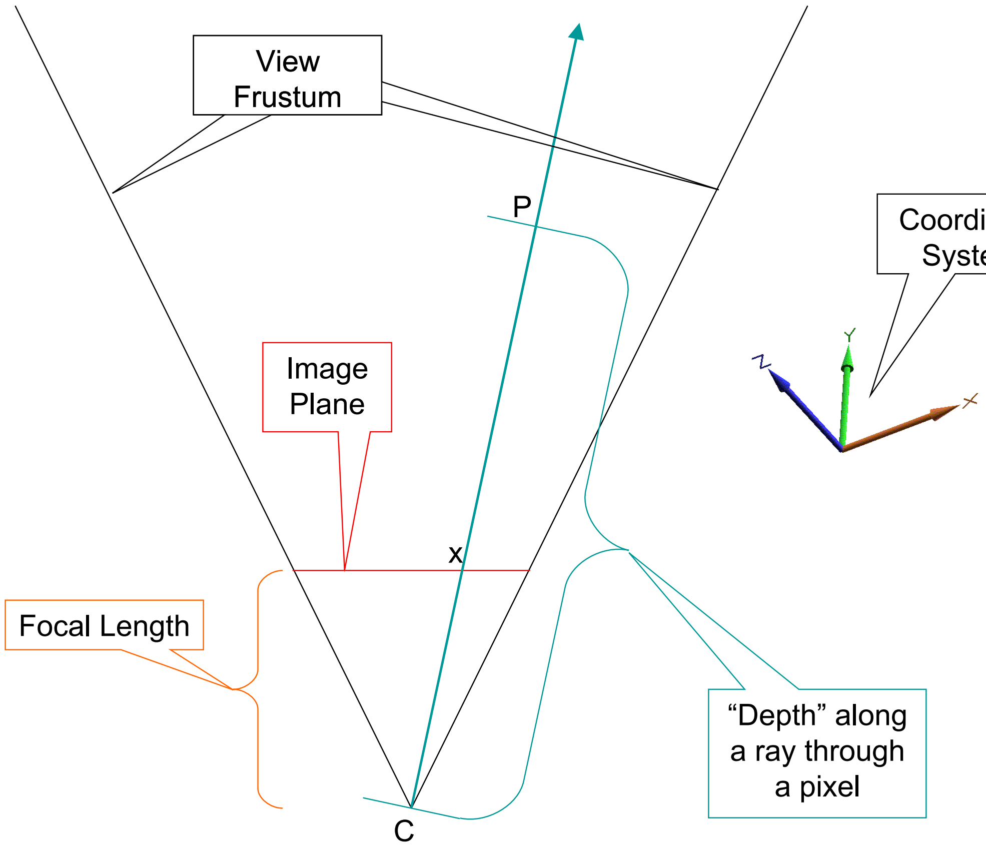


Figure 1.3: Diagram of the essential camera aspects and interpretation of the "depth" value for a single pixel.

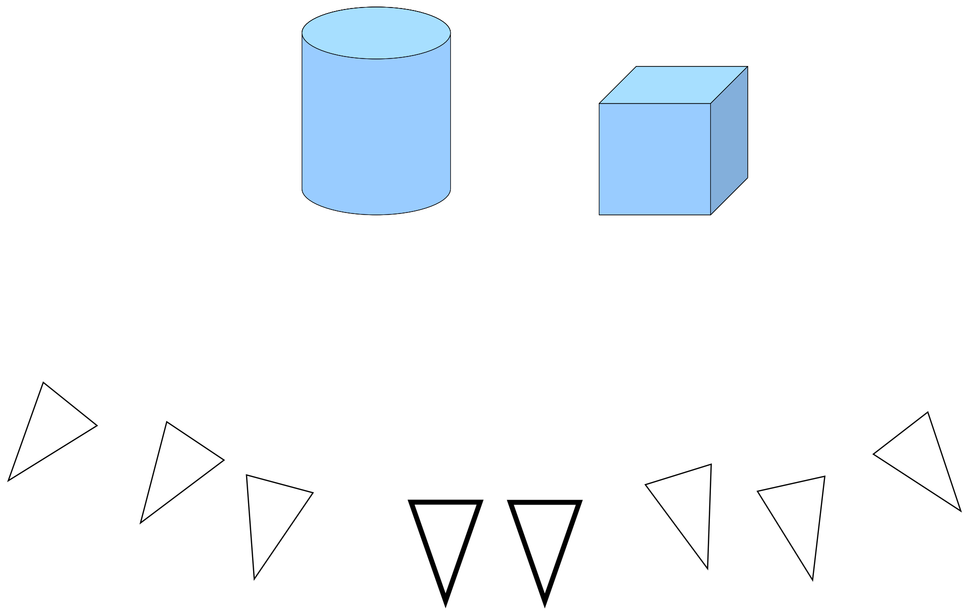


Figure 1.4: Illustration of camera positions and orientations for more than one camera. The two in the center, highlighted, are arranged in a common stereo orientation: being only a horizontal shift and having parallel image planes. The other illustrate a more general position setting with more than two cameras.

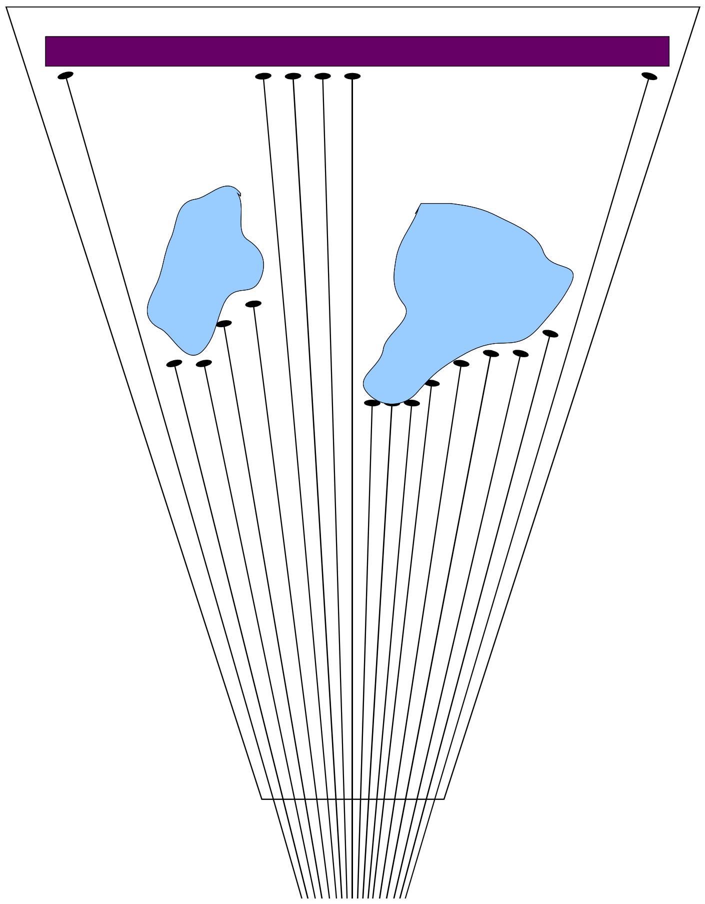


Figure 1.5: Depths for each pixel on a single horizontal line of an image. Observe that these represent a sampling of the scene's surfaces.

it is more difficult. The first aid I use is images taken from additional cameras, as illustrated in Figure 1.4. The use of one additional camera (total of two) is commonly referred to as “stereo” computer vision and more specifically, “binocular stereo.” The use of more than one additional camera is referred to as “multi-view stereo.” Early computer vision was focused on binocular stereo, and the field has expanded to incorporate additional images as such datasets became more available. Many binocular algorithms do not trivially extend to the multi-view setting, and many multi-view algorithms do not trivially extend to binocular settings, though the underlying difficulties are predominately the same. The differences tend to be in scalability of the algorithms: complex algorithms with two images do not always scale to more images, and algorithms taking advantage of many images do not always scale to settings with only two available images.

The second aid I will use is the conversion of the cameras from still-frame images to video sequences. This brings additional challenges due to moving surfaces, but it also provides significant additional information, *e.g.* when the algorithm has two equally likely candidate depths for a surface at one moment in time, it can potentially resolve the ambiguity by observing which depths the surface is estimated to be at in temporally neighboring frames.

1.2 Motivation

I would like to engage for a moment in a little rendering.... Imagine for a moment a trauma patient who shows up in the emergency room. The attending physician recognizes that a particular procedure must be done, and that it must be done within the next few moments. This particular procedure is one that this physician can do, but he has not done recently. The good news is that this physician has a very good friend who performs this procedure quite often and is an expert on this procedure. The bad news is that this expert is across the continent having dinner with his wife. Now, imagine a situation such that the attending physician can wheel a special light fixture over the operating table. This light fixture has the usual lights and is augmented with an array of cameras surrounding the lights. These video cameras process the scene and build a real-time 3D model and representation of all that is in their view frustums. This data is quickly transported to the expert, who steps out of the restaurant for a moment and uses special viewing software on his PDA. This software enables him to treat his PDA as a little window into the surgery room. Additionally, as he moves the PDA with his hand, the image changes in such a way as to enable him to see into the surgery room as if he were actually there, looking inwards through a little window. With this wonderful setup, the attending physician can perform this urgent procedure with more confidence, because he is backed up by his guide, in real-time: an expert who is as if he were really present.

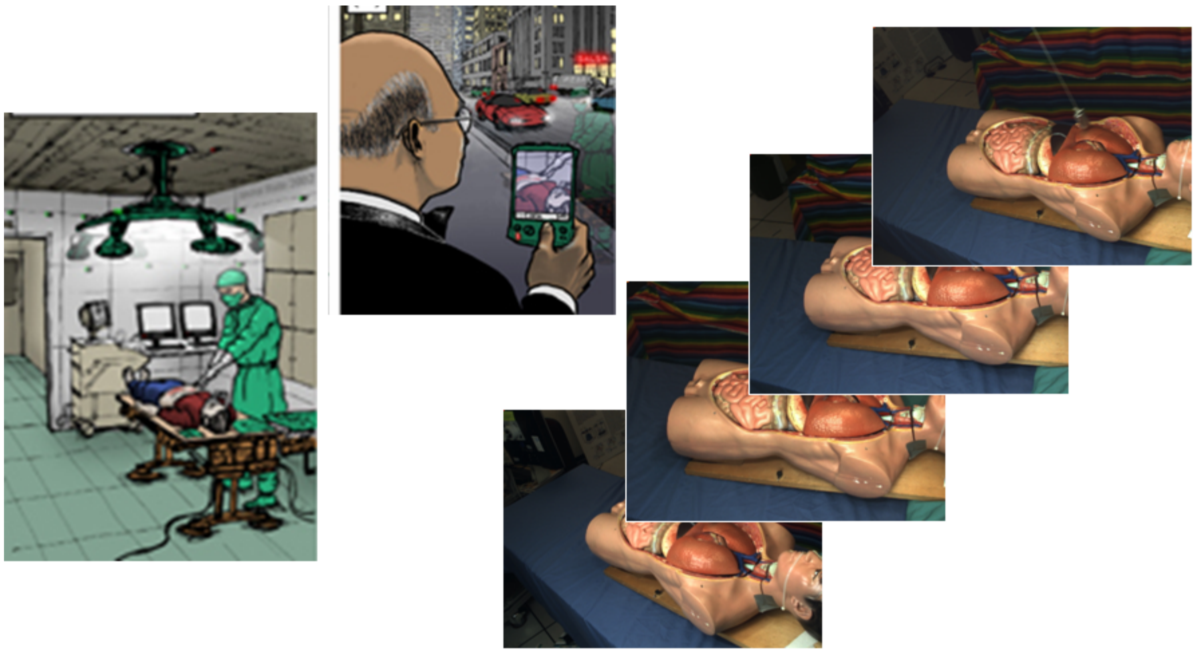


Figure 1.6: An illustration of the 3DMC (Sonnenwald et al., 2006) project. Cameras capture the images (of a static model in this simplified example) and allow for the expert (above) to provide consultation to the attending physician (below).

Admittedly, the rendering above is a little futuristic. Yet, it is definitely a worthwhile goal. It is a goal that, once accomplished by better and better technology, will make a very positive difference in people's lives. I hope this goal is not too far away. Active research and strivings toward this goal is ongoing. This objective is addressed by the work of (Welch et al., 2007; Sonnenwald et al., 2006) The "3D Medical Consultation" project is illustrated in Figure 1.6.

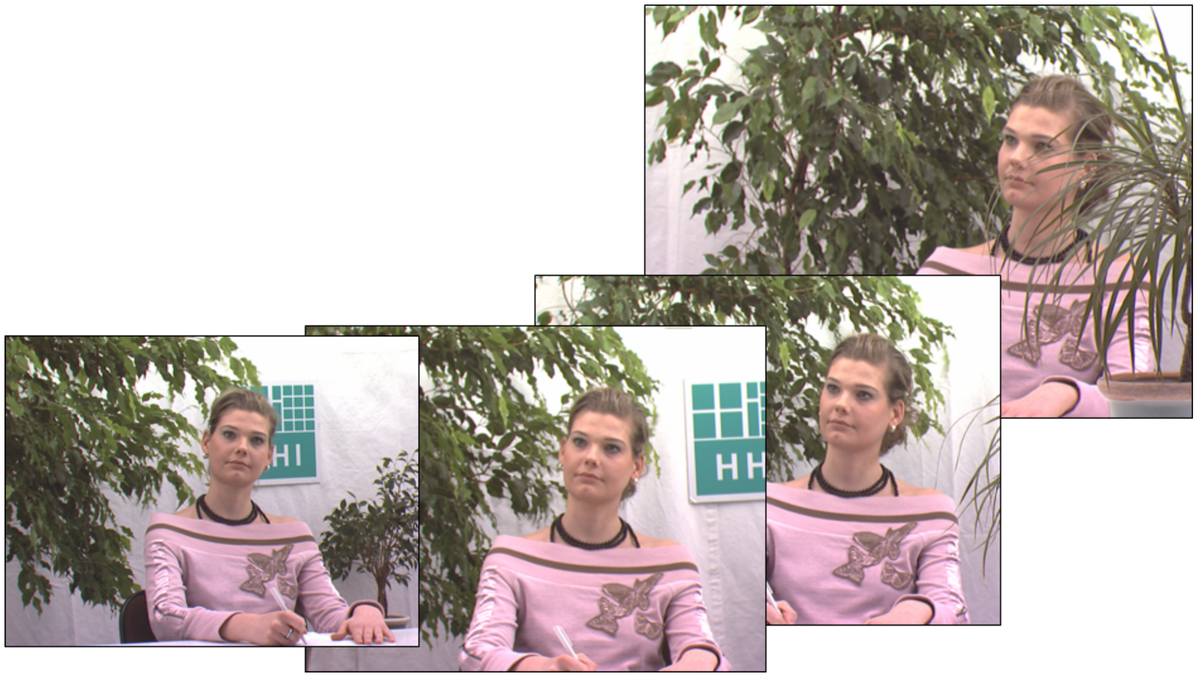


Figure 1.7: Sample images from the 3DTV project. The focus here is a multi-camera broadcasting center, with research emphasis on transport of the multiple video streams and nice rendering of the videos for end users. These images tend to be less synchronized than many other datasets and also suffer from more color calibration issues - which are both cases more typical of this scenario. Further, common focus in this area is real-time processing of the video streams.

There are other applications, some apparently quite different, which require solutions to problems that are closely related to the problems needing to be solved for the medical consultation scenario above. These include 3DTV and multi-camera video transmission (Webpage: 3DTV, 2007), as illustrated in Figure 1.7. Another application is so-called "multi-view video" in which a video recording is played, yet the viewer has the ability to control the camera's position as the video is playing. Figure 1.8 illustrates a sample application of this as pursued by (Zitnick et al., 2004).

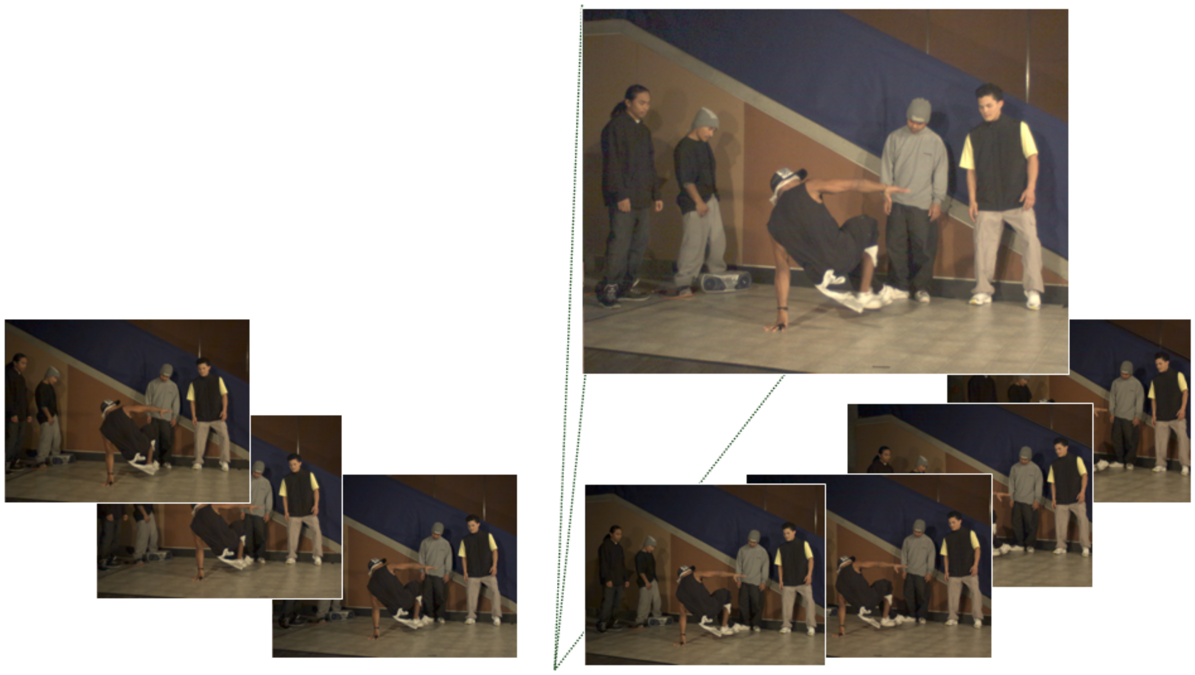


Figure 1.8: A sampling of images from the work of (Zitnick et al., 2004), focusing on the presentation of multiple video streams. The focus of this work is less on the actual depth extraction and more on the usage of the information for higher quality viewing by an observer of the video, observing at novel viewing positions.

The challenges all of these applications face are outlined in the next subsections, with a summary in Section 1.2.3. Section 1.2.4 gives a brief introduction to the specific datasets used in this dissertation, each derived from current work which pursues the objectives of these driving applications.

1.2.1 Still Scenes

As further explored in Chapter 2, much of the research in reconstruction has focused on "static" scenes. This is accomplished either by requiring nothing in the scene to move, or by requiring all the cameras to take their images at exactly the same instant – which can be a technical challenge in some cases. The benefit of these requirements is that they imply that all the cameras image every point in the scene consistently. The static scene case is a simplification that greatly aids many algorithms.

1.2.2 Dynamic Scenes

No current algorithm will consistently be right for all of the pixels all of the time. A consequence of this is that some scene surface will be assigned one depth in one frame and be assigned a different depth in the next frame, when the actual surface has not moved. For many applications this effect is devastating to the usefulness of the results. Consider the medical consultation example given earlier in this section: if some portion of the patient's body appears to be jumping back and forth between two depth values, it may be because their pulse is strong there, or it may be because the algorithm is oscillating between two hypothesis. It can be vitally important for the physician to be able to differentiate between these two causes. This problem does not evidence itself for applications of static scenes. Compared to static scenes, there are relatively few algorithms which explicitly attempt to address temporal inconsistency and dynamic scenes.

The primary theoretical problem in stereo, as in all reconstruction, is ambiguity: there are often multiple depth hypothesis that are as good as the correct one. There are many reasons for this, detailed in Chapter 2. When video is used, portions of the scene which are "the same" from one frame to the next are usually *not* exactly the same to the camera sensors. Part of this comes from the fact that much of the sensor noise is usually different from frame to frame. Part of this is a function of change in portions of the scene – e.g. someone moves and hence their shadow moves, changes the lighting effects on a portion of the floor, though the floor has not changed at all. These are just two examples of things that stay "the same" even though they appear to change in video. A primary theme in this dissertation is the notion that the ambiguity can be reduced as more data is acquired. This can reduce the number of valid hypothesis and aid the program in finding, for more of the scene, the correct hypothesis.

1.2.3 Application Demands

The application domains sketched in this section demand satisfactory results as measured by these metrics:

Temporal Consistency It can be vitally important for the reconstructed surfaces to change if and only if the scene surfaces change correspondingly, *i.e.* if the scene does not change, then the reconstruction results should not change either.

Geometric Accuracy In particular, a measurement on the reconstruction needs to correlate with some metric in the scene, *e.g.* the physical location of the surface should be directly and correctly extractable from the reconstruction.

Wide Range of Viewing Positions These applications may require quality results even when the viewer's position is outside the range of the cameras, *e.g.* if the physician wants to look at the patient from the side, but all the cameras are located in the front of the patient, we need to still give correct results to the physician - given all the information available in the cameras. This is referred to as extrapolation of camera viewpoints.

Thus, it is via these metrics that I measure my results in Chapter 7.

1.2.4 Case Examples

Unfortunately, there are not a large number of publicly available datasets useful for this work: most related research over the decades has been targeted at still scenes, and few of those that target moving scenes have made their data available, with even fewer providing results for comparison. Here, I introduce the datasets I use.

Middlebury

(Scharstein and Szeliski, 2002) provides the so-called "Middlebury" datasets which have become the *de facto* benchmark for still frame stereo algorithms - available from (Webpage: Stereo, 2007). Many modern algorithms target still frame dual camera settings and therefore this dataset provides comparisons with published results of many algorithms. Even though two camera single-scene depth extraction is *not* the focus of this dissertation, I provide compelling comparisons of my algorithms' results to others for this dataset, in Chapter 7.

This dataset, shown in Figure 1.9, consists of a set of carefully constructed scenes. Details of the setup, camera arrangements, etc. are in Chapter 7 and further detailed in (Scharstein and Szeliski, 2002). This dataset consists of rectified binocular pairs (explained in Chapter 2).

Breakdancer

(Zitnick et al., 2004) provided a dataset that is very useful for working towards our target domain: eight video cameras arranged in an arc. This dataset was carefully constructed such that it would be amiable to segmentation-based reconstruction approaches (see Chapter 2). There is a high dynamic range of signal-to-noise ratios across the scene. There are portions of the scene which are static, portions slowly moving, and portions which move very fast. Example images are in Figure 1.10. There are no published results on this dataset which target geometrically correct reconstructions as we target, but there are results in (Zitnick et al., 2004) which target pleasing novel view renderings. Those results therefore have different aims than our work, but a few comparisons are still made - seen in Chapter 7.

3DTV

A European consortium: "3DTV Network of Excellence," at (Webpage: 3DTV, 2007), has produced a few datasets which I use. Samples of the Janine dataset are in Figure 1.11. This dataset was intentionally constructed to be difficult for traditional stereo algorithms. It contains many difficult components such as complex occlusions, significant

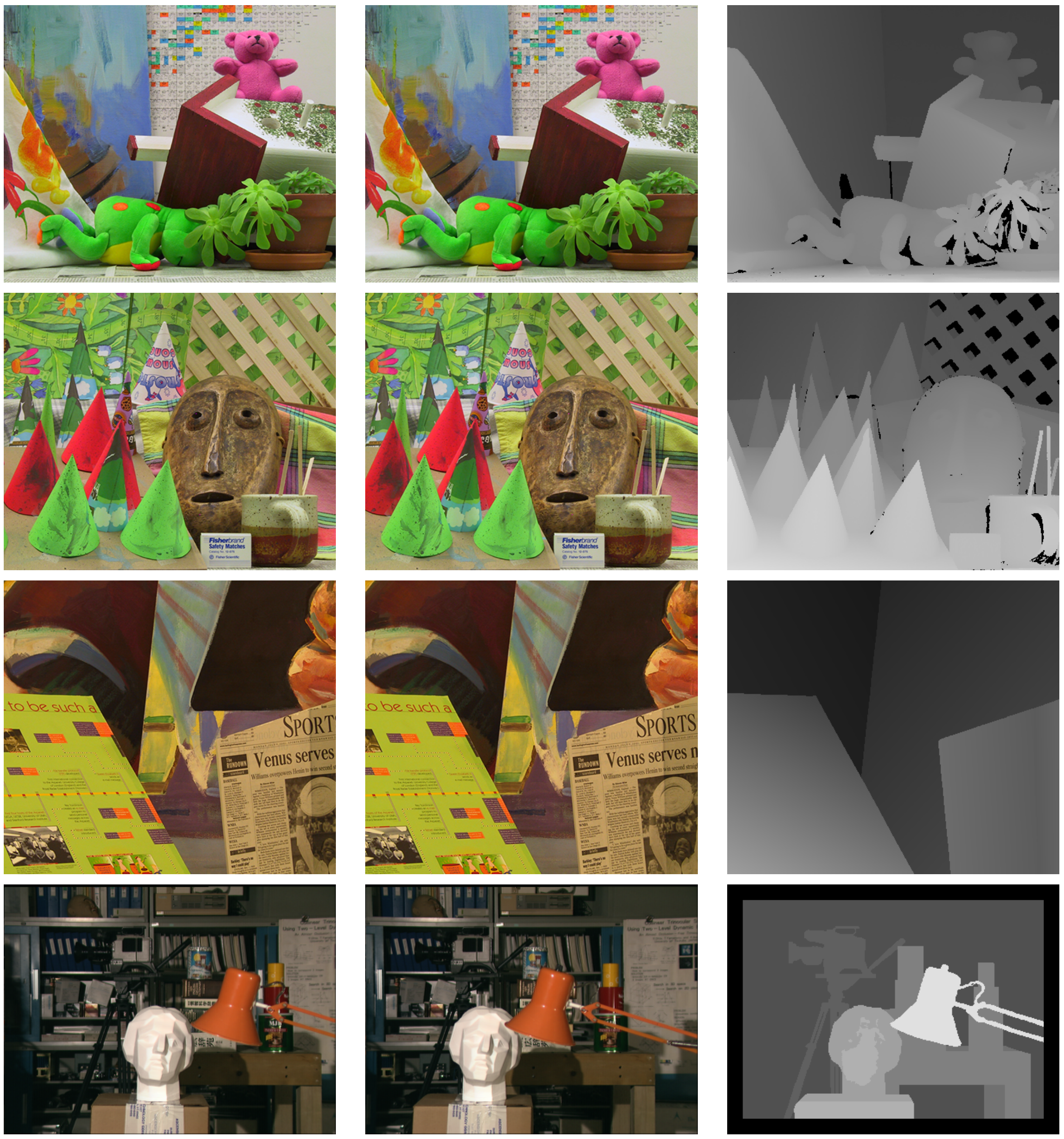


Figure 1.9: Images from the Middlebury Stereo Evaluation webpage (Scharstein and Szeliski, 2002). These are binocular, rectified images of carefully constructed scenes. True depth values are known for each pixel, providing quantitative metrics for comparisons between stereo algorithms.

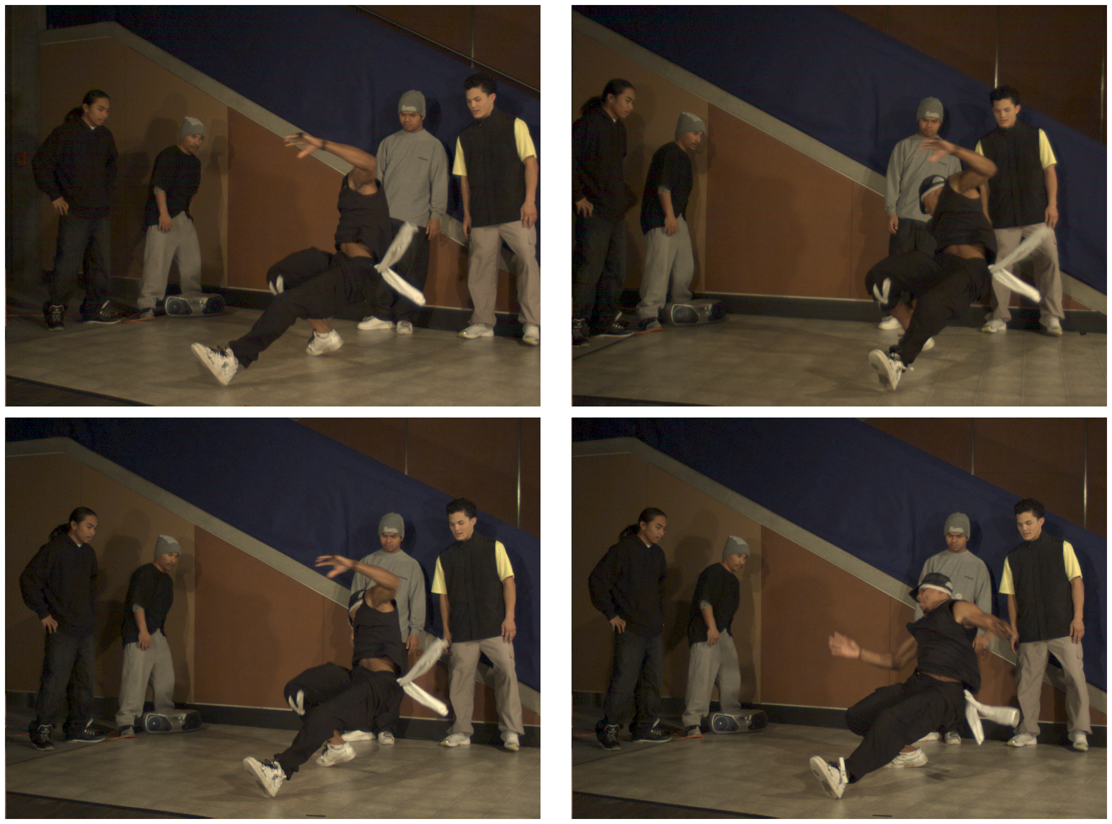


Figure 1.10: Images from the breakdancer dataset from (Zitnick et al., 2004). The top row shows images from the two extremal cameras on the arc. The bottom row is images from a central camera at two consecutive frames, showing dramatic motion from frame to frame.

color inconsistencies between cameras, and large uniform regions. My results on this dataset are detailed in Chapter 7.

Others

There are a handful of other datasets, of which the Multi-View Stereo Evaluation webpage, at (Webpage: Multi-View Stereo, 2007), is representative. These I did not use for the following reasons, visible in the representative images in Figure 1.12:

Static Scenes These datasets are entirely of static scenes. Therefore they are useless for the primary focus of this dissertation: using temporal data to improve reconstruction quality.

Circumspecting Cameras These datasets use cameras which entirely surround a single object and focus on reconstruction of only that object. My research has been focused entirely on reconstructing all pixels in an image. So my approaches have focused on datasets in which the majority of the pixels in a single image are visible in most of the other images also.

Single Central Object These datasets are focused on a central single object, and no attempt is made to reconstruct any other part of the scene other than this object. This allows for volume-based algorithms to be used, which are not applicable to the scenes I am targeting – due partly to memory and precision tradeoffs making good solutions intractable on commodity machines at this time.

1.3 Belief Propagation

This section introduces Belief Propagation‘ at a high level, providing enough context for the rest of this chapter. Fuller discussion and details about Belief Propagation are found in Chapter 3.

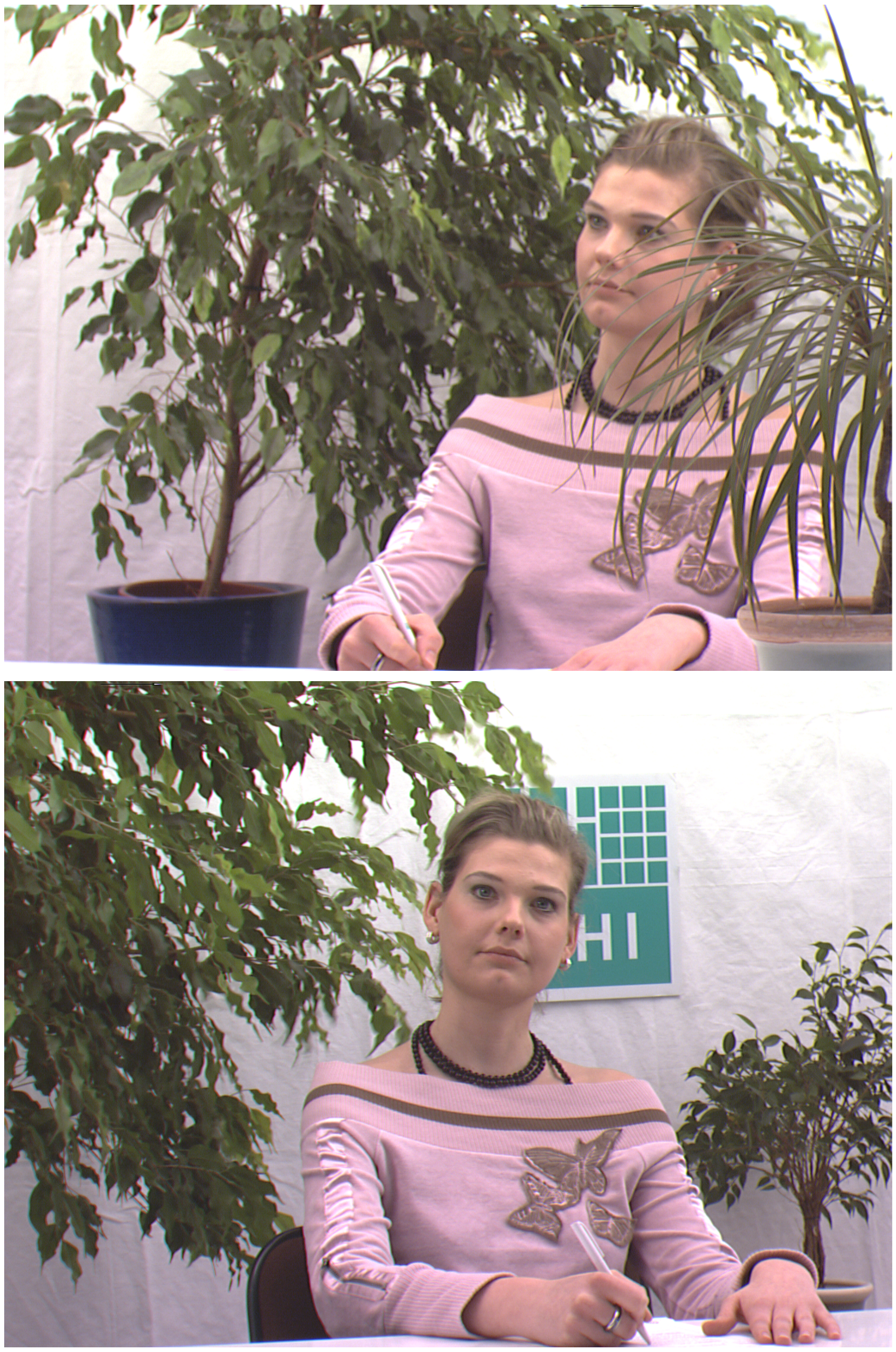


Figure 1.11: A few sample images from the Janine dataset provided by the 3DTV consortium at (Webpage: 3DTV, 2007).

Figure 1.12: Sample images from the Multi-View Stereo Evaluation webpage, at (Webpage: Multi-View Stereo, 2007). This dataset (and a few others like it) was not used for my research for a variety of reasons, detailed in Section 1.2.4.

Belief Propagation originated with the work of (Pearl, 1998) with its first significant application being Turbo Codes (McEliece et al., 1998), in the domain of error correcting codes (Caire et al., 2003). Belief Propagation is known as an *"inference engine"*: given some information, it attempts to infer some unknown information. It is used in computer vision to estimate unknown depth values given the images. Tremendous advances in quality of results for binocular stereo have been made by Belief Propagation algorithms.

Belief Propagation is an iterative algorithm. It works on a graph by updating estimates stored at each node in the graph. These estimates are updated in each iteration by incorporating some of the estimates stored at neighboring nodes (those immediately connected to it in the graph). The estimates are all represented as probability distributions over some set of candidate states.

For binocular stereo, each pixel in an image corresponds to a node in the graph, and each node is connected to the nodes corresponding to its four neighboring pixels: up, down, left, and right. For each node, there is some set of possible depth values, each of which is a candidate state in the Belief Propagation algorithm. Figure 1.13 illustrates.

One might suspect that extending these node connections into the previous and next frames of a video sequence could be beneficial. In this dissertation, this approach is successfully taken for the first time. Naïvely connecting nodes across time introduces a few problems, particularly with determining which node in the next frame to connect, and also magnifies a few existing weaknesses of Belief Propagation which have hitherto not been magnified. This dissertation addresses these.

1.4 This Dissertation

The aim of this research is to improve the Belief Propagation algorithm in ways that allow it to be more useful "out of the box" for use in reconstructing multi-camera video sequences: to provide inherent temporal consistency, overcome large variances in signal

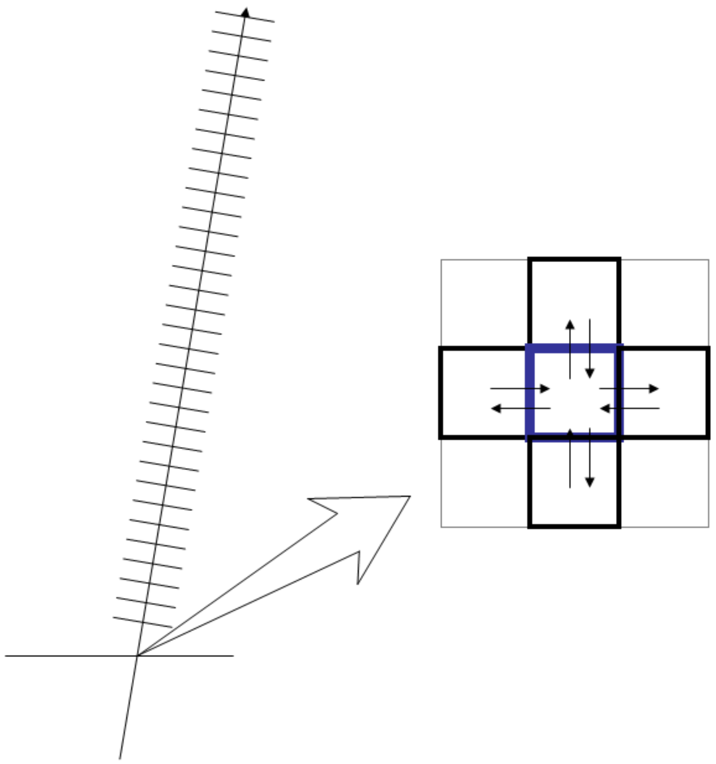


Figure 1.13: Candidate depth values along a single ray through a single pixel. Also shown is the four “neighboring” pixels: those connected to it in the Belief Propagation node graph.

to noise ratios in the same scene, and allow for native integration of priors from higher level algorithms. This dissertation presents my work toward that aim. In this section, the thesis of this my work is presented followed by an overview of the remainder of this dissertation.

1.4.1 Thesis

Belief Propagation can be enhanced to improve 3D reconstructions from multi-view video sequences. These improvements address scenes having

Rapidly Moving Objects :

- Enable extension of Belief Propagation to the temporal domain with tolerance for imperfect object tracking.
- Enhance Belief Propagation for robustness to erroneous neighborhood assignments.

Large Occlusions :

- Remove large occlusion effects.
- Enhance Belief Propagation providing for intelligent integration of an additional prior.

Nearly-Uniform Featureless Regions :

- Reduce noise and error in featureless regions.
- Enhance Belief Propagation allowing for better signal propagation through nearly uniform regions.

1.4.2 Overview

The remainder of this dissertation unfolds as follows:

Computer Vision Background Chapter 2 provides relevant background on computer vision, including exploration of the significant aspects of the problem, and discussion of related works,

Belief Propagation Background Chapter 3 introduces Belief Propagation, with historical contexts, and discusses its related applications to computer vision.

BiasedBP Chapter 4 presents a modification of Belief Propagation aimed at correcting large regions of consistently erroneous input,

RobustBP Chapter 5 presents the problem of erroneous neighborhood connections in Belief Propagation, which implies significant outliers, and an algorithm for detecting and removing these outliers,

QuietBP Chapter 6 presents a problem with Belief Propagation that I have not found documented – ineffective propagation through noisy nearly-uniform regions – and discusses potential solutions,

Results Chapter 7 presents the results, including details of implementations, experiment environments, parameters settings, etc,

and Chapter 8 concludes.

Chapter 2

Background

Computer Vision as a field has evolved over several decades. While, as a research field, it has matured and increased in rigor, it is simultaneously more poorly defined than ever: It has blended in, and developed strong research and practical interactions with, the various fields of pattern recognition, artificial intelligence, signal analysis, robust statistics, etc.

On the other hand, this work is focused on a small portion of computer vision which can be clearly defined and also traces its roots back to early classic computer vision problems. The landmark work of (Marr and Poggio, 1976) introduced a "2.5D Sketch." Now commonly known as a "depth map," the 2.5D Sketch is simply a 2D image with a distance encoded at each pixel. First, it is important to distinguish this from a 3D model: the depth map only encodes information about the distance to the first surface intersected by each ray through the image: there is no information about the back facing surfaces, occluded surfaces, or surfaces outside the view of the image. Secondly, it is important to note that the 2.5D Sketch, which in fact conveying some information, is often practically useful only when coupled with some information about a physical camera (e.g. the focal length of the camera).

(Trucco and Verri, 1998) defines two classic problems identified in (Marr and Poggio, 1976) and enduring through the decades of research:

Correspondence The correspondence problem is the problem of identifying points in more than one image which are the projections of the same 3D point, as illustrated in Figure 2.1. If correspondences are output for all pixels in an image, then the output can be represented as a depth map.

Reconstruction Given one or more correspondences, the reconstruction problem is the problem of making statements about the 3D location and structure of the scene components.

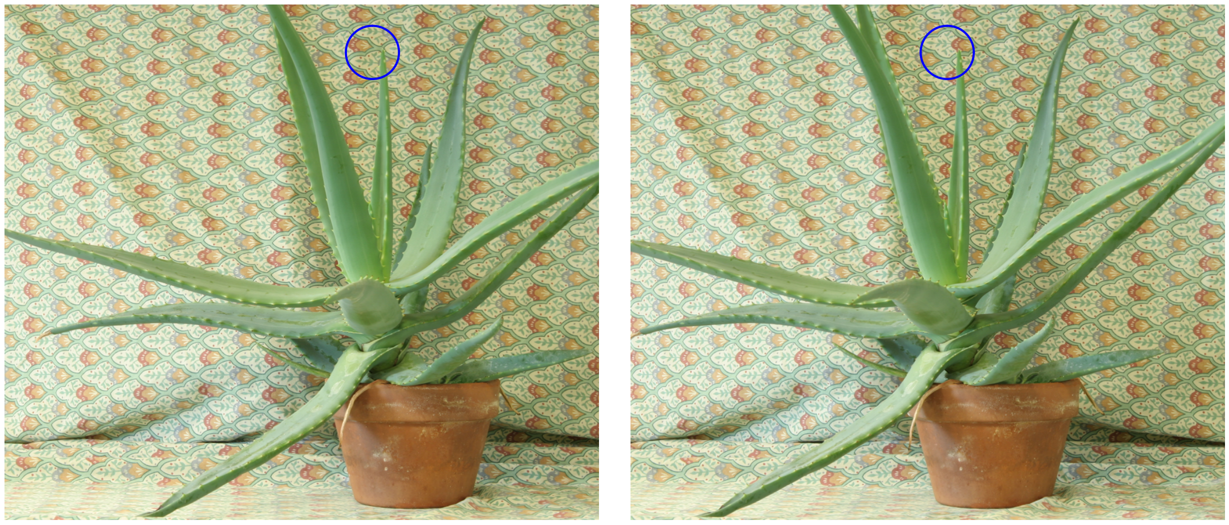


Figure 2.1: Two images with a correspondence marked. Finding correct correspondences automatically is an ongoing research area in computer vision.

Though "reconstruction" has evolved a little in modern connotations to include attempts to solve the correspondence problem, it is primarily focused on what to do with the correspondences estimated. Note that found correspondences may be dense or sparse *e.g.* it is common to find obvious features such as corners in images and attempt to find correspondences for them only, ignoring (sometimes only temporarily) the task of finding correspondences for the rest of the image. Methods only finding correspondences for select "features" identifiable in the scene shift the burden of figuring out what to do with the rest of the scene into the realm of the reconstruction algorithm, hence the modern blurring between those terms.

This dissertation's focus is completely on the correspondence problem. Simple reconstruction algorithms are used to aid in visualization and analysis of my results. Further, I have taken a more modern approach of attempting to solve the correspondence problem for a given pixel by taking the rest of the image as context, including the current correspondence estimation data for all other pixels. The output I aim to achieve is a correct depth map.

2.1 The Correspondence Problem

The "correspondence problem" is one of the oldest problems in computer vision, and it remains an area of ongoing active research, (Scharstein and Szeliski, 2002). The problem is simply: given a pixel $I(x)$ in an image, find the pixel $I'(y)$ in another image for which both pixels correspond to the same world-space points. There are a few common simplifications that should be noted:

Rectified Stereo If one is careful about the camera, and with a little image processing on the images, it can be arranged such that the corresponding pixels are always on the same horizontal scanline in both images. The process of ensuring this situation is known as "rectification." This is done for computational reasons: it simplifies the space in the second image in which the corresponding pixel may potentially lie: on the same scanline.

Optical Flow The optical flow problem is one in which the same camera takes two consecutive images, and either something in the scene has changed, or the camera has moved. The objective is still the same in both cases: for a pixel in the first image, identify the pixel in the next image corresponding to the same 3D point.

To aid in appreciation for the difficulties of the correspondence problem, I note only a few of the many subtle complexities:

Sampling The given images are composed of discrete samples of the scene, known as pixels. A pixel does not represent a perfect point sampling of the scene, but instead represents an integration of light over a sampling function (see e.g. (Castleman, 1996; Sonka et al., 1999) for details). The effects of this are apparent in many ways. One is that the 3D "point" we see correspondences for is actually a surface in 3D, and this surface may/not be continuous (e.g. part of it may be on a foreground object and part of it on a background object). Further, this 3D surface may image to more than one pixel in the other image. The potential difficulties due to sampling are many. The interested reader is directed to (Castleman, 1996; Sonka et al., 1999).

Ambiguity It is frequently the case that the correct corresponding pixel in the second image *appears* very similar (often even identical) to other pixels in that second image. This results in a situation in which it becomes harder for a computer algorithm to distinguish between the candidates. A simple example is a brick wall: there are many pixels in the second image which are perfect matches to a pixel in the first.

Occlusions An occlusion occurs when some portion of the scene is not visible to the imaging device due to some other object. Examples include the back of an object; a portion of a background object which is blocked by a foreground object in one image but not another; a portion of the scene which is not visible in the other camera's view frustum (occlusion by the camera body, actually); and so forth. Occlusions also arise in the optical flow situation as a function of motion in the scene. The result of occlusions is that there does not exist a correspondence, but that the algorithm does not always know *a priori* which pixels do not have correspondences.

Noise Sensor noise is of obvious difficulty: a pixel with a completely random color value is not likely to find a correct match in the other image (imaging sensor noise tends to change randomly from image to image, mostly). But "noise" also extends to include "anything that does not match the assumptions." A few scene aspects that are often not modeled (though some of these have been, none of them are modeled generally) include complex light reflecting surfaces, e.g. specular reflections, the retro-reflective portions of a stop sign, complex light functions as in a goblet partially filled with liquid, off-camera occluding objects, and so on.

The correspondence problem is at the heart of many computer vision problems and remains unsolved generally, particularly for complex situations.

2.1.1 Optical Flow

The optical flow problem is the correspondence problem in a special setting: same camera, sequential images. Part of the difficulty here is that things move from one image to the next. The complicating effects here usually fall into two categories: one is still just the occlusion problem – things visible in one image are not visible in the other image; and the other is that the range of candidate correspondences can be much larger – as things in the scene can potentially move to very different parts of the images.

Due to computational restraints, the effects of the second is that optical flow algorithms do not always do well when there is large motion in the scene, as illustrated in Figure 2.2.

The effects of the first (occlusion) show up similarly to non-optical flow (portions of the scene are visible in one image but not in the other), but have solutions in one application domain sometimes make assumptions that are do not hold in the other domain. This is also seen in Figure 2.2.

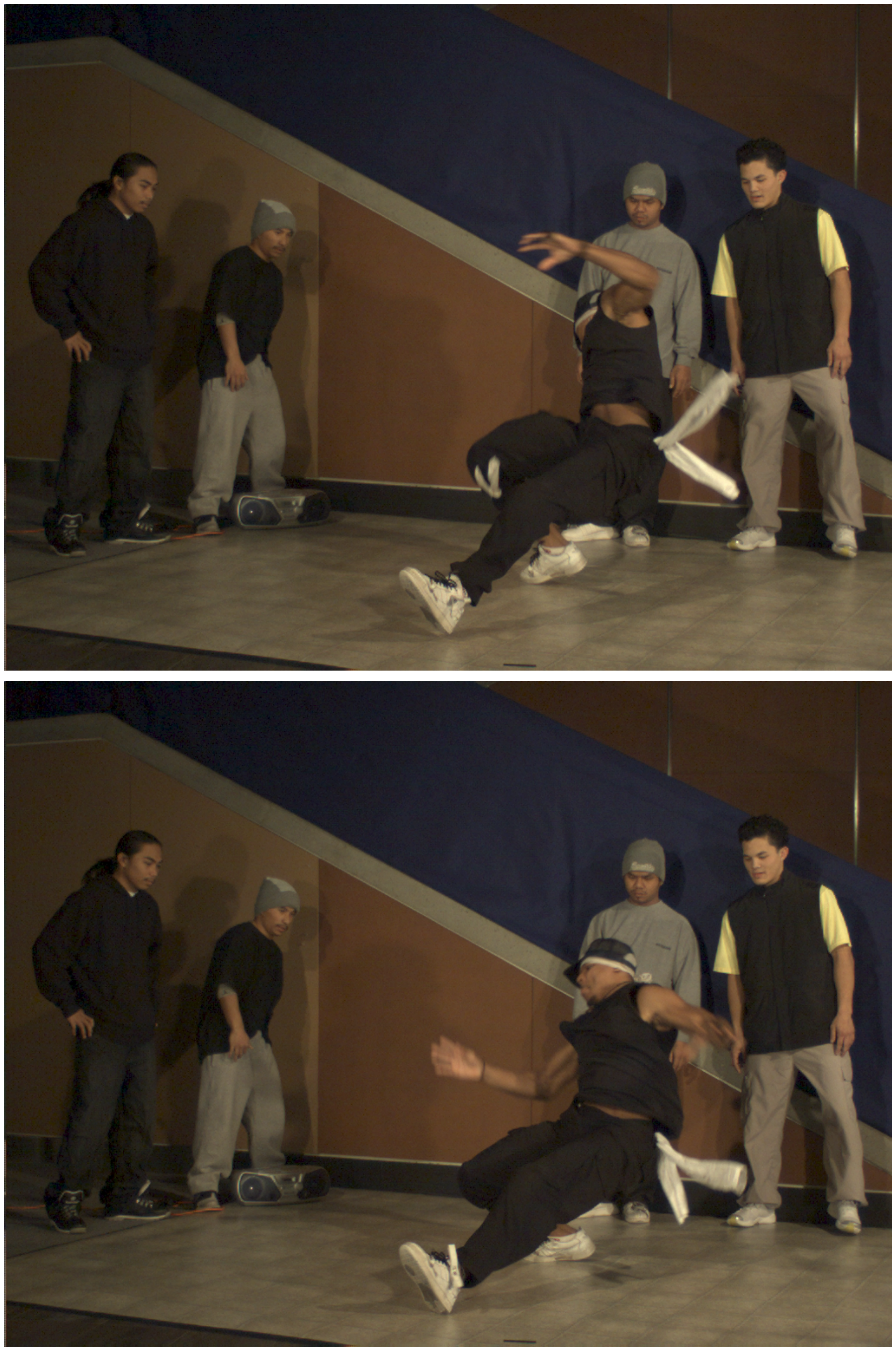


Figure 2.2: Consecutive frames from the breakdancer sequence. Optical flow will have a very hard time tracking the relatively large motions of the near arm, an impossible time with the arm which is occluded in one image, and the shadow in the middle of the back wall is also quite confusing to most systems.

2.1.2 Scene Flow

What has been termed *3D scene flow* has been explored by (Vedula et al., 1999; Vedula et al., 2005; Carceroni and Kutulakos, 2002; Neumann and Aloimonos, 2002; Zhang and Kambhamettu, 2003; Goldluecke and Magnor, 2004; Gong, 2006) but they typically require reasonable estimates of the optical flow.

A seminal approach was published by (Vedula et al., 1999; Vedula et al., 2005). It introduces the concept of scene flow which is the three-dimensional equivalent of optical flow. Scene flow is a dense motion field for all surface points of the scene from frame to frame. Its projection on one of the images gives optical flow. A key choice is to perform regularization in the images instead of the scene surfaces. A different approach was taken by in (Vedula et al., 2000) where they borrowed the idea behind the space carving algorithm (Kutulakos and Seitz, 2000) to carve away voxels that violate photo-consistency constraints in the space of the coordinates of two temporally corresponding voxels (there are six degrees of freedom here: each correspondence is a 3D position in the volume at one time and 3D position in the other). The shape approximation produced by this algorithm is tighter than what would be produced by applying space carving to the two frames separately.

(Carceroni and Kutulakos, 2002) uses dynamic surfels as primitives that encode local shape, reflectance and motion of a small surface patch. In contrast to scene flow (Vedula et al., 2005), regularization here takes place on the surface itself. The approach is limited to a small number of surfels due to the high complexity of the proposed surfel sampling algorithm and requires known illumination of the scene. (Neumann and Aloimonos, 2002) employ a time-varying multi-resolution surface that is fitted to the data using spatio-temporal multiple-view information and silhouette constraints. The shape of the surface is optimized in a multi-resolution fashion on the subdivision surface. The subdivision here, unlike (Carceroni and Kutulakos, 2002), can be updated from frame to frame.

Static multiple-view reconstruction techniques often pose the problem as the extraction of a 2D iso-surface of an appropriate function in 3D. (Goldluecke and Magnor, 2004) extend this approach to temporal reconstructions and seek a 3D iso-surface in 4D using a variational method. The 4D space can be constructed by stacking the 3D states of the scene for each time instance. The extracted 3D iso-surface represents the evolution in time of the scene surfaces.

(Tao et al., 2001) present a method based on image segmentation that models the scene as a collection of planes. It is well suited for scenes with many uniform surfaces where optical flow is not accurate but is correct at the segment level. Temporal depth hypotheses are generated for each plane based on neighboring planes and are verified in the following frame. The method is able to produce sharp surface boundaries due to image segmentation and the fitting of a plane to each segment. We achieve similar results by introducing a bias in belief propagation.

(Zhang and Kambhamettu, 2003) present two viewpoint-based approaches for the estimation of 3D scene flow. One is based on a piecewise affine motion model and operates on N frames assuming constant velocity for each patch. The second approach computes structure and motion simultaneously taking into account image segmentation information and using validation between two depthmaps to detect reliable matches. Recently, (Gong, 2006) proposed an algorithm for the estimation of disparity flow, which is a motion field that maps pixels and disparities to the corresponding pixels and disparity values in the next frame. The disparity flow fields are used to predict the new disparity maps by biasing the cost volume in favor of the predicted values. To achieve a throughput of several frames per second for the binocular case, the search range for optical flow has to be small, but the algorithm is able to recover from missed correspondences using cross-validation.

A limitation of methods such as (Vedula et al., 2005; Vedula et al., 2000; Goldluecke and Magnor, 2004) is that they require accurate spatial and temporal correspondences for most points of the scene. This makes them inapplicable to scenes with very fast motions or with significant occlusions and self-occlusions. The surfel sampling algorithm (Carceroni and Kutulakos, 2002) may be more effective when exact correspondences cannot be found but it is limited by the requirement for known illumination, its high computational complexity and the independent optimization of adjacent surfels. My approach is more similar to (Zhang and Kambhamettu, 2003) and (Gong, 2006) since it is viewpoint-based and most importantly able to recover from failures of optical flow. Unlike these approaches, I have not incorporated validation across depthmaps, but it is a candidate for future work.

2.2 Reconstruction

Reconstruction is the problem of: given one or more correspondences, make statements about the 3D location and structure of scene components. This can take many forms, both as a function of the correspondences given and as a function of the application domain and any *a priori* information about the scene (e.g. is it outdoors, a single object on a turntable, people's faces, etc.). There are a vast number of algorithms and approaches and work on various special domains. In this section I give a sampling of some common issues, components, and algorithms. Although the field is continually changing, (Seitz et al., 2006) provides a good treatment of the current state of the art.

There is interesting information that can be extracted given only a single correspondence: *e.g.* given the center of the red light in a stop-light, in two images, how far away is the stop-light? With this simple case, additional information about the two cameras is crucial. The essential information is the center of projection and the view direction of the cameras. Camera information is further discussed in Section 2.2.1.

On another extreme, correspondences are given for all pixels in both images - with pixels not visible in the other image correctly labeled as occluded. In this case as 3D point-cloud can be used to represent the surface, but some applications benefit from smooth surface representations (e.g. when the scene is to be rendered to a novel view, it is often nice to not have holes in between all the points). But it is not always that case that all points belong to the same surface, so problems arise in determining which points belong to which surface.

The latter approach does open some interesting opportunities though: one can attempt to simultaneously solve the surface identification problem and the correspondence problem. For example, one could use hypothesis about surface continuity and same-surface status to attempt to solve the ambiguity problem in estimating correspondences, and vice-versa. This hybrid approach is fast becoming one of the dominant approaches.

Another variant to the hybrid approach is to associate confidence values with the correspondences, or to maintain multiple candidate correspondences simultaneously. Then use the surface estimation algorithm to guide the correspondence algorithm in refining its estimates. For example: it is common to be able to un-ambiguously find correspondences for scene points that are singular in some way, e.g. the corners of a box. One approach is to identify correspondences for these, then approximately fit a surface to those points (e.g. planes to the faces of the hypothetical box), and then use that hypothetical surface as a base from which to search for additional correspondences.

2.2.1 Camera Calibration

Most reconstruction algorithms, whether coupled with correspondence algorithms or not, depend on camera information. The essential camera information typically includes information which is a function of the camera itself: the "intrinsices", *e.g.* the the horizontal and vertical focal lengths, position of the center of the image in the image plane, the number of pixels in the width and height of the image, and various lens distortions; and the "extrinsics", *e.g.* the position of the camera, the direction it is pointing, etc. There are a number of algorithms for estimation of these values, e.g. (Zhang, 2000; Hemayed, 2003; Svoboda et al., 2005), but interestingly, many of these techniques rely on taking a number of pictures and solving the correspondence problem for points in each one. These correspondences often need to be measured very precisely. Many common techniques involve placing known objects in the scene with the intent of making correspondence finding significantly easier for a computer to solve correctly and unambiguously (e.g. colored light emitting diodes, fabricated checkerboards, etc.). See (Hemayed, 2003; Svoboda et al., 2005) for current surveys.

2.2.2 Triangulation

When the calibration information is known for two cameras, there is an induced relationship between points in each image and points in 3D space. "Triangulation" is the term that describes the use of this relationship. The relationship is illustrated in Figure 2.3. If you know a point in each image, there can be constructed a single point in 3D which represents the approximate intersection of the rays induced by those points and their cameras. As these rays may not numerically intersect, the point nearest to both of them is usually taken as their approximate intersection. If the two image points are declared as correspondences then this intersection point is the 3D position of the surface they image.

2.2.3 Back Projection

The principle of triangulation can be turned on end for many purposes. One common one, and one illustrative of this technique – termed "back projection", is used in estimating correspondences. Given a pixel of interest in one image, for which we would like to either find its correspondence or find the 3D point it images (as in the voxel based approaches outlined in the next section), one may take a discrete sampling of candidate 3D positions for that surface. At each of these candidates, the 3D point is

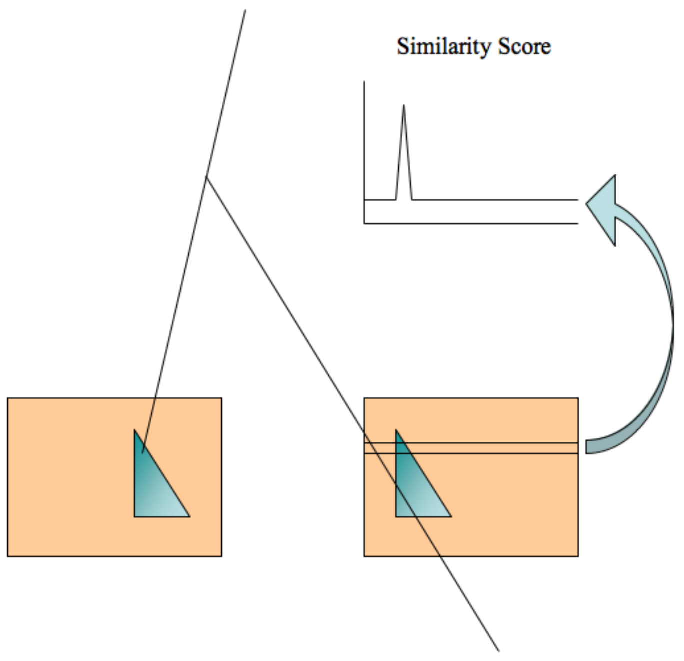


Figure 2.3: Samples along the ray through a specific pixel in the left image are back-projected onto the right image. The similarity between the pixel in the left and the pixel in the right converts to a score for each hypothesis depth value along the original ray. The graph shows scores for pixels along the scanline of the correct solution, showing a peak at the correct corresponding pixel.

projected into another camera and then the two images are compared at that pixel to "score" that candidate. Figure 2.3 illustrates this technique. One benefit of this approach to correspondence finding is that it scales trivially to many cameras. Thus, the problem becomes focused more on estimating correct 3D positions than on finding correspondences (correct 3D positions leads naturally to correct correspondences, but often opens the door to more advanced techniques which are difficult to formulate in the strict correspondence finding context).

2.3 Representations

In formulating an approach to solving the correspondence problem one must make representation choices and these influence what algorithms and application domains for which the approach is suitable. I here present a few classes of representations. These classifications are selected due to the actual differences in the algorithms and applications suitable to each representation: It is not always trivial to convert from one representation to another.

2.3.1 Depth Map

The first representation is that of a "depth map," seen earlier. Again, this is a 2D image with single values at each pixel. These values encode the distance for each pixel from the image to the 3D scene point imaged at that pixel. It is important to note a subset of these known as "disparity images" in which what is encoded is an offset count indexing the pixel in another image for which this pixel corresponds. The important note about this class is that the correspondence problem can be solved, completely and correctly, in this representation yet the exact 3D point can remain unknown. This is due to the ability to find correspondences in another image using strictly image processing techniques, *i.e.* no camera calibration information is used. With the availability of camera information, a disparity map can be converted to an image of 3D points, and vice-versa.

2.3.2 Disparity Volume

A "disparity volume" is another representation. It is used most often as an intermediate stage in the algorithms attempting to produce a depth or disparity map. This is an image which encodes some information (often current estimates of correspondence likelihoods) at each pixel for each candidate depth/disparity.

2.3.3 Visual Hull

Often referred "visual hulls", another representation is that of some "hull" information. A common approach here is to identify silhouette curves in the reference images and to represent the object implicitly as the spaces withing the intersection of the 3D general cones defined by those curves. These approaches usually need to be augmented in order to represent concavities and shape information for portions of the objects which are not silhouettes. (Baumgart, 1974; Laurentini, 1994; Matusik et al., 2000; Liu et al., 2004).

2.3.4 Voxels

A completely different representation format (but often related computationally (Kutulakos and Seitz, 2000)) is that of a voxel grid. These are usually taken to be regular grids, for computational simplicity. Common approaches include space carving (Kutulakos and Seitz, 2000; Broadhurst et al., 2001) and many variations on it (for example, (Carceroni and Kutulakos, 2002; Slabaugh et al., 2002)). This representation is commonly used via the approach of storing at each voxel some information corresponding to whether this portion of 3D space is occupied by air or by an opaque object. So the final output is a subset of the original voxels, with "empty space" voxels pruned from the set.

The application domains most commonly selected for these are those in which there are a large number of positions of the cameras, usually surrounding an object of interest. These representations have the benefit of usually representing full “water-tight” objects – as opposed to a depth image which only contains a the portions of the surfaces which are imaged by one camera (no information is known about back facing surfaces, the backs of any objects, etc.). Due to memory constraints, space carving solutions are usually lower resolution than depth map solutions, although this is not always the case (Slabaugh et al., 2002). Voxel algorithms can be used to estimate, at each voxel, additional properties of the scene, such as surface orientations (Carceroni and Kutulakos, 2002).

For this dissertation, I store at each pixel many probability distributions over the candidate depth values. The final output is a function of those returning a single depth for each pixel, in the usual depth map format.

Chapter 3

Belief Propagation Background

Belief Propagation is an "inference engine" (an engine for estimating some unknown conclusion based on some set of inputs). It is formulated as a message passing system in which messages (representing various hypothesis) are iteratively updated based on the other messages in the system. Via these iterative updates, the messages prune out hypothesis which are less supported and attempt to resolve ambiguities. This is due to the propagation of hypothesis (and support for them) through a graph of connected nodes, *i.e.* global information is integrated via modulation at local levels.

This chapter explains the formulations and how the algorithm works, attempting to both provide the rigorous and the intuitive explanations. It further dicusses the use of Belief Propagation in computer vision.

3.1 Formulation

Belief Propagation gained popularity beginning with the work of (Pearl, 1998). From this work arose the usage of Belief Propagation for error correcting codes ((Caire et al., 2003), with a significant application being Turbo Codes (McEliece et al., 1998). Belief Propagation found usage in computer vision applications with (Freeman et al., 2000; Petrovic et al., 2001; Sun et al., 2003; Tappen and Freeman, 2003).

Belief Propagation is an inference engine, *i.e.* an engine for estimating some unknown conclusion based on some set of inputs. For the depth extraction application here, this is estimating depth given functions of the input images (examples of these functions are detailed in the next section).

Belief Propagation is formulated as an iterative message passing system. It is often applied in the context of a Markov Random Field (MRF), defined by the energy function:

$$E(x) = -\ln \sum_{i} \phi_i(x_i) - \ln \sum_{(ij)} \psi_{ij}(x_i, x_j) \tag{3.1}$$

where x is a node in the graph and edges in the graph connect x_i and x_j. ϕ is the energy "potential" at each node (also often termed the "observation" term or the "data" term). ψ is the potential on the edges (also often termed the "clique compatibility" term).

The data term, ϕ, is a probability distribution function (PDF) across some set of states, often discrete. Further, there exists some set of states (not necessarily the same set) which are possible outputs for the inference engine. For our application, the inputs are PDFs across the potential depth values at each pixel, taken from processing the images as detailed in the next section, and the outputs are also probabilities for each potential depth value. All candidate depth values, for all pixels, are simultaneously retained at all stages of the algorithm: BP updates the currently estimated support for each candidate as a function of the current support for the other candidates combined with the estimates that neighboring pixels have in their candidate depth values.

The compatibility function, ψ, is a regularizing term expressing the affinities between neighboring nodes. For our application, this is often a simple low-pass filtering function on the PDFs (belief in one state means close states are somewhat likely), with the strength of the blur sometimes a function of the color similarities in the image pixels corresponding to the nodes (as detailed in the next section).

Belief Propagation is an iterative message passing system. It stores, at each node, an outgoing message for each of its neighbors. The incoming message from ϕ is fixed. For our application, the neighborhoods (the edges in the graph) are most often taken to be the 4-neighborhood of pixels around the pixel corresponding to each node. As illustrated in Figure 3.1, this corresponds to a simple 2D regular discrete lattice. All messages in this application are simply *PDF*s over candidate depth values. Belief Propagation iterates on updating these messages. There are a few slightly different notations used for this (Sun et al., 2003; Felzenszwalb and Huttenlocher, 2004; Yedidia et al., 2001), I will use the common notation formulated as:

$$m_{ij}^{s}(x_j) \leftarrow \max_{x_i} \phi_i(x_i)\psi_{ij}(x_i, x_j) \prod_{k \in \mathcal{N}(i) \setminus j} m_{ki}^{s-1}(x_i) \tag{3.2}$$

where s is the iteration, i and j identify nodes, x_i is a random variable over the state space at node i, $\mathcal{N}(i)$ is the neighborhood of x_i (the set of nodes that have edges connecting to it), and m_{ij} is the message from node i to node j. Again, ϕ is the observation and ψ is the compatibility function. Note that no term represents the "output" or "current estimate" concept. After any number of iterations, the current best estimate can be obtained from the system via the "belief" at each node:

$$b^{s}(x_j) \leftarrow \max_{x_i} \phi_i(x_i)\psi_{ij}(x_i, x_j) \prod_{k \in \mathcal{N}(i)} m_{ki}^{s}(x_i) \tag{3.3}$$

i.e. all the incoming messages are combined with the observation to produce a single *PDF*, and the most likely hypothesis is directly selected from that. Thus, the *belief* can be seen as a method for sampling the current state of the system. As noted earlier, the belief is not used internally to the system: there is no reason to store it and it is not a component of the message passing.

These are the key components involved in Belief Propagation. I recommend and use the method of (Felzenszwalb and Huttenlocher, 2004) for determining the order of processing message updates, though there are other update schedules and some appli-

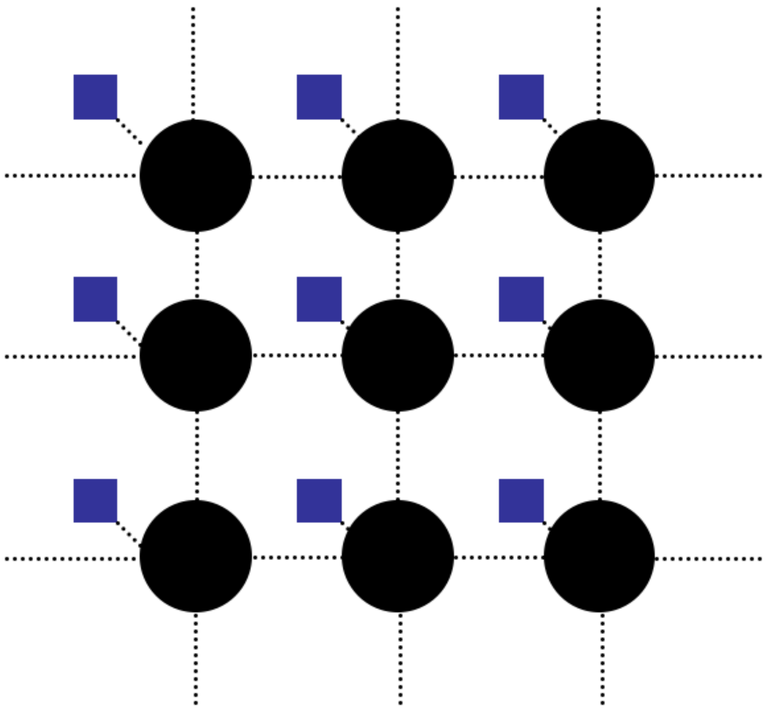


Figure 3.1: Illustration of common Markov Random Field lattice used for reconstruction. The bold black dots represent nodes, corresponding to pixels in the reference image. The blue squares are the observation inputs. The dotted lines represent edges in the graph, along which messages pass in Belief Propagation and along with edge potentials are defined in Markov Random Fields.

cations where this schedule will not work (*e.g.* irregularly connected graphs). Their schedule follows a checkerboard pattern, updating all the "blacks" then all the "reds," with coloring as a checkerboard: no black is adjacent to a black and vice-versa. The interested reader is directed to (Yedidia et al., 2001) for an excellent presentation of the topic, and (Sun et al., 2003) for an example of Belief Propagation used for inference in images, related to this work and discussed in the next section.

3.2 Belief Propagation in Computer Vision

(Sun et al., 2003) introduced Belief Propagation to the application domain of binocular stereo. Their approach places one node for every pixel in the reference image and connects each node to its four neighbors (those adjacent horizontally and vertically). Their compatibility function is a Gaussian blur with the variance term decreasing as pixel color is more similar between the pixels for the node and its neighbor. I use the same method. The observation term for each node is computed via image processing: for each node, for each candidate depth, compare the pixel intensities in the reference image to those in the other image at the pixel(s) corresponding to this depth value. There are a handful of metrics that can be used here. One very common metric (particularly for Belief Propagation implementations) is simply the single pixel intensity difference:

$$\|I_L(s) - I_R(s')\| \tag{3.4}$$

where $I_{L\|R}$ is the left and right image, respectively, and s and s' are the respective pixel coordinates. The norm, here represented via $\|x\|$, is usually the L2 norm in the three color channels: Red, Green, and Blue - though a few researchers use more advanced color spaces and norms. (Birchfield and Tomasi, 1998) improves on this to make it less susceptible to sampling artifacts. I implemented this and also the significantly more expensive metric of (Yoon and Kweon, 2005), finding very comparable results after Belief Propagation converged.

3.2.1 Fundamental Differences

Belief Propagation has been used substantially for stereo since (Sun et al., 2003), as surveyed in (Klaus et al., 2006). These papers all take the approach of using Belief Propagation as is, *i.e.* the algorithm remains unchanged. All of these approaches make their advances via factoring additional information into the observation and compatibility functions, ϕ and ψ. These primarily include variations targeting better occlusion estimators and better edge detection metrics. In contrast, the approaches in this dissertation all leave ϕ and ψ untouched, but alter the Belief Propagation algorithm itself. Chapter 4 explores adding biasing terms to the equations – this is a fundamental change in the approach taken for accomplishing this goal. Chapter 5 explores dynamically changing the connectivity of the graph itself, automatically, on each iteration – though conceivably possible, it is not clear how to frame this as a change to the model and not the algorithm. Changing the algorithm appears significantly simpler and more intuitive. Chapter 6 explores an approach aimed at fixing a fundamental flaw in Belief Propagation: its susceptibility to large regions of consistent errors.

All of these approaches differ from current Belief Propagation research in that, instead of integrating more information into the model and leaving the algorithm untouched, I have altered the algorithm itself in specific ways in order to more simply, more intuitively, and I would argue more efficiently achieve better results than the current state of the art. Further, my modified algorithm is more suited to the multi-camera temporally consistent application domain of office environments of which I have focused on.

Chapter 4

Biased Belief Propagation

Belief Propagation in practice is robust to “salt and pepper” noise (noise in which the noise at each node is independent of noise at neighboring nodes.) Belief Propagation does not do so well when the noise is consistent for large regions of nodes. This is due to the fundamental nature of BP: if a node and all its neighbors are wrong consistently, they will reinforce each others error and make it harder for nodes further away to correct their unanimous error. By “noise” I mean: The observation data does not match our model assumptions. By consistent, I mean that the input for all the pixels in the region is wrong in the same way: it is not an effect which varies randomly over the region. For a specific example, consider the regions of the reference image in which the other images do not have corresponding pixels - so called “occlusion” regions. Section 4.0.2 illustrates how the *PDF* for all the rays in a region can be consistently unreliable.

To help overcome this weakness, I present BiasedBP. BiasedBP is so named because it provides a mechanism for providing and integrating a bias. The expectation is that the bias is constructed by some algorithm which does not suffer from this specific weakness, so the two algorithms can complement each other. Higher-level processing can produce a prior estimate which BiasedBP uses to help overcome regionally consistent errors in the input. Many higher level algorithms could be used to produce this prior. In this chapter I investigate two simple ones: plane fitting and background estimation.

Section 4.1 presents our integration of a general bias into the BP equations. Sections 4.2 and 4.3 discuss two different ways to produce the bias, each showing results of the technique on common computer vision datasets. BiasedBP can significantly reduce errors due to occlusions. BiasedBP remains general enough that it could potentially be applied to applications of BP outside of computer vision, although this has not been explored in this dissertation. Section 4.4 summarizes this chapter.

4.0.2 Occlusion Effects

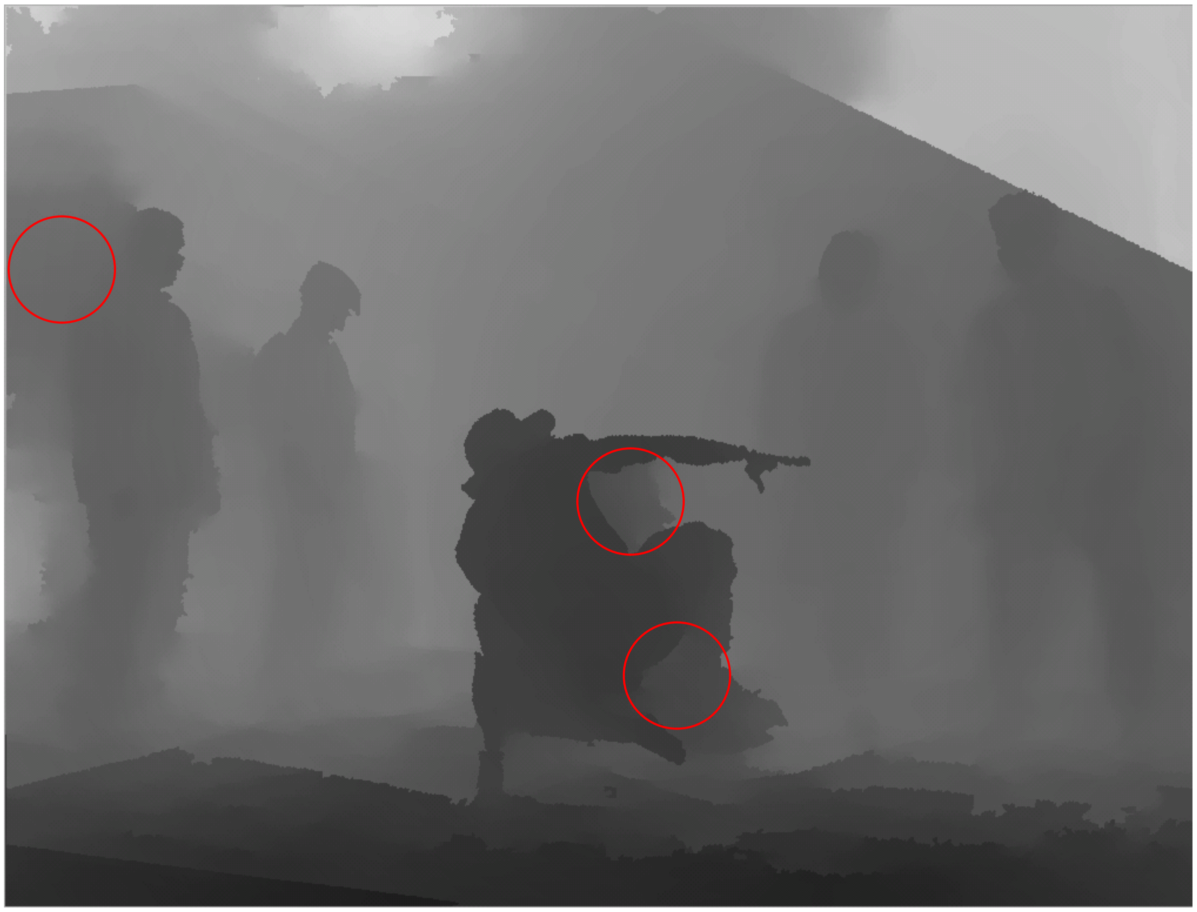


Figure 4.1: State of the art depth map for breakdancers (Zitnick et al., 2004) showing regional artifacts caused by occlusions. Some specific ones are highlighted with red circles. Note that (Zitnick et al., 2004) focused on video quality and not correct depth values.

A pedagogical example of regionally consistent error is the very common occurrence

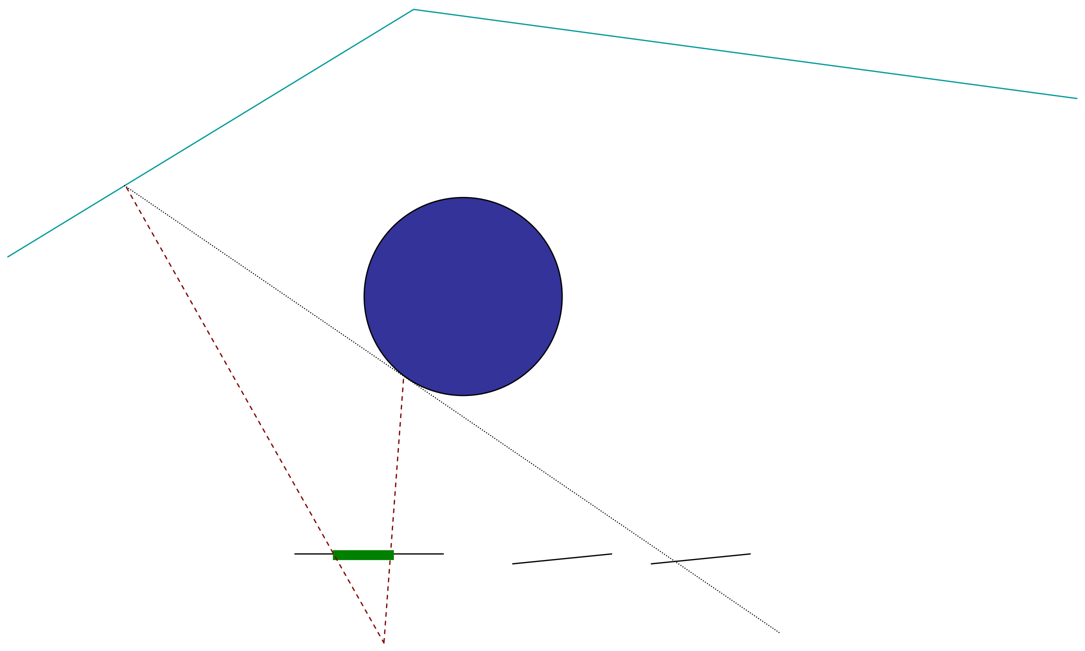


Figure 4.2: Diagram of occlusion effects: There are three cameras across the bottom. All pixels in the rightmost camera which are to the right of the sight line see the blue ball. All rays from the left camera between the two red dotted lines have their observation PDFs computed via scoring that includes these pixels from the right-most camera. As a consequence, observe that for *all* the rays in the left most camera that are within that range (highlighted by the green thick bar), the PDFs contain error consistent for large regions. Note that this error is consistent for neighboring pixels within that band. Thus, simple neighborhood comparisons will not fix this problem: all pixels in large regions are affected the same.

of occlusions. Figure 4.2 illustrates why occlusions caused by scene geometry cause large regions of the input to have consistent errors. The core observation here is as follows. An occluder in the scene causes the correlation score for some depths to be computed reflecting the incorrect assumption that some other image contains the corresponding point. The scores for a very similar set of depths computed for a neighboring pixel are very likely to be very similarly incorrect. This is because the occluder occupies some finite amount of image space. Thus, correlated error is induced in *PDF*s for neighboring pixels. If all pixels in a large region have consistently erroneous input, as in this example, then Belief Propagation does a poor job of overcoming that effect. Figure 4.1 highlights a few places where these effects are seen in a real dataset. BiasedBP produces the correct results in these regions.

4.1 Biased Belief Propagation

In this section, I present BiasedBP: Belief Propagation with explicit biasing. The "bias" is a prior applied to each node. It may be different for each node. Each prior is a full *PDF*. The representation of the prior need not be the same as that for other *PDF*s in the system. For example, a parameterized Normal distribution could express a prior.

The purpose of BiasedBP is to overcome regionally consistent erroneous input which Belief Propagation does not handle well. As this is the case, we turn to some other process to generate the priors. This could also be thought of as "augmented BP" i.e. Belief Propagation augmented with the output of some other algorithm. The important thing to note though is that this is fundamentally different from pre-processing the input given to Belief Propagation. This is because the integration of the prior into Belief Propagation's messages is varied per node and per iteration as a function of the current messages in the system. Precisely, given the energy equation

$$E(x) = -\ln \sum_{i} \phi_i(x_i) - \ln \sum_{(ij)} \psi_{ij}(x_i, x_j), \tag{4.1}$$

Belief Propagation would commonly solve this by initializing all outgoing messages to uniform *PDF*s and then using the update rule

$$m_{ij}^{s}(x_j) \leftarrow \max_{x_i} \psi_{ij}(x_i, x_j)\phi_i(x_i) \prod_{k \in \mathcal{N}(i) \backslash j} m_{ki}^{s-1}(x_i). \tag{4.2}$$

Using the same energy equation, BiasedBP uses the modified update rule

$$m_{ij}^{s}(x_j) \leftarrow \max_{x_i} \psi_{ij}(x_i, x_j)\omega_i^{s-1}\theta_i(x_i)\phi_i(x_i) \prod_{k \in \mathcal{N}(i) \backslash j} m_{ki}^{s-1}(x_i). \tag{4.3}$$

The prior is θ_i which is a full *PDF* over all potential states x_i. Its application to the update is controlled via the scalar ω_i^{s-1} - per node and per iteration. Intuitively, ω is a function of the uncertainty the model has in its current belief at that node: low uncertainty results in the prior having minimal effect while high uncertainty leans more heavily on the prior. This formulation allows for soft determination of regions in which the bias is needed, as detection of such regions (e.g. occlusion region detection) remains an open research area. This formulation also is independent of the method used to construct the prior, θ. Sections 4.2 and 4.3 present two very different approaches for setting the prior.

ω is set on a per-node, per-iteration basis. The aim of its value is to correct for regionally erroneous common observations, ϕ, while having minimal impact on nodes that otherwise would have done fine. Regionally erroneous ϕ values occur in regions of occlusion. Identifying these regions remains an unsolved problem, but setting up the equations as I have allows for producing a super-set without negative effects to regions that are actually not occlusion areas. Specifically, labeling as occlusions the areas that are in occlusion and also small bordering regions is sufficient. So a perfect solution to the occlusion labeling problem is not necessary.

4.2 Local Plane Fitting

One approach to establishing a prior is by plane-fitting, on the assumption that if you do not have enough information, then linear approximation is a reasonably solution for segmented regions (regions which are estimated to be very likely to be the same surface). I use the common mean-shift algorithm (Comaniciu and Meer, 2002) for image segmentation. Each resulting patch is some collection of pixels in the image, based entirely on color similarity. At every iteration in the Belief Propagation algorithm, the belief at each node can be calculated. From this, the single most likely depth for each node is taken as the temporary candidate depth for that pixel. Note that before any iterations of Belief Propagation are run, the observation at each node is the belief at that node, so the plane estimation phase can be (and is) done before each iteration.

The current best depth estimate at each pixel corresponds to some 3D point for that pixel. The 3D points for each of the pixels in a patch (from the color based segmentation earlier) form an estimate of the scene surface for that patch. If we do robust plane fitting to these points, we could convert the resulting plane to a prior for the pixels in this patch. This is the approach detailed here.

Q_i^{s-1} denotes the plane fit by RANSAC (Fischler and Bolles, 1981) to the 3D points given by the depth estimates at iteration $s-1$ for all the pixels inside the patch which includes pixel i. This plane intersects the ray through pixel i at point P_i^{s-1} which corresponds to slot D_i^{s-1} in the prior PDF for pixel i. Then θ is set as

$$\theta_i^{s-1}(x_i) = \|x - D_i^{s-1}\| \tag{4.4}$$

and then θ_i^{s-1} is normalized such that it sums to 1.

ω_i^{s-1} is set to reflect the current uncertainty of the belief at a node combined with the similarity between the color at that pixel and the average color of the patch to which it belongs. Then

$$\omega_i^{s-1} = \frac{\max_x(belief_i(x_i))}{\|Color_i - PatchColor_i\|}. \tag{4.5}$$

θ and ω are set simplistically and could be explored more, but the purpose here is just to show that a very simple prior can be used successfully. One common first approximation for the prior is fitting planes to the apparent surfaces in the scene. I illustrate that this can be easily done. Drawbacks of this approach include the fact that a patch in an occlusion area needs to be large enough that RANSAC will find the right plane even with all the pixels in the occlusion region being outliers. This is fine if the occlusion regions are small, as in the Middlebury dataset, but with wider baselines the occlusion regions become larger. Larger segmentation's have the drawback of the plane being less of a good surface approximation, and also the increased potential for under-segmentation (patches which cross physical surface boundaries), in which case the prior can be completely wrong.

4.2.1 Results

Figure 4.3 compares results on the left frame of the Tsukuba dataset (Nakamura et al., 1996) using plane fitting as a prior and using no prior. For the plane fitting, I use the mean-shift algorithm. It has a parameter for color similarity (which I set to 40) and a parameter for the minimum patch size (I set to 256 pixels) producing the segmentations showing in Figure 4.4.

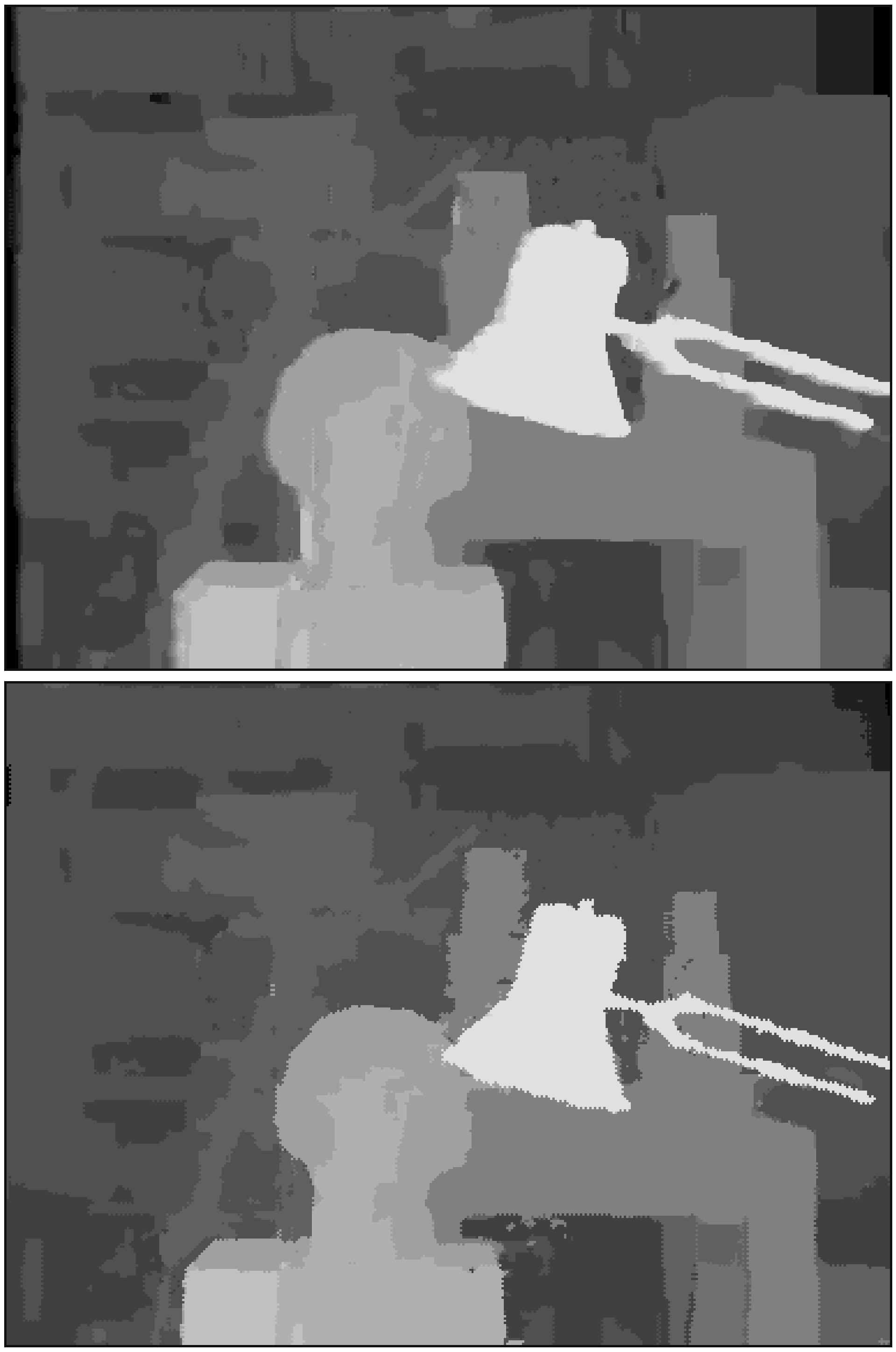


Figure 4.3: Comparison of results without (top) and with (bottom) biased via plane segmentation on the Tsukuba data (Nakamura et al., 1996). While obtaining a tighter fit (removing occlusion effects), it produces a more jagged look, due to making the more difficult decisions about pixels on the boundaries of objects (pixels often image both the foreground and background objects, so labeling it either is incorrect – future work includes not using border pixels in this stage, or weighting those pixels as belonging to both).

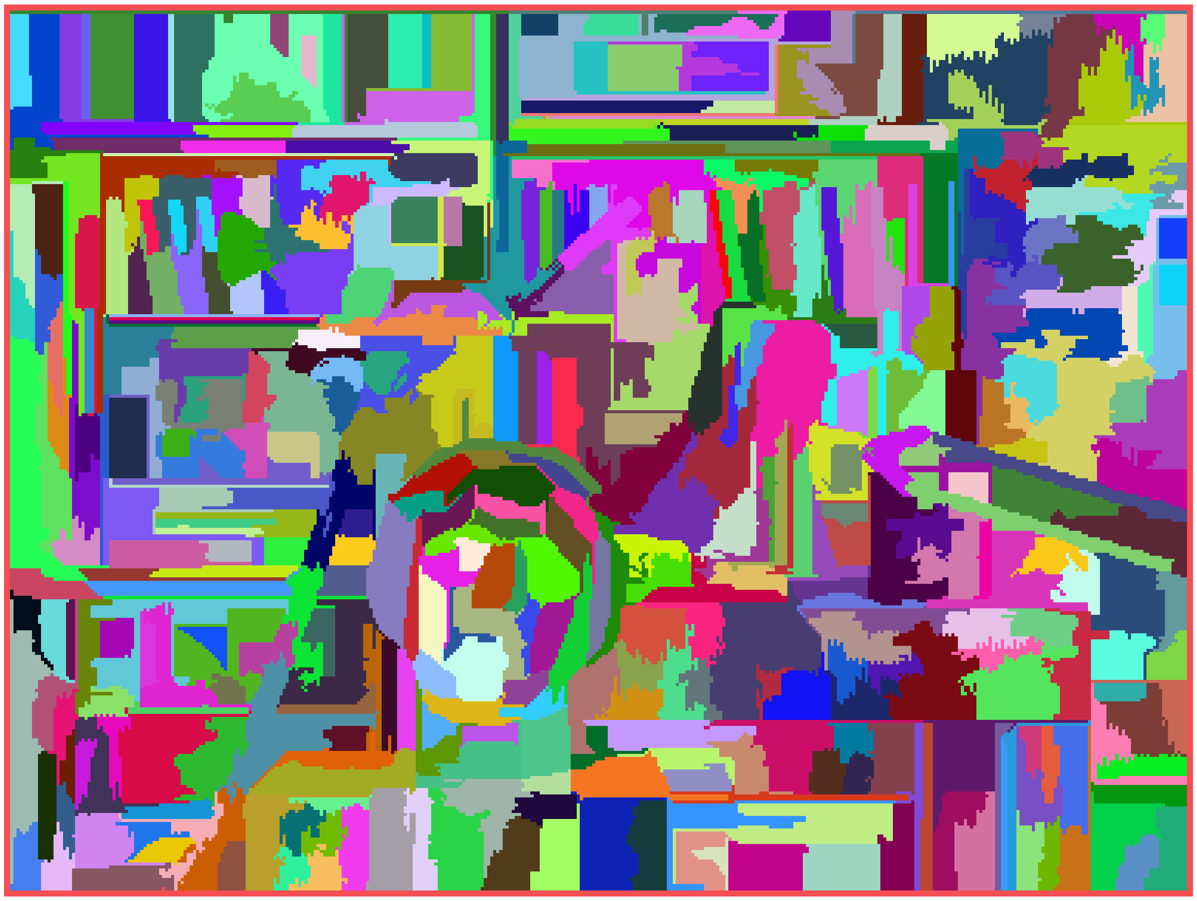


Figure 4.4: Segmentation of the left image in the Tsukuba data (Nakamura et al., 1996) using mean-shift algorithm.

4.3 Background Model Estimation

When we consider the problem of regionally consistent errors in the observations, e.g. in occlusion regions, one observes that often these regions change temporally in video streams. An example of this is the breakdancer in the breakdance sequence (Zitnick et al., 2004): the breakdancer occludes certain regions in every frame, but the regions he occludes in any given frame are not occlusion regions in all the frames. This observation lends itself to the suggestion that one approach to addressing regionally consistent errors is to search through the video for frames in which this effect is not present, e.g. find frames in which the breakdancer does *not* occlude the specific region. The approach taken in this section is to build a prior for BiasedBP based on an estimate of the static portions of the scene, a "background" model.

4.3.1 The Background Model

The "background" is a color image and a depth image: at every pixel is a color value and a depth value. The color and depth represent the estimated color and depth of the static portion of the scene. I selected the HSV (Shoup, 2001) representation for color for the reasons that it is commonly used in computer vision and image processing domains: it provides a simple (and admittedly slightly simplistic) way to reduce the effect of lighting changes, via down weighting the "value" channel of the color. Notationally, the background model is then an image with the following components, termed to be in "HSVD" space:

Hue the hue of the estimated background color, represented as an angle,

Saturation the saturation component of the estimated background color,

Value the value component of the estimated background color: down weighting this term relative to the others reduces the significance of how bright the pixel is, e.g. for robustness to when it is and is not in shadow,

Depth the estimated depth of the background.

Note that this estimation of the background is based on an assumption that the camera is fixed relative to the background objects in the scene.

Estimation

Estimation of the background model is done on a per-pixel basis. Every frame in the sequence provides a single sample which is used to estimate the depth and color of the background. For each frame in the sequence, I produce a point in HSVD space via the frame's color image and the best depth estimate from the observation. With N frames in the sequence, we now have N points in HSVD space for this pixel. These points are then clustered via K-Means(MacQueen, 1967) with a distance metric which blends only the depth, saturation, and hue components. The value terms are dropped to lessen shadow sensistivity. This results in distance being defined by

$$\begin{aligned} \|A, B\|_{HSVD} &= \alpha(\|A_{Depth}, B_{Depth}\|_{L1}) + \beta(\|A_{Saturation}, B_{Saturation}\|_{L1}) \\ &\quad + \|A_{Hue}, B_{Hue}\|_{angle} \end{aligned} \tag{4.6}$$

where $\|\cdot\|_{angle}$ correctly measures the angular distance and α and β are weights which put all terms in a compatible space. The single point representing the background is then taken as the median point in the cluster with the largest depth, i.e. the cluster furthest away from the camera is taken as the background, and the median from that cluster is taken as the most noise-free representative of that cluster. Note that the mean is not selected due to the undesirable blending qualities of HSV space. Further, simply selecting the point for each pixel which has the largest depth has a susceptibility to noise, in practice.

To select how many clusters to use for k-means clustering, I independently evaluated the points for each pixel using $2 \leq k \leq 10$ and selected the value of k having the smallest total distance from each point to the mean of its corresponding cluster. This is a very simple approach aimed at demonstrating that a simple background prior is obtainable and useful. More sophisticated background determination could be substituted within the same framework.

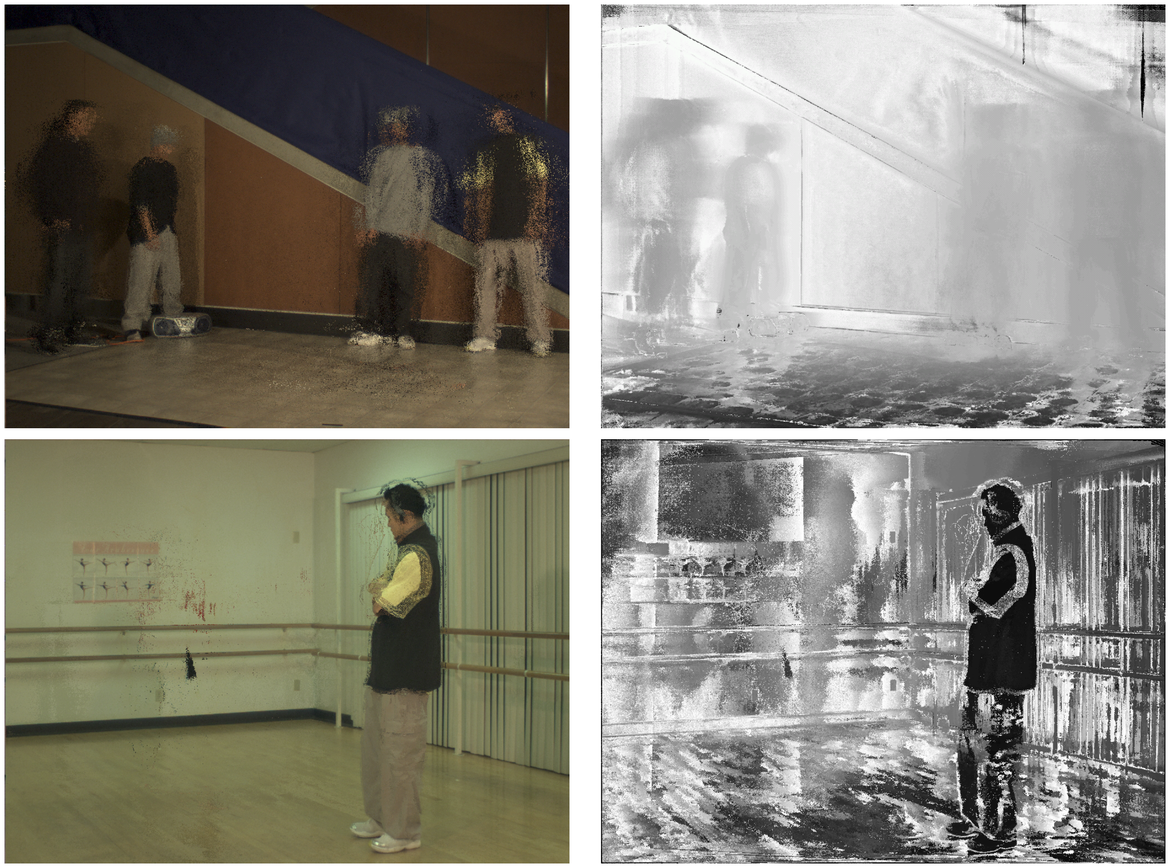


Figure 4.5: Background model (RGB Color on left and Depth on right) for the breakdancer and ballet sequences ((Zitnick et al., 2004)). Both results were obtained using 100 video frames and between 2 and 10 clusters (automatically selected per-pixel). Note that the people show up as "background" in regions where the true background is never visible.

4.3.2 Usage and Results

Figure 4.5 shows the background estimates for two datasets. A hundred consecutive frames from each sequence were used, with between 2 and 10 clusters per-pixel. Note that foreground objects with large motions are likely to be removed (like the breakdancer) but foreground objects with small motions are not removed as well (like the bystanders). Also note that there are no frames for which the observation depth estimates are correct for the occluded regions around the bystanders: the assumption that the background estimate is correct does not hold here.

Once the background model is built, it can be applied in the BiasedBP framework via

$$\begin{aligned}
\omega_i^{s-1} &= (\max_x(belief_i(x_i)))^2 \exp(-\|Color_i - PatchColor_i\|_{HSV}/\gamma_{color}) \quad (4.7)\\
\theta_i^{s-1}(x_i) &= \|x - Depth(Background_i)\| \quad (4.8)
\end{aligned}$$

Figure 4.6 shows that using the simple background model presented here as a prior for BiasedBP significantly improves the quality of the results in large consistently erroneous regions. This is clearly seen in the occlusion regions. Further, using BiasedBP does not have a negative impact on the result quality in regions where the inputs are not regionally consistently wrong, as illustrated in the rest of the images and highlighted in Figure 4.6.

4.3.3 Discussion

The k-means clustering approach assumes there is more than one surface visible at each pixel, through the video. This does not hold for some regions of the scene, such as the floor near the very front of the breakdancer sequence. It also assumes that the furthest back cluster corresponds to some static background object. This assumption does not hold for the pixels near the center of some of the bystanders as they move but not enough that the scene behind them is visible. Finally, this approach assumes that a sufficient number of depth values in the observations for each pixel are close enough to the correct surface that the prior will help.

In spite of many of the assumptions not holding all the time for all the pixels, the results suggest that this simple approach may be sufficient for many application cases.

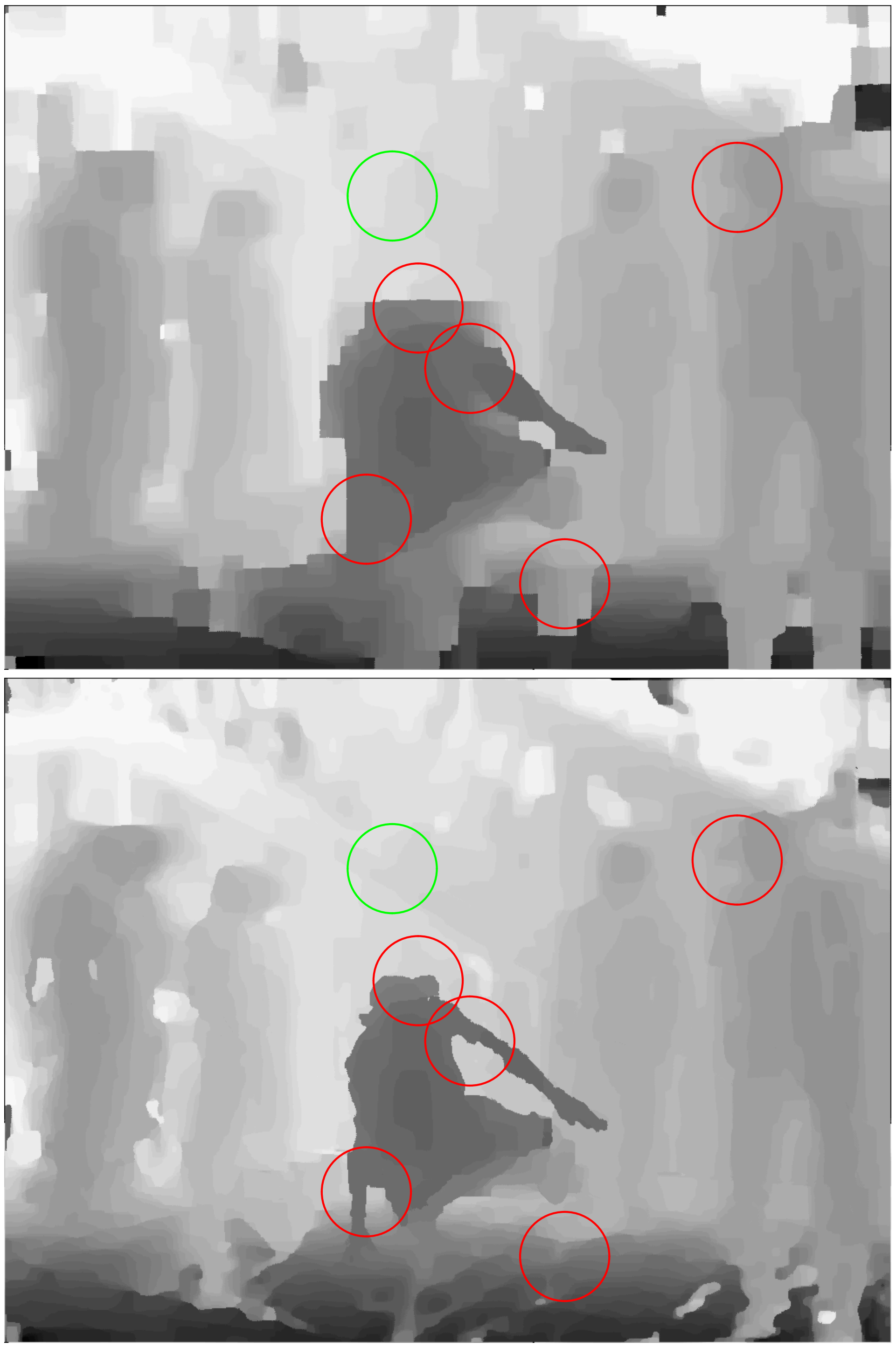


Figure 4.6: Comparison without (top) and with (bottom) using BiasedBP, red circles highlighting improvements in occlusion regions and green highlighting that minimal change has occurred in non-occluded regions.

4.4 Summary

There are many potential reasons for the occurrence of regions in an image in which the observations are consistently erroneous; the focus here has primarily been occlusion regions. Belief Propagation does not natively handle this class of noise well. Many times, these regions are cases for which a higher-level algorithm can be employed to provide a prior estimate for these regions. BiasedBP as presented here provides a simple manner for these priors to be automatically integrated into the Belief Propagation updates on a per-node, per-iteration basis. I have shown two simple higher-level algorithms for producing priors and shown that they quite simply plug into the BiasedBP framework and can provide significant improvements to the quality of the results.

Chapter 5

Robust Temporal Belief Propagation

The motivating premise in this chapter is the notion that results from estimation on one frame, using modern algorithms, should potentially be useful in producing estimates for frames temporally near (*e.g.* the previous and/or next frames). (Vedula et al., 1999; Vedula et al., 2005; Carceroni and Kutulakos, 2002; Neumann and Aloimonos, 2002; Zhang and Kambhamettu, 2003; Goldluecke and Magnor, 2004; Gong, 2006) attempt to improve the quality of the results given multiple frames in a video sequence but they rely on reasonable optical flow results. Chapter 2 discusses how those relate to the work presented here.

To make the topics in this chapter more clear and prevent confusion, I briefly define some terminology. Each of these will be further discussed throughout this chapter, the first is the driving motivation (but it does not work), the second fixes the weaknesses there, and the third is a generalization of the second:

TemporalBP This is the naïve approach, first presented here but I presume surely not novel – it does not work very well as is.

TemporalRobustBP This is the primary focus of this chapter. It is a modification to TemporalBP enabling it to work much better.

RobustBP The modification making RobustTemporalBP can be more generally applied to non-temporal settings, so it is presented as simply "RobustBP."

The approach taken in this chapter is to connect node neighborhoods such that they include temporal neighbors in addition to spatial neighbors. The temporal neighbors are nodes in previous and next frames for pixels which correspond to the pixel in the center frame. Specifically, if node x_i correctly corresponds to node x_j in the next frame, then the neighborhood of x_i should include x_j; otherwise x_j should not be included.

Assuming the correspondences are correct, then this approach is implying an important assumption. This assumption is that temporally corresponding pixels have nearly the same depth values. Depending on the formulation of ϕ, this commonly includes cases where linearly interpolating the motion is correct – those approximated by constant first derivative. This assumption does not always hold, of course, but in practice it holds for many cases, including objects moving at constant speed toward, or away from, the camera.

Obtaining temporal correspondences could be done in many ways. In one way, it is very similar to the stereo correspondence problem, except where things have changed appearance. A more advanced approach is to recognize that this is exactly the problem set out to be answered by the approaches termed "optical flow" (Galvin et al., 1998; McCane et al., 2001). But, there are settings where current optical flow algorithms frequently fail. These commonly include scenes with large and/or fast motions, scenes where traditional stereo correspondence problems fail, scenes including large untextured regions, etc.

The objective of this chapter is to provide a system which retains the improvements of Temporal Belief Propagation (detailed in the next chapter) where the optical flow is correct and does no worse than 2D Belief Propagation where the optical flow is incorrect. The approach taken towards this aim is modification of Belief Propagation such that it attempts to identify when node neighborhoods are incorrectly connected and break those connections.

In this chapter, I present Robust Temporal Belief Propagation. Temporal Belief

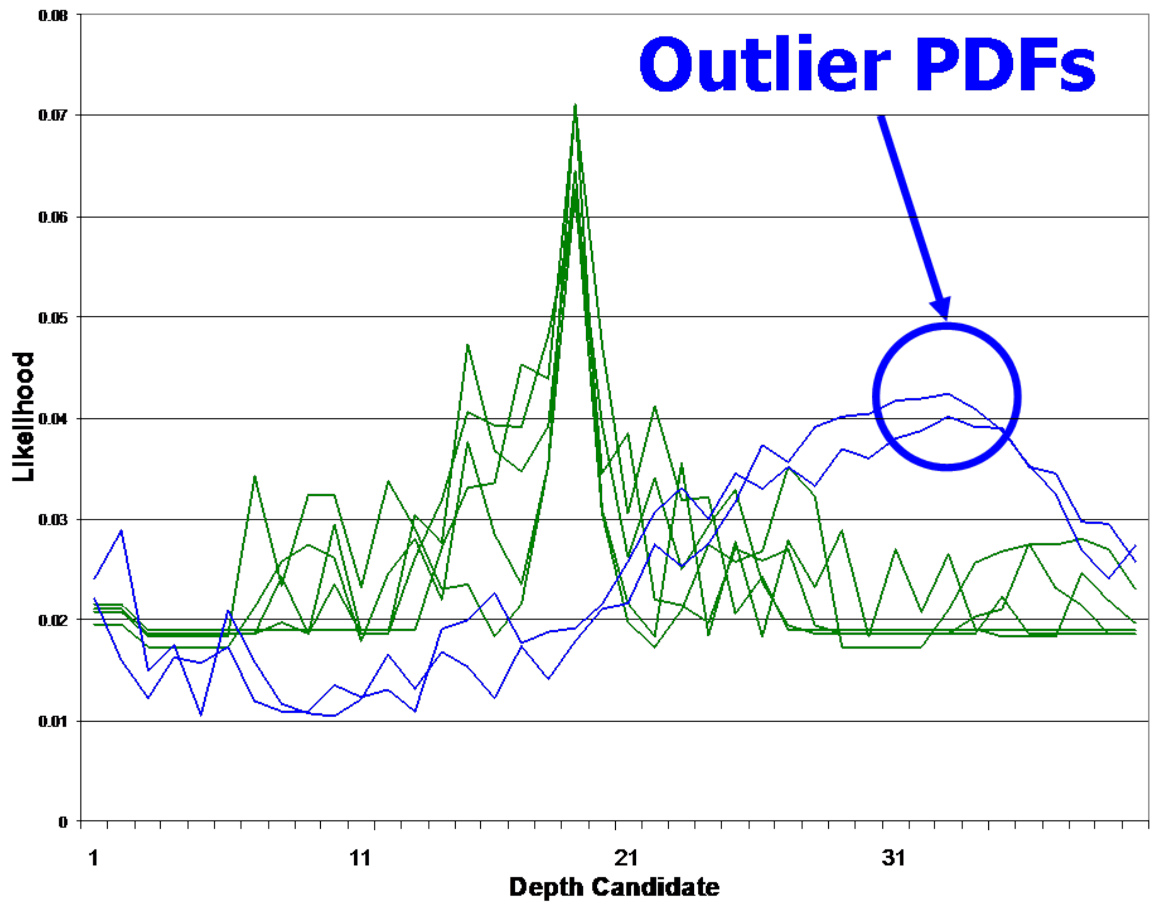


Figure 5.1: Seven incoming *PDF*s at a single node (one observation, four spatial neighbors, and two temporal neighbors). Note that two of them appear to be "outliers" - *PDF*s corresponding to completely different surfaces. These may have come from erroneous temporal connections or spatial connections. The objective of this chapter is to make Belief Propagation robust to such outliers. These are actual *PDF*s from the breakdancer sequence (Zitnick et al., 2004), frame 2, a pixel near the breakdancer's elbow. The two highlighted blue ones are the erroneously connected temporal neighbors.

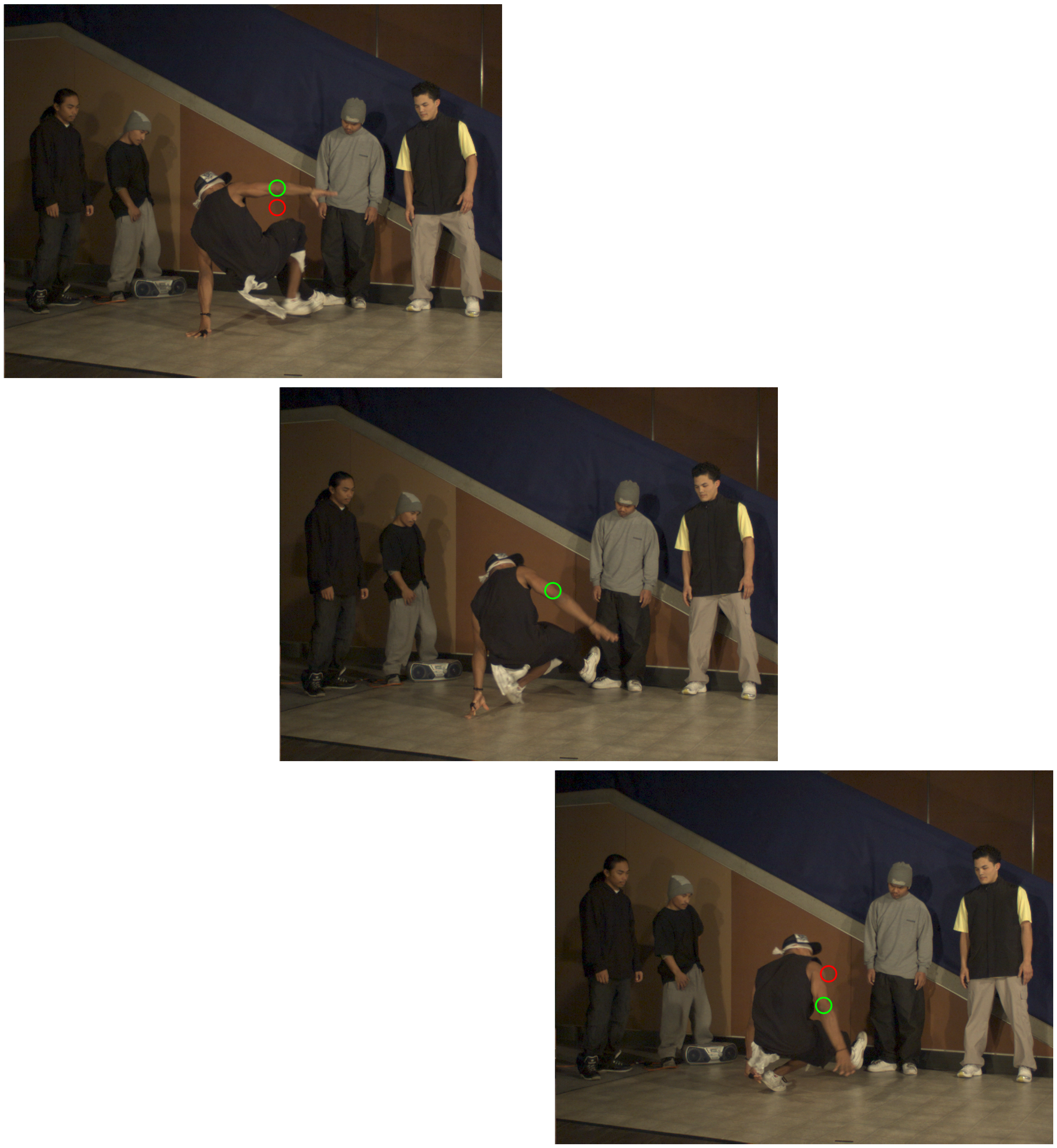


Figure 5.2: The first three frames of the breakancer sequence (Zitnick et al., 2004). Circled are corresponding pixels for the breakdancer's elbow. Note the relatively large motion. For temporal Belief Propagation to work, the three green pixels need to be correctly labeled as temporal correspondences. The red pixels are the ones that would be connected if no motion were assumed (no optical flow used).

Propagation by itself is an attempt to extent the strengths of Belief Propagation to both the temporal and spatial domains, and is discussed and motivated in Section 5.1. By itself, this approach has some weaknesses, with the most significant of these the erroneous neighborhood connections from the automatic naive neighborhood definition. Robust Temporal Belief Propagation is a modification to Belief Propagation which makes it robust to these erroneous neighborhood assignments. Temporal Belief Propagation is the primary motivation for the robustness modifications, but the modifications themselves do not rely on any temporally-specific information. Therefore, the same modifications are investigated when applied to the common strictly spatial domain. The results show that RobustBP is an improvement in this domain also.

The robustness modification proposed involves identifying potential outliers (erroneously connected neighborhoods), and discarding them. This approach is discussed in Section 5.2. Then Section 5.3 shows that Robust Temporal Belief Propagation is a positive improvement over both Temporal Belief Propagation and classic Belief Propagation. Section 5.4 concludes the chapter.

5.1 Motivation

In video sequences common to office environments, it is common for some pixels to change from frame to frame ("changed" as in, they correspond to a different portion of the scene than they did in the previous frame; hopefully this is reflected as a change in color values for the pixel), and for some pixels not to have changed (*e.g.* a wall in the background). Typical causes include sensor noise, high frequencies in the scene (in color or in depth, as in near the silhouette of an object), and objects in the scene moving relatively quickly. Some examples of pixels with a low likelihood of change are those for which the objects in the scene are not moving quickly, as in walls and other stationary objects. Classic Belief Propagation does well at dealing with sensor noise, but other than that, the places that Belief Propagation does poorly (and the places that most common vision algorithms also fail) tend to be fundamentally orthogonal to those classifications. The implication is that, in areas where we classically are doing poorly, some of the situation is different in the adjacent frame, and some of it is the same. If we have the result of some algorithm applied to one frame, intelligent use of that information when applying the algorithm to the next frame poses both some potential improvements and some potential pitfalls. This chapter is about an approach to successfully using multi-frame information to improve the results at each frame.

There are many sources of error in the physical imaging process, e.g. sensor noise, sampling artifacts, etc. One of the strengths of Belief Propagation (and Markov Random Field approaches in general) is the ability to be robust to local spot noise by sophisticated integration of larger scale information.

5.2 Outlier Elimination

The approach taken by RobustBP is to detect when an incoming message is an outlier, and to discard that message in the update computations. Each node re-evaluates each incoming message each iteration. This differs from using inter-pixel color similarities to weaken propagation across color edges. RobustBP makes the implicit "same surface" determination when high frequencies are present in the colors in the scene. In low frequency portions of the image where discontinuities exist, *e.g.* a foreground and background object of the same color, RobustBP can again detect discontinuities due to the differing PDF shapes. Because outlier detection is re-evaluated at each iteration, edge decisions are refined as correct depths are converged upon.

5.2.1 Discarding Detected Outliers

Interestingly, while the trend in algorithms making use of edge detection is to be soft (instead of making concrete yes/no decisions, they make continuously valued decisions, *e.g.* "80% yes") (Felzenszwalb and Huttenlocher, 2004), the trend in robust methods (Fischler and Bolles, 1981) is to make a hard binary decision to discard outliers. This is partly because even a significantly weakened outlier can significantly skew results. Another reason influencing these different approaches is that robust approaches often make the assumption that the right result can be arrived at via a small number of high confidence results (implying that it is acceptable to discard even correct information) while soft decision algorithms are often applied in settings under the assumption that additional information, even if low confidence, can be useful.

My work here follows the trend of the robustness community: to be conservative, *i.e.* discard even if not one hundred percent sure because it is assumed there will remain enough information that the right answer can be arrived at in spite of perhaps mis-classifying some non-outliers as outliers. I work on the assumption that erroneously discarding a non-outlier has small negative effects, while erroneously accepting an outlier has significant negative effects.

5.2.2 Outlier Detection

To address the problems caused by propagating messages between nodes that do not belong to the same surface, I develop a scheme for dynamically adjusting the structure of the graph, limited locally to only disconnecting existing edges. It detects and removes outliers among the incoming messages based on variance reduction (Perrone, 1993). If a message is inconsistent with the others, removing it results in a reduction of total per state variance that can be computed using all incoming messages.

$$R(m_{ij}) = \sum_{k} \sigma_{all}^{k} - \sigma_{all \setminus m_{ij}}^{k} \tag{5.1}$$

If the reduction $R(m_{ij})$ is positive, the edge between nodes i and j is removed from the graph and the node under consideration (j) remains connected with fewer neighbors. It is worth pointing out here that this process applies to all edges, not only the ones connecting temporal neighbors. Due to the small size of the neighborhood, I limit the implementation to remove no more than two edges so that enough information can still flow in cases where noise and edges also restrict the propagations. Figure 5.1 shows the six messages for a point on the elbow of the breakdancer (Figure 5.2), where optical flow has failed. The two temporal neighbors are detected as outliers and removed.

5.3 Results

Results are presented here for applying my robustness approach to Temporal Belief Propagation. In addition, since this technique can equally be applied to still-frame applications, I present results for some of these also, though this is not the focus of this chapter.

5.3.1 Temporal Belief Propagation

Figure 5.3 shows a visualization of which neighbor links got severed via the robustness algorithm. Black indicates no neighbors were dropped, while red and blue indicate horizontal and vertical neighbors were dropped, respectively Green indicates a temporal neighbor was dropped. The two images in Figure 5.3 compare the algorithm without and with (respectively) optical flow, *i.e.* the first image shows the neighborhood connection status when the neighbors are connected with no optical flow estimate, that is to say, an assumption of no motion.

Figure 5.4 shows that this algorithm improves upon both the single frame traditional approach and the result obtained by leaving all temporal connections intact, as assigned by optical flow.

5.3.2 Still-Frame Belief Propagation

Observation of Equation 5.1 reveals that the outlier detection does not include information regarding temporal nodes vs. spatial nodes. Further, the results shown in Section 5.3.1 indicated that in the temporal setting, it is still occasionally best to discard spatial messages. These observations suggest that it may be beneficial to consider applying RobustBP to data for which the nodes are connected entirely spatially, *e.g.* in classic still-frame stereo. To that end, I present results here of RobustBP on the Middlebury (Scharstein and Szeliski, 2002; Webpage: Stereo, 2007) datasets, in Figure 5.5. These results show that RobustBP, while not a perfect outlier detector in these cases, is still a positive improvement. Further results are shown in Chapter 7.

5.4 Summary

In this chapter, I have shown that outlier messages are present in common 2D Belief Propagation, and is an even more significant problem in Temporal Belief Propagation. Further, I have shown a robustness algorithm that significantly improves the results in these settings. The outlier detection method chosen may not be the best method, but the results suggest that the approach of robustness in Belief Propagation can be a fruitful path for future research.

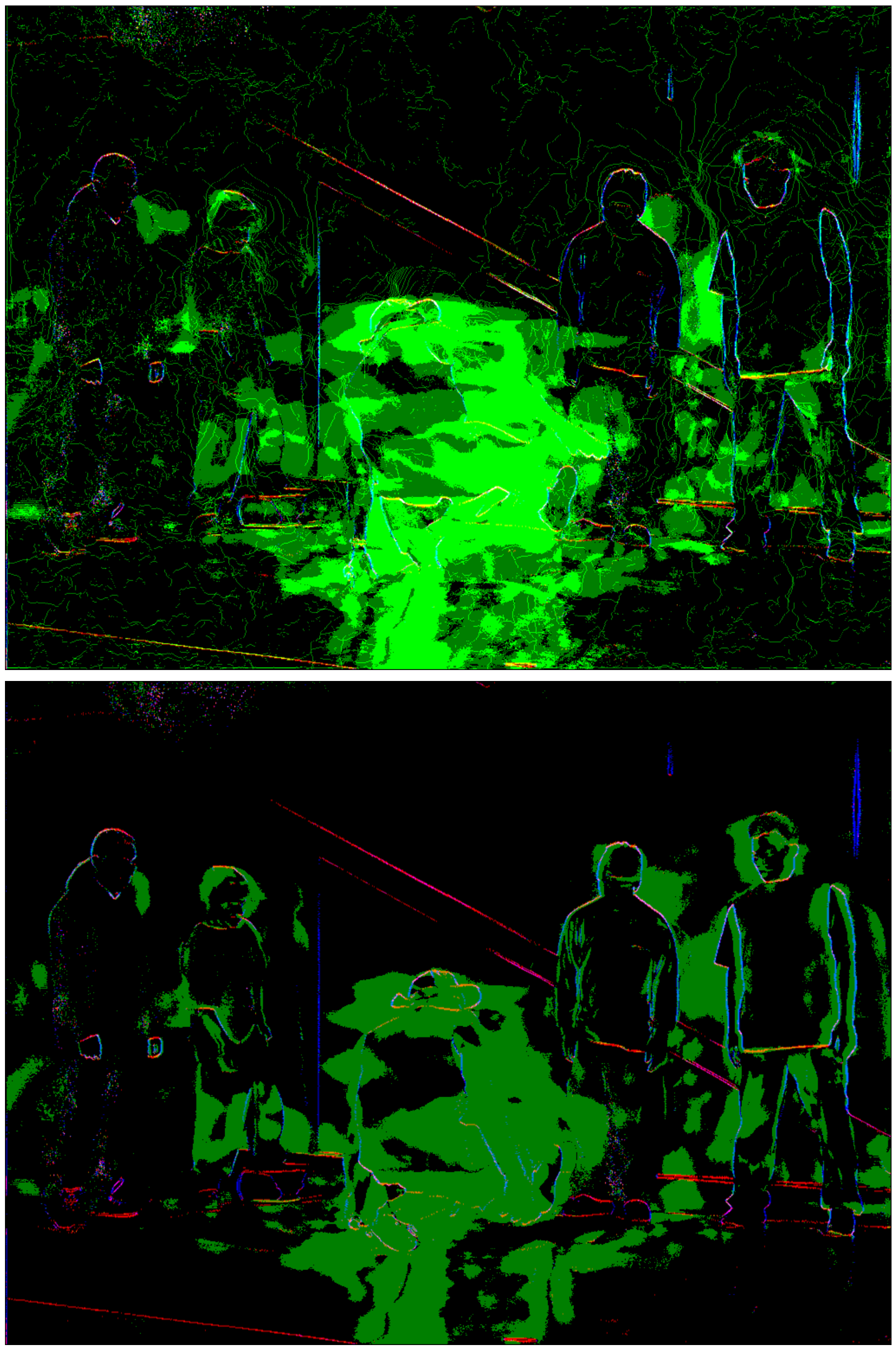
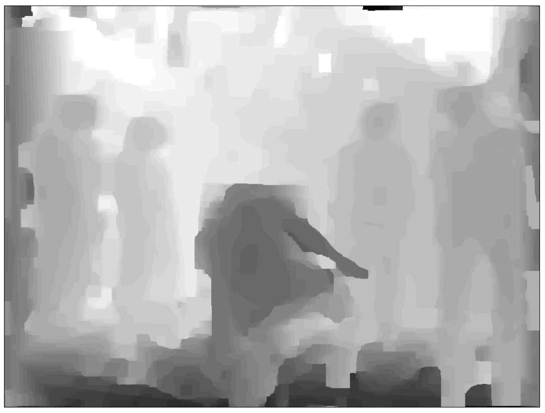
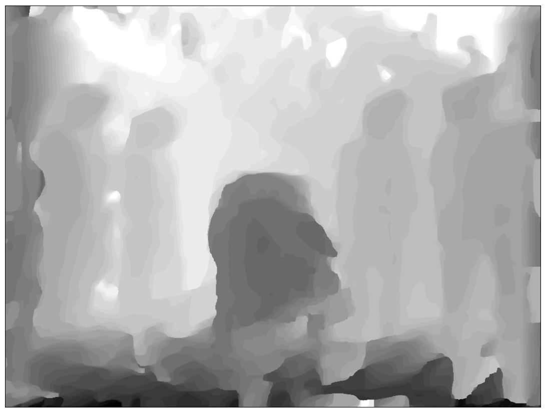


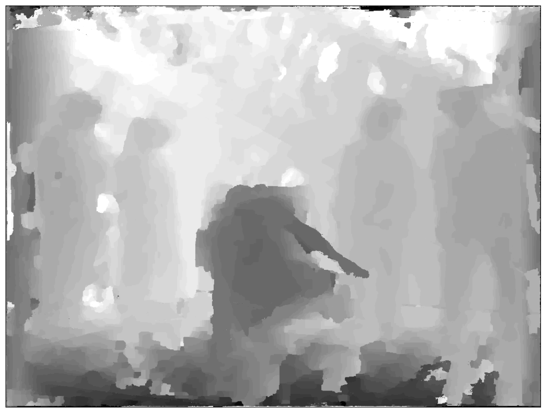
Figure 5.3: Visualizations of Robust Temporal Belief Propagation in detail, for the second frame of the breakdancer sequence. The top image uses no optical flow (e.g. straight through, assumes no motion), while the bottom uses the optical flow algorithm of (Proesmans et al., 1994). The non-black pixels indicate which neighbor nodes are disconnected by my robustness algorithm. Observe that less temporal neighbors are disconnected with a better optical flow estimate.



(a) Single Frame Belief Propagation



(b) Naïve Temporal Belief Propagation using three frames



(c) My RobustBP with three frames

Figure 5.4: Comparison between three approaches (top to bottom): traditional Belief Propagation on the central frame, my Robust Temporal Belief Propagation applied when the previous and next frames are connected using optical flow, and naïve Temporal Belief Propagation using the previous and next without my robustness improvements. Note the dancer's extended arm and that, in this case, my approach is an improvement over both other algorithms.

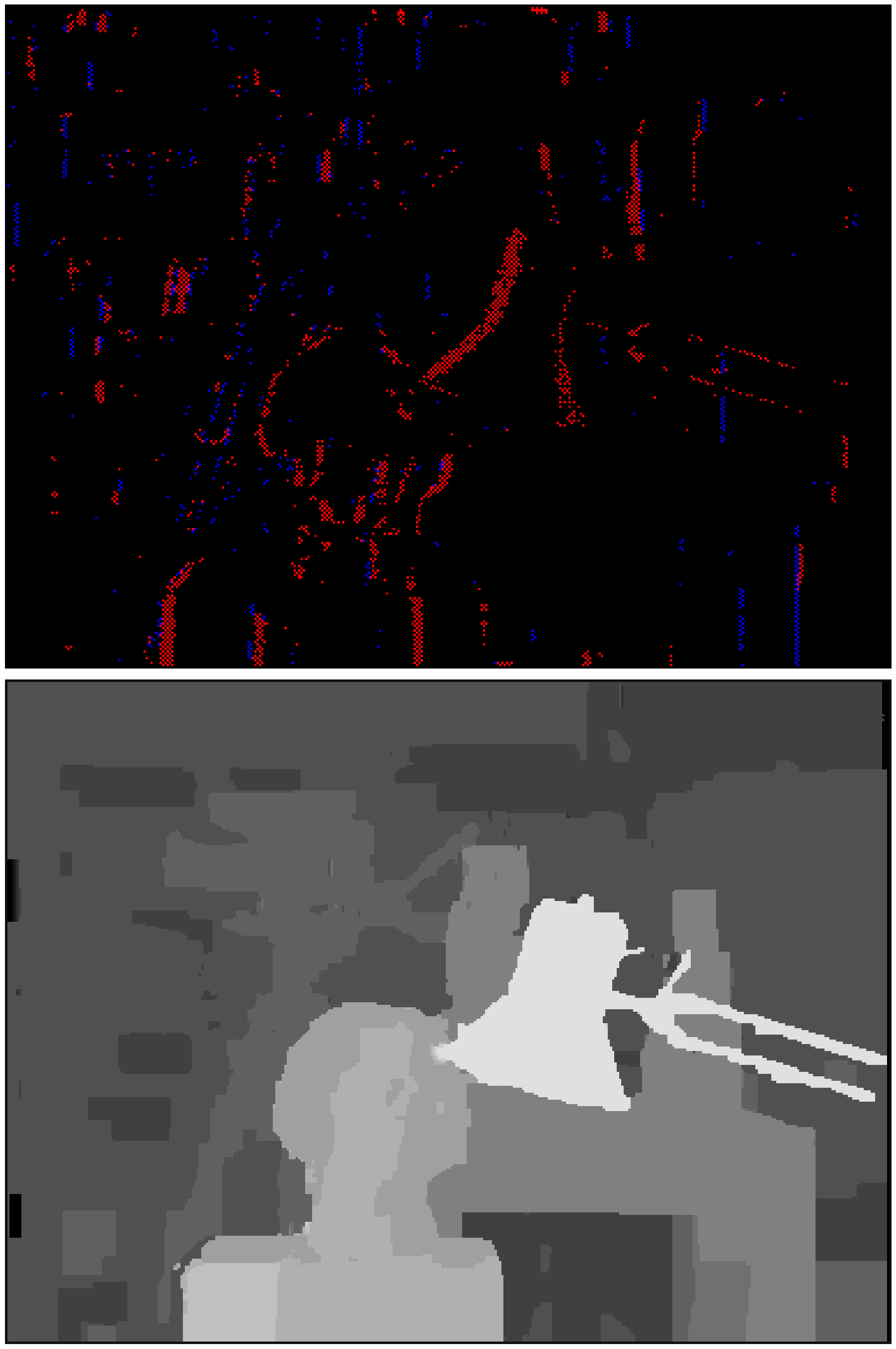


Figure 5.5: Similar to the results in Section 5.3.1, a visualization of which connections were deemed outliers and the final result. Again, the non-black pixels in the top image show the detected outlying connections while the bottom image shows the final depth result. Note that the outliers align with depth discontinuities in the image, but that they are not necessarily nice clean edges.

Chapter 6

Quiet Belief Propagation

Belief Propagation is commonly considered insensitive to noise. A strength of BP is the ability to extract "signal" from noisy input, but this strength becomes a weakness when the input is dominated by noise. I present here "QuietBP" for its improved behavior in the presence of noise. In this chapter, I

- show that in a synthetic example, "loopy BP" (Belief Propagation in a graph containing cycles) (Yedidia et al., 2001) can be *very* sensitive to noise,
- show that this sensitivity extends outside synthetic examples to common computer vision datasets,
- present a solution to this, and show result improvements on common computer vision datasets.

The first two of these are detailed in Section 6.1. My solution is presented and discussed in Section 6.2. Section 6.3 shows that this technique noticably improves BP results on common computer vision datasets.

6.1 Motivation

I begin with motivation via a common computer vision real-world dataset: the break-dancers dataset from the Interactive Visual Media Group at Microsoft Research (Zitnick

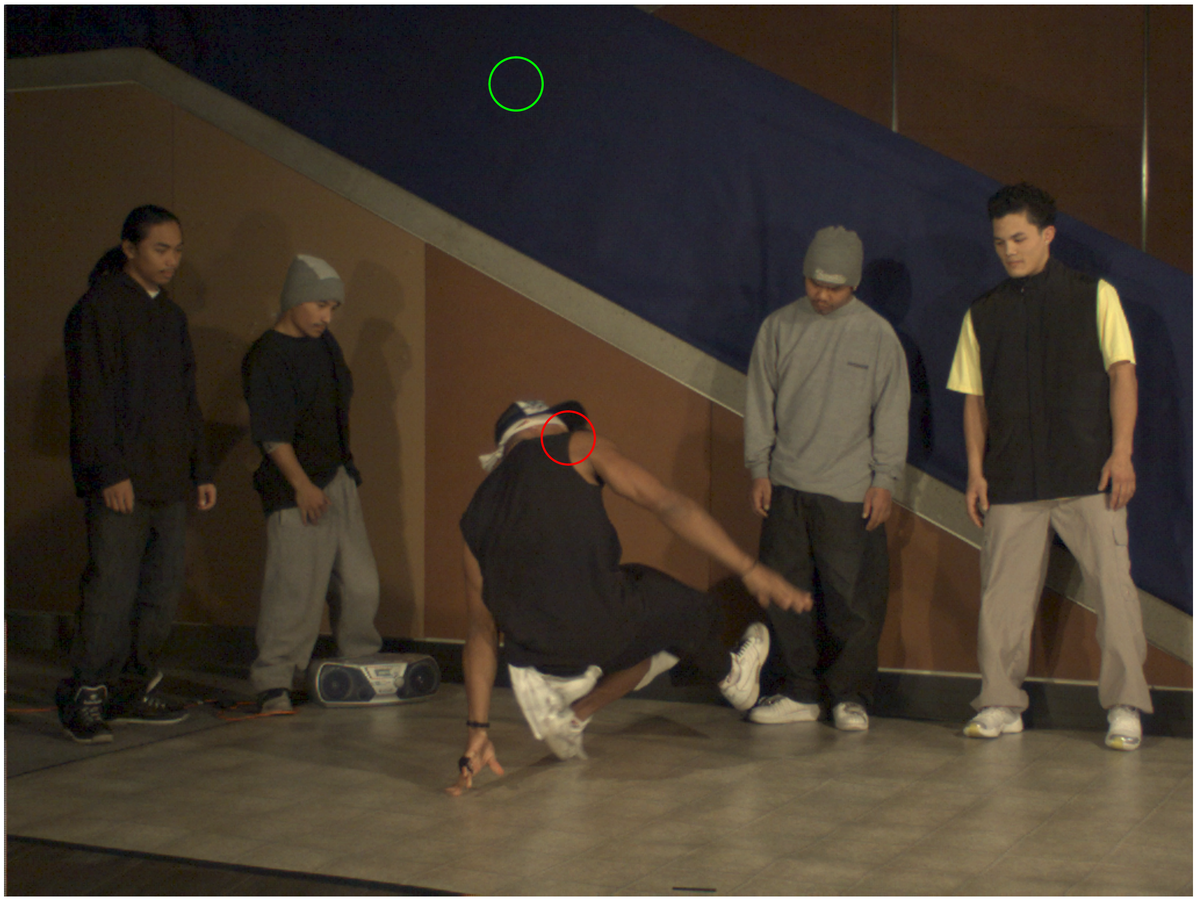


Figure 6.1: The color reference image used for the examples in this section. Figure 6.2 refers to the pixels highlighted by the green and red circles, contrasting the strength of the signal to noise ratios between those two sections. Specifically, the sensor noise in the black textureless background areas is of significant relative magnitude compared to the black, while the sensor noise on the brighter foreground areas with high contrast have much smaller relative magnitudes.

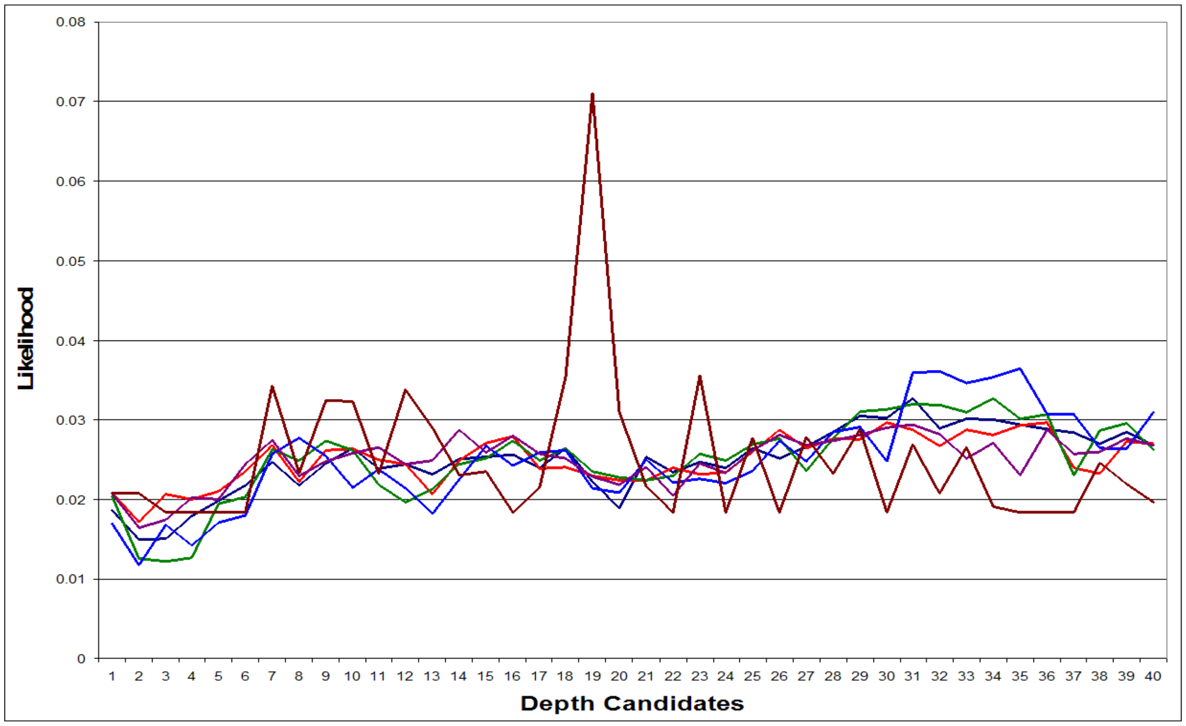


Figure 6.2: Comparison of *PDF*s for a pixel on the back wall and its four neighbors, location indicated by the red circle in Figure 6.1, and the *PDF* for a pixel in the foreground, location indicated by the green circle in Figure 6.1. Observe that, in contrast to the pixel on the dancer's shoulder, the *PDF*s on the back wall have very little confidence and the magnitude of the noise is nearly as significant as the peaks themselves.

et al., 2004). Figure 6.2 shows the observation *PDF*s of a pixel on the back wall, along with those observation *PDF*s of its four neighbors. From this graph, *we* as humans can see that the peak ought to form near depth candidate “34,” and it appears that it is getting sufficient support there. But noticing the scale and re-plotting these in comparison to the observation *PDF* for a pixel that is getting a good match, as in Figure 6.2, it can be seen that

- the *PDF* strength in the correct depth is very weak compared to the strength of the noise.

This produces the “mottled” effect seen in the highlighted portion of the back wall in Figure 6.7.

6.1.1 A Synthetic Exploration

To better understand what the problem is, I present a simple synthetic example. This is actually a family of specific examples. In each, a single node in the center of many has a perfect impulse *PDF*, for the central state. This *PDF* looks like that shown in Figure 6.3. All of the other nodes in the field are initialized with a uniform *PDF* slightly altered by noise, as illustrated in Figure 6.3. The noise is bounded by ϵ. Note that the specific effect is randomly different for each node An illustration of an example run is shown in Figure 6.5.

If ϵ were zero: no noise, then the impulse properly propagates throughout the entire field. When ϵ is greater than zero, one would hope that the same behavior occurs, yet it does not. By controlling ϵ, this synthetic example can give us insight into the effect of noise on Belief Propagation, and we can verify the “sanity” of our solution.

Each specific instance in this family of synthetic examples is selected so as to illuminate the effect of the “loopiness” of the node network. Each is illustrated in Figure 6.4. In order from simplest to “loopiest”, they are:

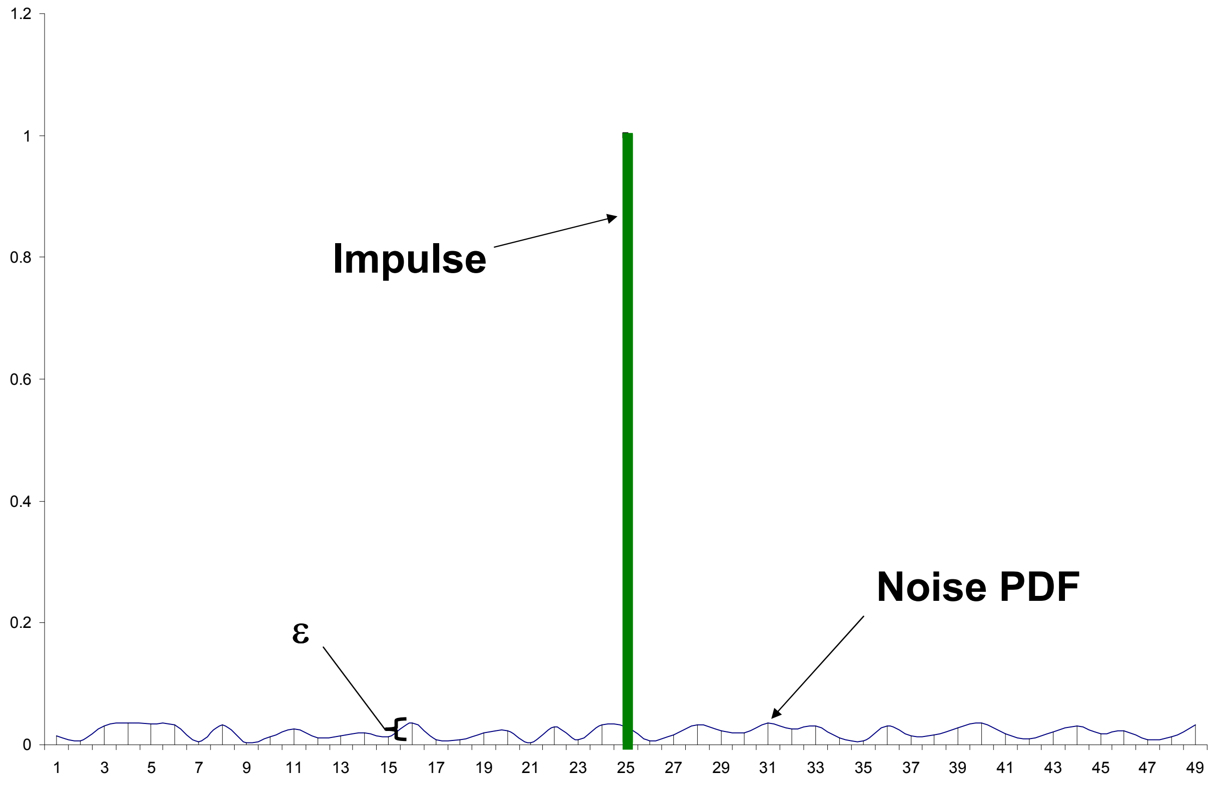


Figure 6.3: The perfect impulse *PDF* used to initialize the central node in the synthetic examples, and a sample noisy *PDF* illustrating the initialization for all the rest. All the rest of the nodes have their own randomly noisy *PDF*, with noise modulated by ϵ, of which this is an illustrative sample.

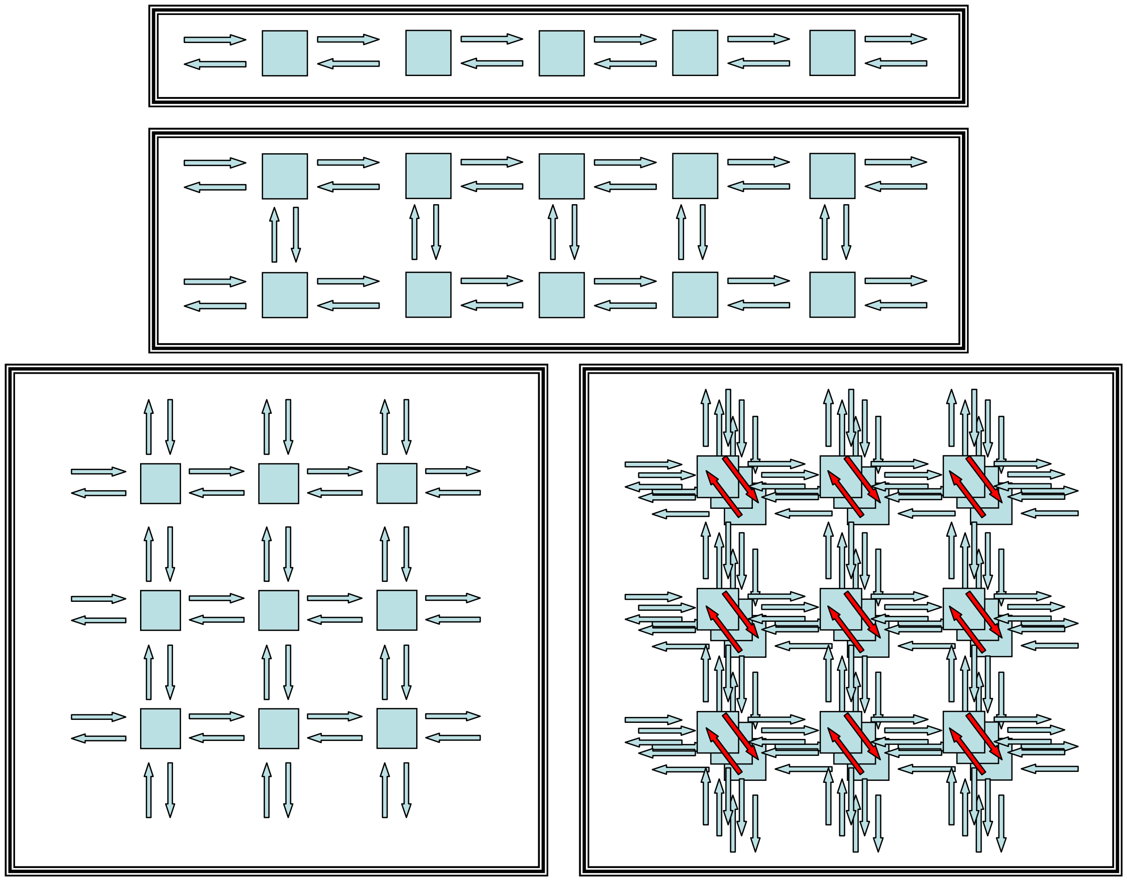


Figure 6.4: Node network diagram for the synthetic examples. Note the increasing number of looping neighbors in each (the difference between the second and third is that the second has all of its nodes on boundaries, so each has only three neighbors). For each, a single node in the center is initialized with a perfect impulse *PDF* (illustrated in Figure 6.3), and all the rest are initialized with random perturbations of a uniform *PDF*, with the effect of the noise modulated by ϵ. Figure 6.3 illustrates a sample.

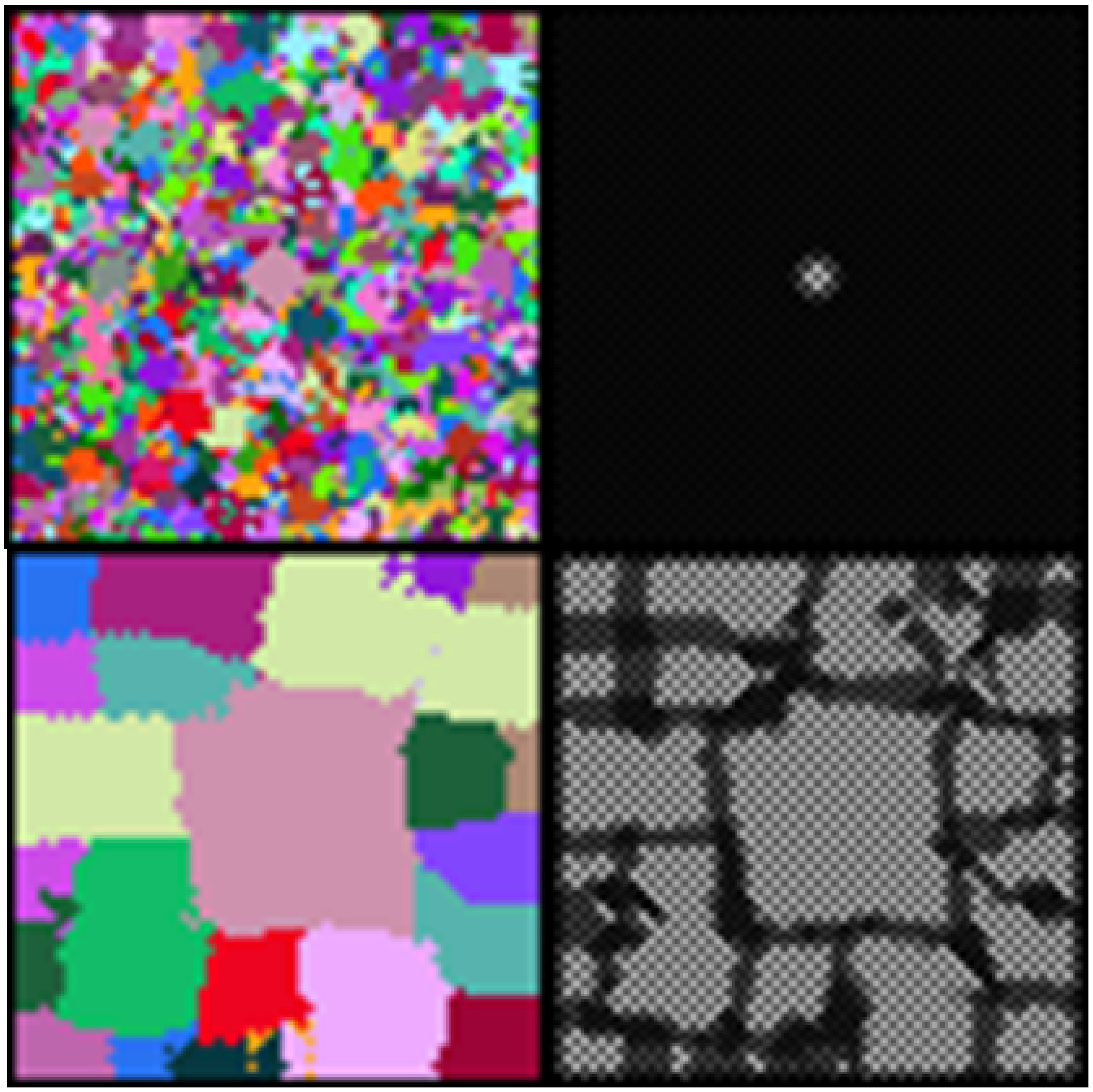


Figure 6.5: Rendering of the progress of an impulse in a 2D field of noise. The top row is after the very first iteration, the bottom row is after convergence. The left column is a pseudo-coloring showing the most likely depth at each pixel. The right column is a scaled gray-scale image showing the likelihood of that depth. Note that after the first iteration, the perfect impulse is clearly seen on the right, and that the most likely depth for all pixels is fairly random. After convergence, this perfect impulse has propagated only a limited distance: other depths have built up sufficient confidence that they limit its propagation. This is seen by the confidence magnitudes in the lower right image. Figure 6.6 shows that the extent that the impulse travels is a function of ϵ.

1. a simple 1D row of nodes: no loops,
2. two rows, close to 1D but with a simple loop on three neighbors,
3. a full 2D field: loops with all four neighbors,
4. three 2D fields aligned with each other: the boundary fields are close to the 2D field example in loopiness (a little more loopy) and the center field has loop neighbors on six sides now,
5. a full 3D field: all nodes have loops with six neighbors (*i.e.* like our video sequence setting, with four spatial neighbors and two temporal neighbors).

If Belief Propagation were completely in-sensitive to noise, then the single impulse will influence all the nodes in the entire network. With $\epsilon \geq 0$ though, the impulse does not always propagate fully through the full extent of the field. Figure 6.6 shows how far the impulse tends to travel in each case. For that figure, the maximum possible propagation extent is 50. 500 iterations of BP were run for each trial. For each setting of ϵ, there were 100 trials run, re-randomizing each time. After each run, the propagation along each major axis was determined by stepping out from the center and determining if the most likely state was still the state that the impulse was in. This counting was done in both directions along each major axis, and the results were averaged together across all runs. An axis is considered major if it has more than three nodes along it (*e.g.* in the case of two rows, there is one major axis: along the row).

Explanation

A simple explanation for the limited propagation of the impulse in the presence of noise is presented here. The key observation is that the network contains loops. This has the effect that messages from one node eventually return back to it and reinforce any portion of the *PDF*s that were somewhat consistent with neighbors. With uniform noise, there

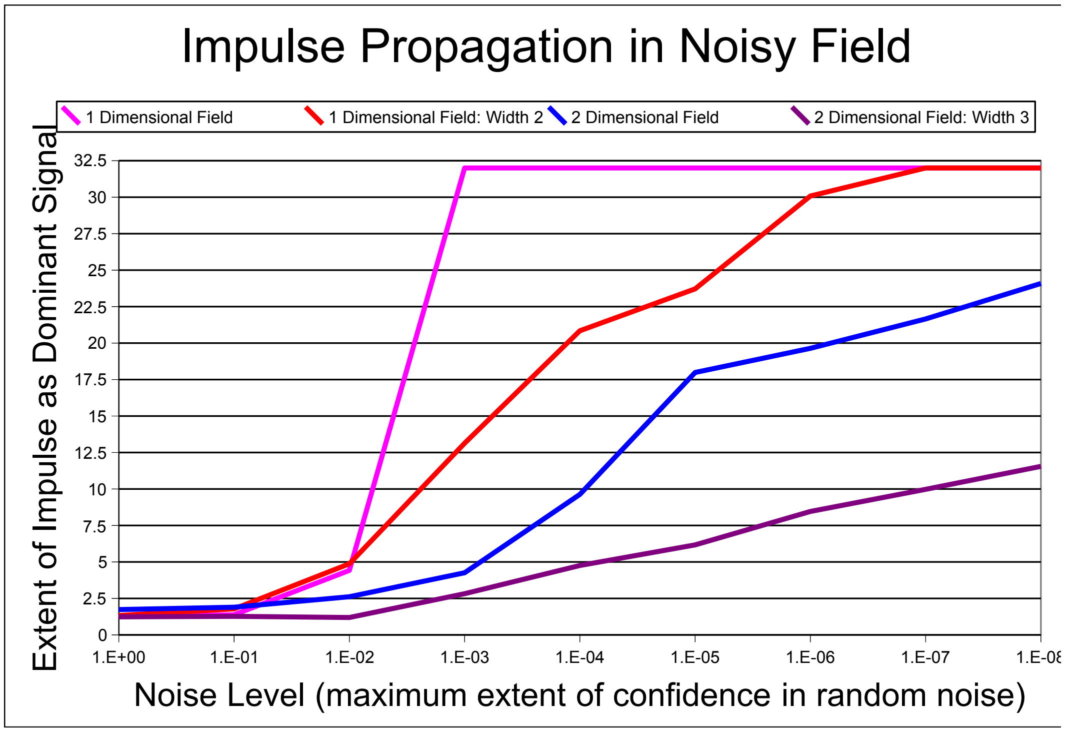


Figure 6.6: The extent that BP propagated the impulse *PDF*, as a function of the noise level. Note the logarithmic scale across the bottom. Ideal propagation is 50 (to the boundaries of the 101x101 mesh, with the impulse in the center). Observe how increased loopiness reduces how far the impulse gets successfully propagated.

is some chance that some state will receive slight support in some number of nodes in a neighborhood.

Belief Propagation extracts and amplifies signal in the presence of noise. If the noise is dominated by signal, than the noise gets reduced and the signal gets cleaner and stronger. *But*, if there is *no* signal there – as in the uniform case – or the signal is not strong enough – as in the real world cases – then Belief Propagation actually extracts and amplifies noise that is relatively strong and also consistent with neighbor messages. In the real-world case, this effect is even stronger than in the artificial case because the noise is a regular function of sensors and sampling so noise in one node is actually likely to re-occur in neighbors, *i.e.* noise in real-world cases is not statistically independent.

The limited propagation of the impulse is due to erroneous states building up enough support, through each iteration, that when the true signal arrives, the support for the true signal is not strong enough to outweigh the artificial support built up due to noise. This effect can be seen quite clearly in Figure 6.7: Near the diagonal features in the back wall, there is a radius of correct values, but in-between these, there are patches that reflect strength in erroneous depth values, *i.e.* the strong signal at the features on the wall are limited in their propagation by support for noise in the near-uniform regions.

6.2 Proposed Solution

Given the energy equation

$$E(x) = -\ln \sum_{i} \phi_i(x_i) - \ln \sum_{(ij)} \psi_{ij}(x_i, x_j), \tag{6.1}$$

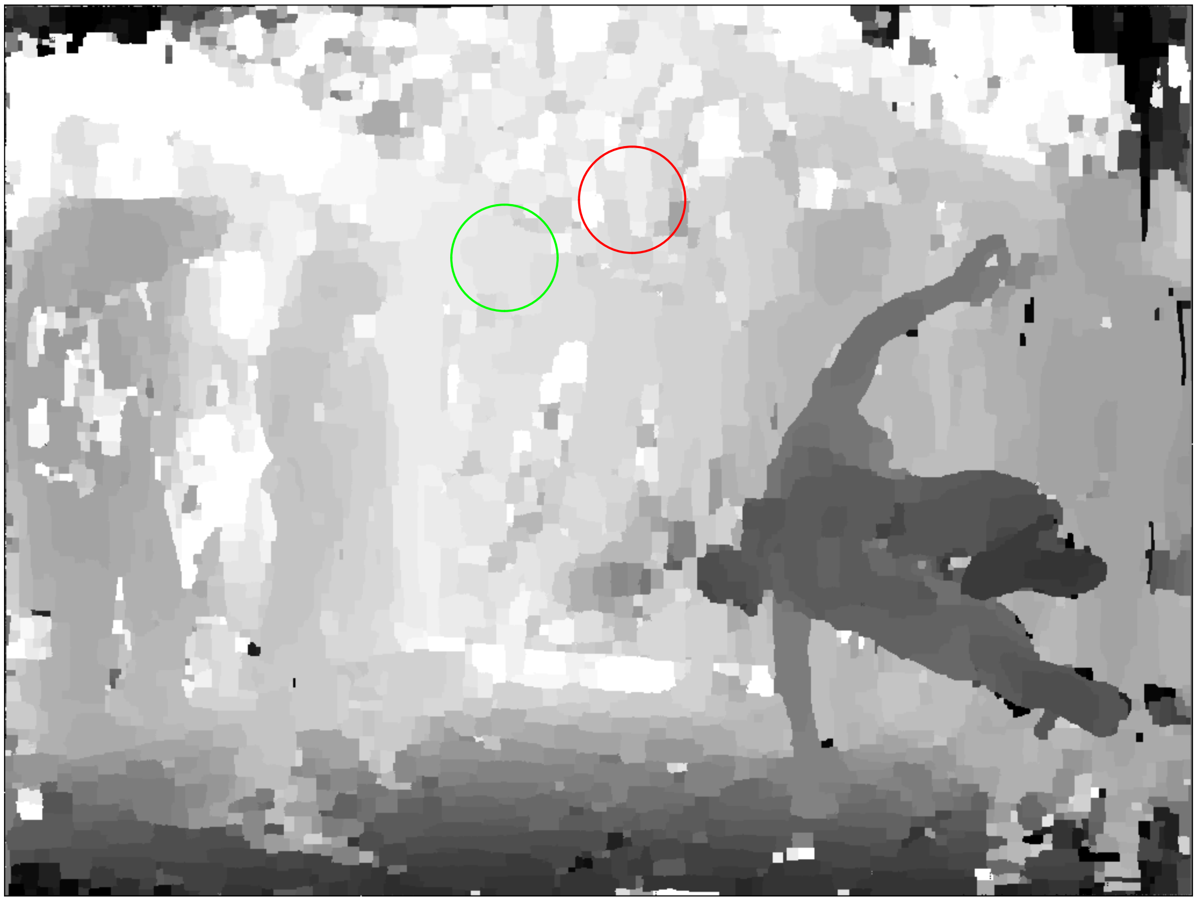


Figure 6.7: Depth map using regular BP. Notice the limited spatial influence of the features along the diagonals on the back wall (the green circle highlights one example). Compare this to the mottled effect seen in the near-uniform regions, as in the red circle.

one using Belief Propagation would commonly solve this by initializing all outgoing messages to uniform PDFs and then using the update rule

$$m_{ij}^{s}(x_j) \leftarrow \max_{x_i} \psi_{ij}(x_i, x_j)\phi_i(x_i) \prod_{k \in \mathcal{N}(i) \backslash j} m_{ki}^{s-1}(x_i). \tag{6.2}$$

The loops in a simple 2D field reinforce the observation (ϕ) due to its being a component in all outgoing messages, which later have traversed loops and return, to reinforce the observation in future updates.

One approach to reducing this may be to analytically determine how to factor the observation out of incoming messages if they contain traces of it. Equationally, following the observation term through a simple 2D loopy field for a few iterations suggests to me no places where a simplification may be made that would allow for the residual observation (after having traversed a loop) to be identified without factoring in all of the messages at each step along the path of the loop. The complexity of this equation grows exponentially, both in space and in computation.

The proposed solution is to instead run BP by first initializing all outgoing messages to be identical to the observation, *i.e.*

$$m_{ij}^{0} x_j \leftarrow \phi_i(x_i), \tag{6.3}$$

and then using the update equation

$$m_{ij}^{s}(x_j) \leftarrow \max_{x_i} \psi_{ij}(x_i, x_j) \prod_{k \in \mathcal{N}(i) \backslash j} m_{ki}^{s-1}(x_i). \tag{6.4}$$

Update rule 6.4 is identical to 6.2 except that ϕ is completely removed, *i.e.* the observation is used only to initialize, and is ignored in subsequent updates.

The effect of this approach is that the observation does not act as a partial "filter" to future message updates. One could conceive of a drawback of this being that messages from far away that are different can come and “over-run,” because the observation is not restricting it. In reality though, the observation is locally re-enforcing itself through the local loopiness. In practice, I have seen only positive results using this technique. The next section delves further into the results on real-world datasets.

6.3 Results

Here, I present results showing that my solution for dealing with BP’s noise issue makes a positive contribution to the results on computer vision datasets. I show comparisons and analysis for regular stereo vision via the Middlebury (Scharstein and Szeliski, 2002) datasets. I then show improvements in multiview reconstructions on the breakdancer (Zitnick et al., 2004) data. Finally, I show that the effect is amplified in a 3D network, described in Section 6.3.3.

6.3.1 Middlebury Results

Figure 6.8 shows a comparison of the depth maps for the datasets used in the Middlebury rankings. It clearly shows that this technique improves the results. Due to the availability of gold-standard reference results on this dataset, it is possible to quantitatively compare the results obtained using this technique.

6.3.2 Breakdancer Results

Figure 6.9 compares the results on the first frame of the breakdancers (Zitnick et al., 2004). It shows that using this method significantly improves the large, near-uniform regions, while not negatively affecting anything else.

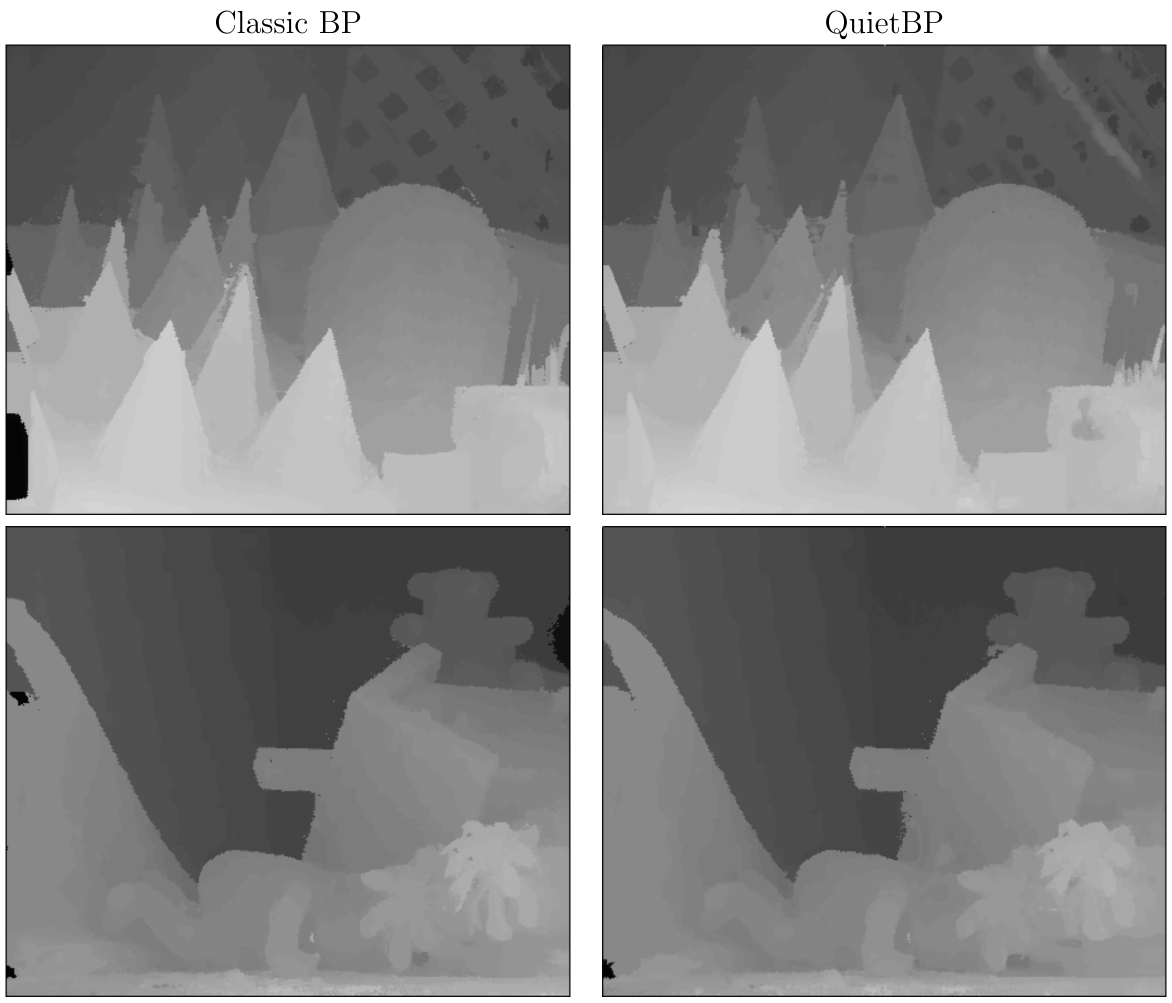
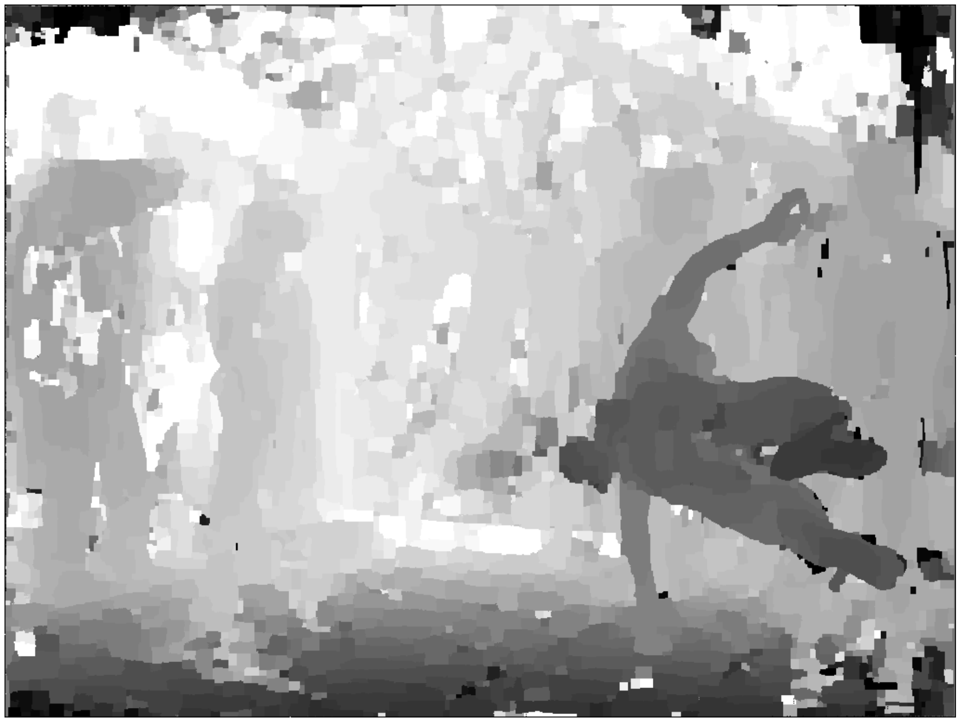


Figure 6.8: Comparison for the Cones and Teddy datasets from (Scharstein and Szeliski, 2002). The left column shows results obtained without using QuietBP. The right column shows the results obtained with everything identical, including all parameters, and with QuietBP enabled.

Classic BP



QuietBP

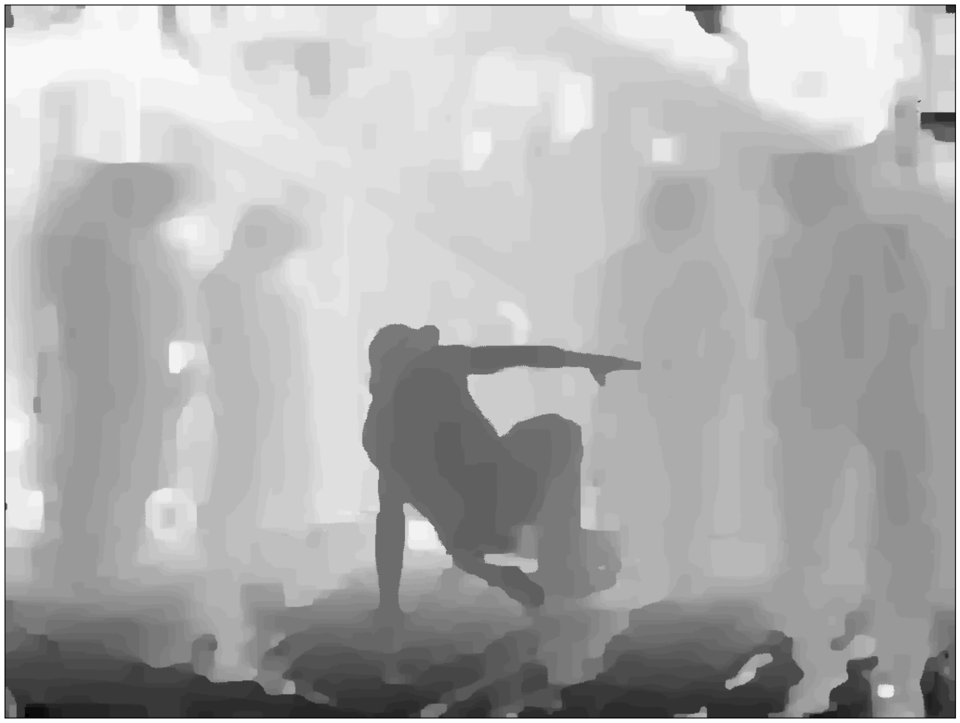
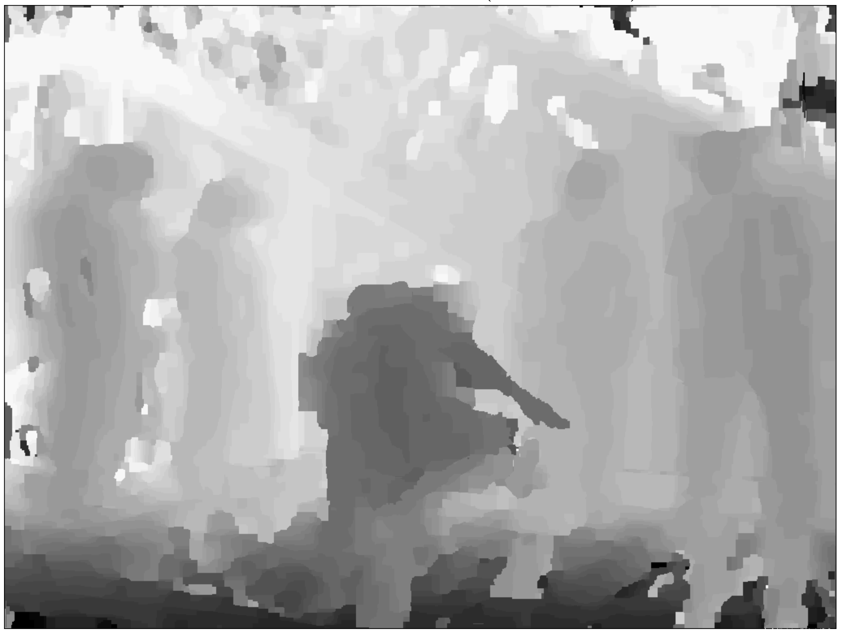


Figure 6.9: On the top is the result on the breakdancer dataset obtained without using QuietBP. The bottom is the result obtained on a similar frame with QuietBP, with all else identical, including all parameters.

RobustTemporalBP (Chapter 5)



RobustTemporalBP with QuietBP

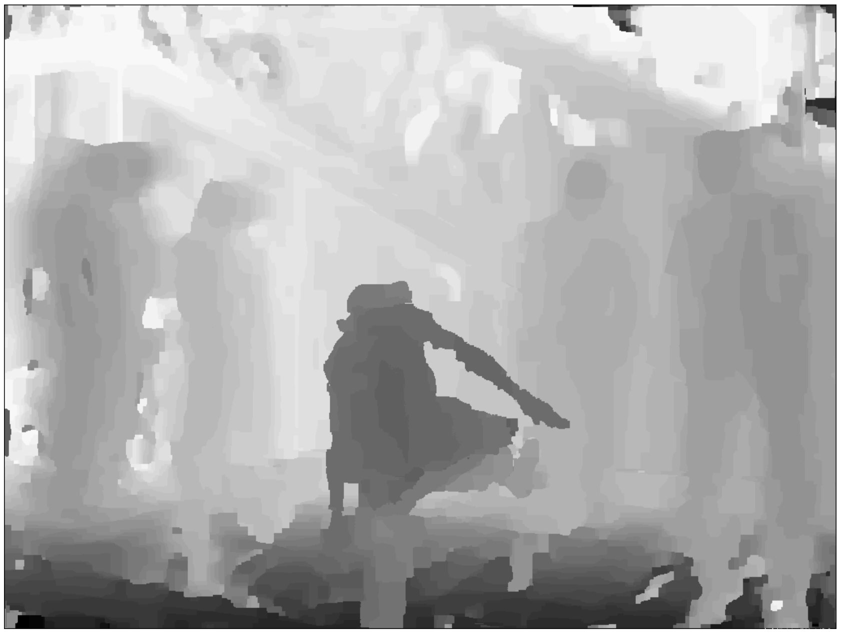


Figure 6.10: The second frame of the breakdancer dataset evaluated using the previous and next temporal frames as direct node neighbors with no optical flow (more advanced treatment is in Chapter 5). Nodes in the center temporal frame (the one shown) have six neighbors now, instead of four, and all six contribute loopiness. On the top is the result on the breakdancer dataset obtained without using the technique of this chapter. The bottom is the result obtained with everything identical, including all parameters, and also with QuietBP enabled.

6.3.3 Results on a 3D Node Network

Chapter 5 discusses 3D node neighborhoods. It uses them to pass messages between temporal frames in addition to spatial neighbors. The results in that chapter depend on the technique of this chapter. In that chapter, I take a much more advanced approach to neighborhood relations between frames, but it is sufficient for this chapter to show results that illustrate the use of QuietBP versus not using it. Figure 6.10 shows two depth maps, one not using this technique and one using this technique. In the portions of the image that motion is occurring, the results are poor - Chapter 5 address those issues. For this chapter, the focus is on the large, near uniform regions that are static, *e.g.* the back wall that does not move. Notice, in comparison to Figure 6.9, that 3D neighborhoods actually *worsen* the results in these large nearly uniform regions. Again, this is due to the increased loopiness that 3D node networks bring. Addressing that is the focus of this chapter, and Figure 6.10 shows that this objective has been accomplished.

6.4 Summary

In this chapter, I identified a weakness of Loopy Belief Propagation, demonstrating it in real-world application examples. The more loopiness in a network, the more the negative effect is seen. I then presented a solution, and showed that it is effective in reducing the effect in the same real-world examples. The effect is most dramatic in the TemporalBP case detailed in Chapter 5 as the loopiness there is prohibitively high.

Chapter 7

EnhancedBP

In this chapter I present the combination of the methods of this dissertation. Enhanced BP, in Section 7.1, combines the methods of combines BiasedBP from Chapter 4, RobustBP from Chapter 5, and QuietBP from Chapter 6. Section 7.2 details the computing environment used and resources needed for the algorithm to run on these datasets. Section 7.3 details the parameter settings used, and Section 7.4 presents the results of the combined algorithm on a variety of datasets.

7.1 Combined Models

As presented in Chapter 1, typical Belief Propagation works on the energy equations given by:

$$E(x) = -\ln \sum_{i} \phi_i(x_i) - \ln \sum_{(ij)} \psi_{ij}(x_i, x_j) \tag{7.1}$$

by initializing all outgoing messages to uniform PDFs and then using the update rule

$$m_{ij}^{s}(x_j) \leftarrow \max_{x_i} \psi_{ij}(x_i, x_j)\phi_i(x_i) \prod_{k \in \mathcal{N}(i)\backslash j} m_{ki}^{s-1}(x_i). \tag{7.2}$$

As in QuietBP from Chapter 6, initialization is now done via

$$m^0_{ij} x_j \leftarrow \phi_i(x_i), \tag{7.3}$$

and the update equation

$$m^s_{ij}(x_j) \leftarrow \max_{x_i} \psi_{ij}(x_i, x_j) \prod_{k \in \mathcal{N}(i) \backslash j} m^{s-1}_{ki}(x_i) \tag{7.4}$$

is used for updates.

Adding the temporal neighbors, given by (Proesmans et al., 1994), I use the robustness metric and decision algorithm as given in Chapter 5:

$$R(m_{ij}) = \sum_k \sigma^k_{all} - \sigma^k_{all \backslash m_{ij}} \tag{7.5}$$

BiasedBP from Chapter 4 adds a bias term, making the update equation used by EnhancedBP:

$$m^s_{ij}(x_j) \leftarrow \max_{x_i} \psi_{ij}(x_i, x_j) \omega^{s-1}_i \theta_i(x_i) \prod_{k \in \mathcal{N}(i) \backslash j} m^{s-1}_{ki}(x_i). \tag{7.6}$$

7.2 Computing Environment

All results presented in Section 7.4 were obtained using an AMD ™64 X2 Dual Core Processor 4400+, with 4GB of RAM. Microsoft ®Windows ™64bit Edition was used to enable usage of all the available memory. Only 40 depth steps could be taken due to the memory requirements and only having 4GB of RAM (40 depth steps required 3.8GB of memory). 150 iterations of EnhancedBP were run – the 150 iterations were verified many times to be sufficient for convergence on all the datasets. For the datasets with video sequences (all but the Middlebury ones), 100 frames of video were used. For the Janine dataset, video from one camera was discarded due to its significant color mis-calibration.

Each single iteration, on the 1024x768 resolution frames in the largest sequences, of the combined algorithm took 15 minutes per frame. The code was highly optimized and parallelized. Making the initial observation, via the algorithm of (Yoon and Kweon, 2005), required 270 minutes per frame. As this was a significant portion of compute time, I implemented that algorithm so it would run on the graphics card in Cg ™. The graphics card is an NVIDIA®GeForce®7800GT. The algorithm on this hardware required only 3.8 minutes per frame, making the entire project practical.

7.3 Parameter Settings

The observations were done with the algorithm of (Yoon and Kweon, 2005) using a window size of radius 15. All color parameters were set to 40, as color was measured in the range of 0 to 255. This was not optimized.

7.4 Comparisons

I begin with investigating how well EnhancedBP performs on common binocular stereo datasets. In particular, Figures 7.1 and 7.2 illustrate performance on the Middlebury (Scharstein and Szeliski, 2002) datasets. As mentioned in Chapter 1, static binocular stereo was not the focus of this research, but has the benefits of being one that many other researchers have focused on and also having known gold standard results for comparisons.

Figure 7.3 compares EnhancedBP with the state of the art method of (Yoon and Kweon, 2005). Note that it outperforms both in the most difficult portions of the scene and in rest of the scene. Unfortunately, ground truth is not available for this dataset, so our improvement is measured subjectively with these images.

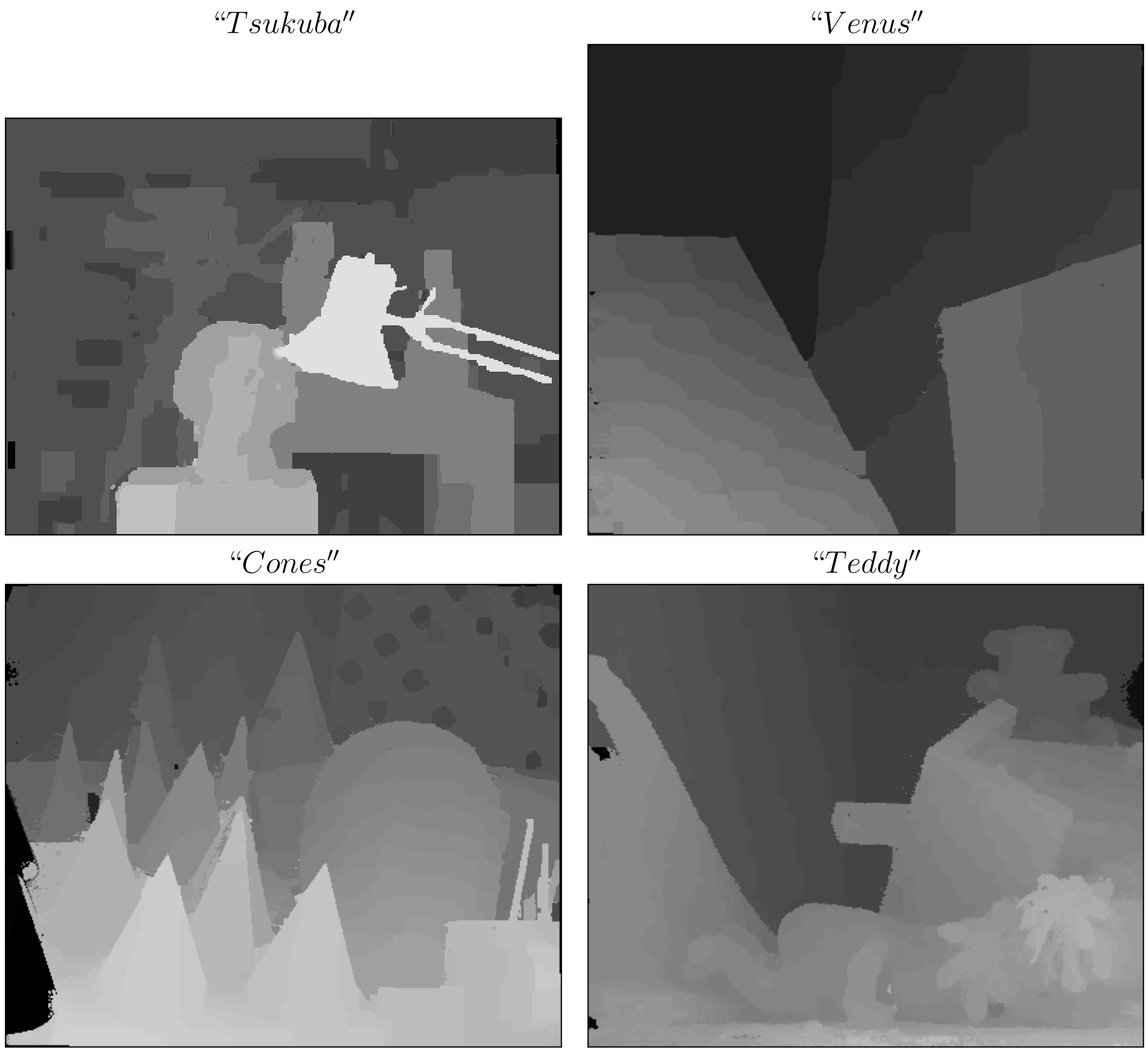
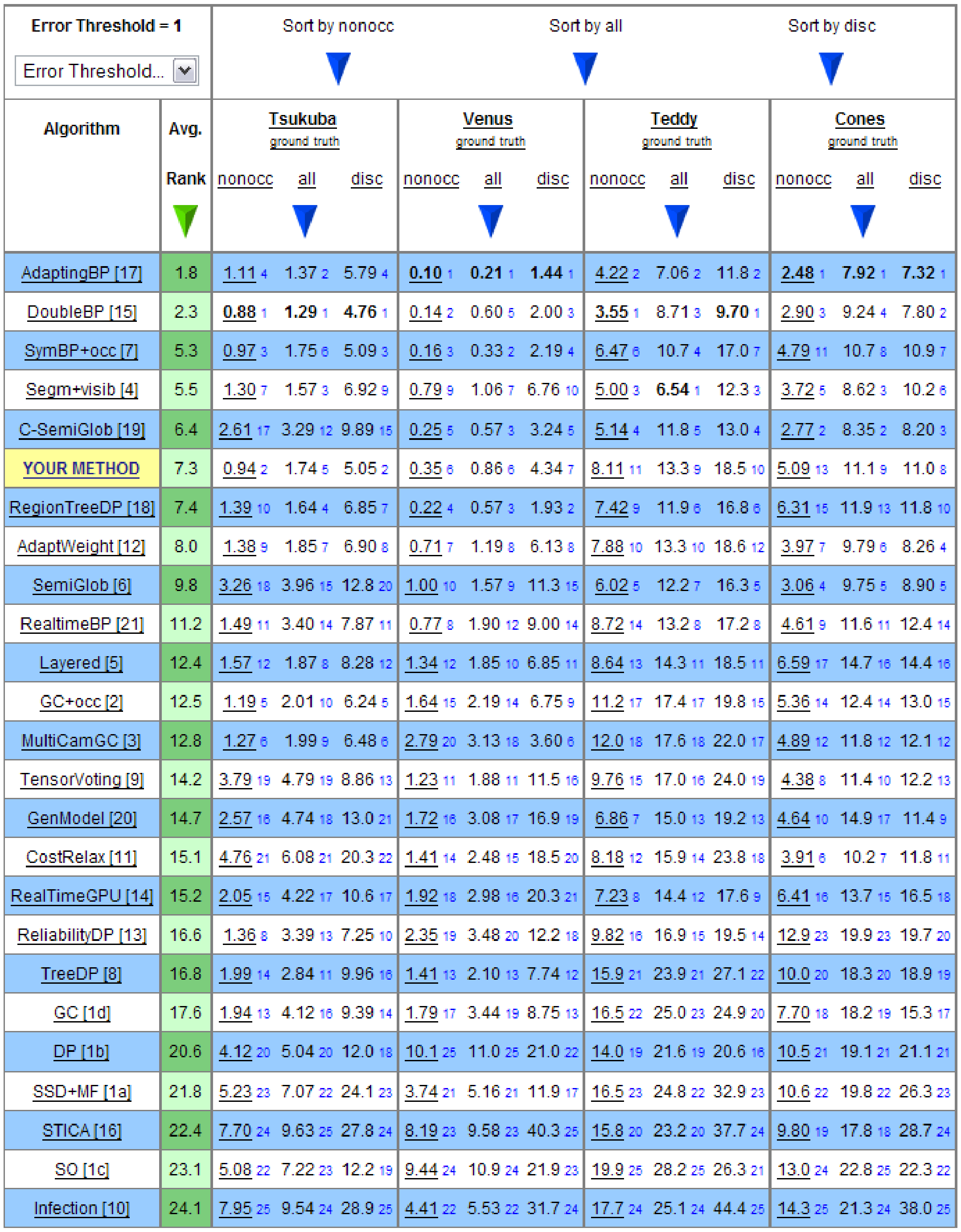


Figure 7.1: Results obtained using EnhancedBP on the Middlebury (Scharstein and Szeliski, 2002) datasets.



Error Threshold = 1

Error Threshold...

Sort by nonocc | Sort by all | Sort by disc

Algorithm	Avg. Rank	Tsukuba ground truth			Venus ground truth			Teddy ground truth			Cones ground truth		
		nonocc	all	disc	nonocc	all	disc	nonocc	all	disc	nonocc	all	disc
AdaptingBP [17]	1.8	1.11 4	1.37 2	5.79 4	**0.10** 1	**0.21** 1	**1.44** 1	4.22 2	7.06 2	11.8 2	**2.48** 1	**7.92** 1	**7.32** 1
DoubleBP [15]	2.3	**0.88** 1	**1.29** 1	**4.76** 1	0.14 2	0.60 5	2.00 3	**3.55** 1	8.71 3	**9.70** 1	2.90 3	9.24 4	7.80 2
SymBP+occ [7]	5.3	0.97 3	1.75 6	5.09 3	0.16 3	0.33 2	2.19 4	6.47 6	10.7 4	17.0 7	4.79 11	10.7 8	10.9 7
Segm+visib [4]	5.5	1.30 7	1.57 3	6.92 9	0.79 9	1.06 7	6.76 10	5.00 3	**6.54** 1	12.3 3	3.72 5	8.62 3	10.2 6
C-SemiGlob [19]	6.4	2.61 17	3.29 12	9.89 15	0.25 5	0.57 3	3.24 5	5.14 4	11.8 5	13.0 4	2.77 2	8.35 2	8.20 3
YOUR METHOD	7.3	0.94 2	1.74 5	5.05 2	0.35 6	0.86 6	4.34 7	8.11 11	13.3 9	18.5 10	5.09 13	11.1 9	11.0 8
RegionTreeDP [18]	7.4	1.39 10	1.64 4	6.85 7	0.22 4	0.57 3	1.93 2	7.42 9	11.9 6	16.8 6	6.31 15	11.9 13	11.8 10
AdaptWeight [12]	8.0	1.38 9	1.85 7	6.90 8	0.71 7	1.19 8	6.13 8	7.88 10	13.3 10	18.6 12	3.97 7	9.79 6	8.26 4
SemiGlob [6]	9.8	3.26 18	3.96 15	12.8 20	1.00 10	1.57 9	11.3 15	6.02 5	12.2 7	16.3 5	3.06 4	9.75 5	8.90 5
RealtimeBP [21]	11.2	1.49 11	3.40 14	7.87 11	0.77 8	1.90 12	9.00 14	8.72 14	13.2 8	17.2 8	4.61 9	11.6 11	12.4 14
Layered [5]	12.4	1.57 12	1.87 8	8.28 12	1.34 12	1.85 10	6.85 11	8.64 13	14.3 11	18.5 11	6.59 17	14.7 16	14.4 16
GC+occ [2]	12.5	1.19 5	2.01 10	6.24 5	1.64 15	2.19 14	6.75 9	11.2 17	17.4 17	19.8 15	5.36 14	12.4 14	13.0 15
MultiCamGC [3]	12.8	1.27 6	1.99 9	6.48 6	2.79 20	3.13 18	3.60 6	12.0 18	17.6 18	22.0 17	4.89 12	11.8 12	12.1 12
TensorVoting [9]	14.2	3.79 19	4.79 19	8.86 13	1.23 11	1.88 11	11.5 16	9.76 15	17.0 16	24.0 19	4.38 8	11.4 10	12.2 13
GenModel [20]	14.7	2.57 16	4.74 18	13.0 21	1.72 16	3.08 17	16.9 19	6.86 7	15.0 13	19.2 13	4.64 10	14.9 17	11.4 9
CostRelax [11]	15.1	4.76 21	6.08 21	20.3 22	1.41 14	2.48 15	18.5 20	8.18 12	15.9 14	23.8 18	3.91 6	10.2 7	11.8 11
RealTimeGPU [14]	15.2	2.05 15	4.22 17	10.6 17	1.92 18	2.98 16	20.3 21	7.23 8	14.4 12	17.6 9	6.41 16	13.7 15	16.5 18
ReliabilityDP [13]	16.6	1.36 8	3.39 13	7.25 10	2.35 19	3.48 20	12.2 18	9.82 16	16.9 15	19.5 14	12.9 23	19.9 23	19.7 20
TreeDP [8]	16.8	1.99 14	2.84 11	9.96 16	1.41 13	2.10 13	7.74 12	15.9 21	23.9 21	27.1 22	10.0 20	18.3 20	18.9 19
GC [1d]	17.6	1.94 13	4.12 16	9.39 14	1.79 17	3.44 19	8.75 13	16.5 22	25.0 23	24.9 20	7.70 18	18.2 19	15.3 17
DP [1b]	20.6	4.12 20	5.04 20	12.0 18	10.1 25	11.0 25	21.0 22	14.0 19	21.6 19	20.6 16	10.5 21	19.1 21	21.1 21
SSD+MF [1a]	21.8	5.23 23	7.07 22	24.1 23	3.74 21	5.16 21	11.9 17	16.5 23	24.8 22	32.9 23	10.6 22	19.8 22	26.3 23
STICA [16]	22.4	7.70 24	9.63 25	27.8 24	8.19 23	9.58 23	40.3 25	15.8 20	23.2 20	37.7 24	9.80 19	17.8 18	28.7 24
SO [1c]	23.1	5.08 22	7.22 23	12.2 19	9.44 24	10.9 24	21.9 23	19.9 25	28.2 25	26.3 21	13.0 24	22.8 25	22.3 22
Infection [10]	24.1	7.95 25	9.54 24	28.9 25	4.41 22	5.53 22	31.7 24	17.7 24	25.1 24	44.4 25	14.3 25	21.3 24	38.0 25

Figure 7.2: Rankings of EnhancedBP compared to other state-of-the-art algorithms on the Middlebury (Scharstein and Szeliski, 2002) datasets. Note that this algorithm ranks 6th (above common Belief Propagation – represented by RealtimeBP) though I targeted multi-camera video sequences instead of the binocular still-frame context common to most algorithms on this list.

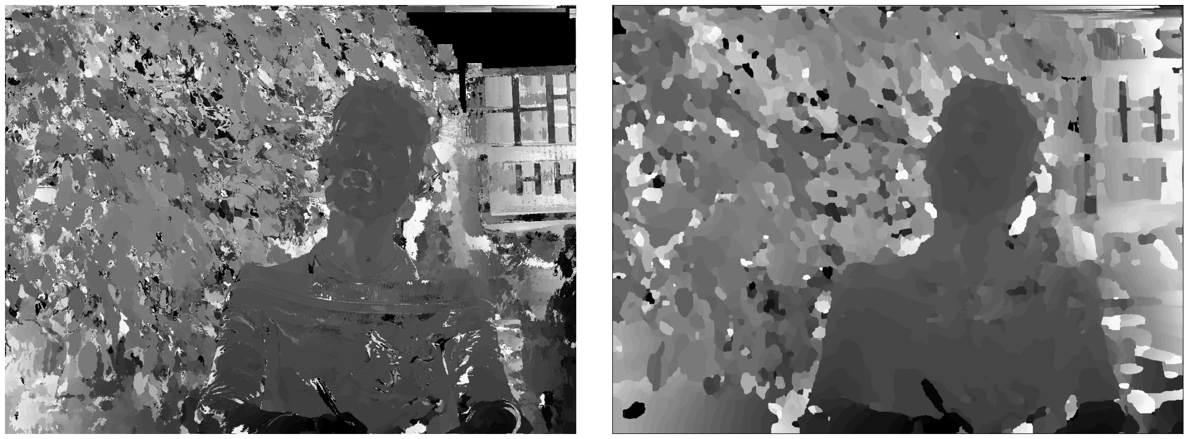

The technique of (Yoon and Kweon, 2005) EnhancedBP

Figure 7.3: Subjective comparison of results obtained on the (Webpage: 3DTV, 2007) Janine dataset. EnhancedBP on the right is compared to state of the art techniques of (Yoon and Kweon, 2005) on the left.

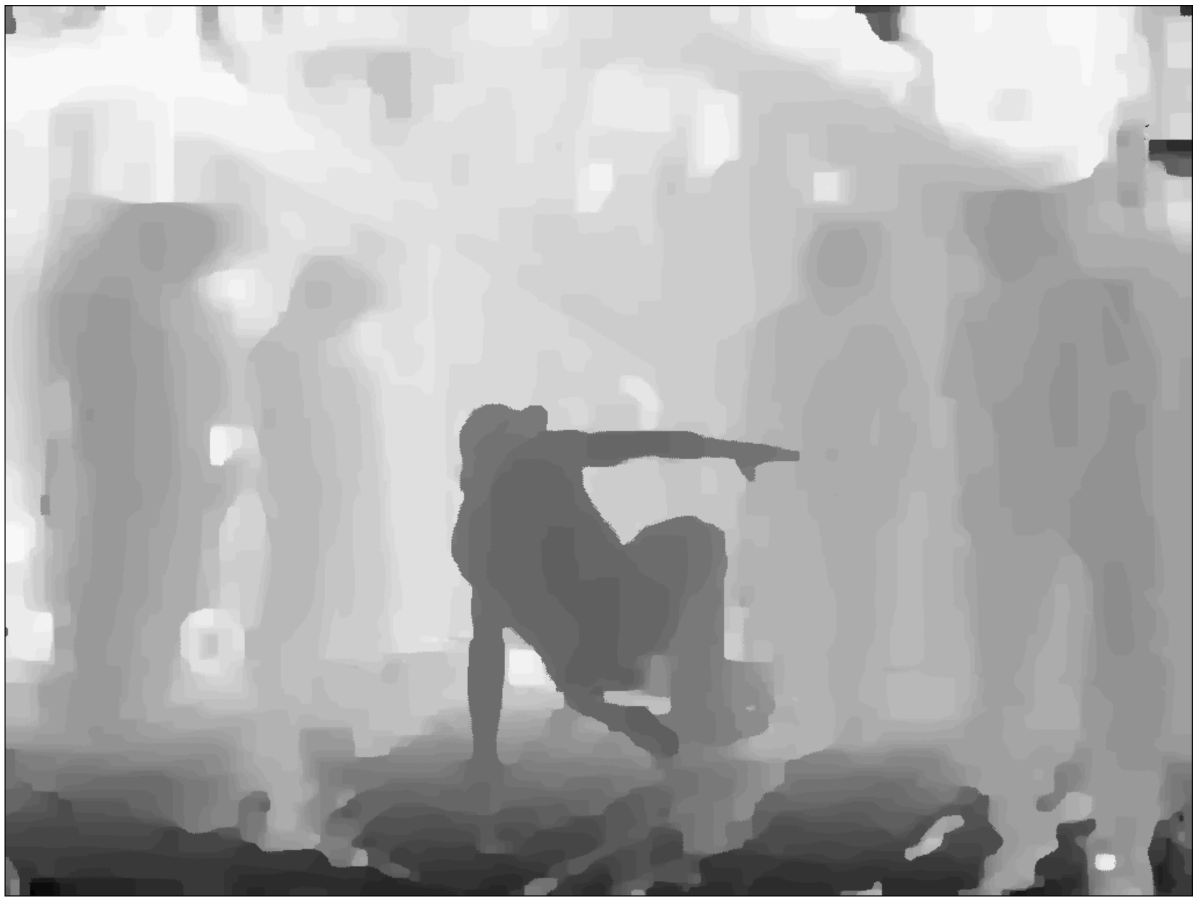


Figure 7.4: An example of the results obtained from EnhancedBP – QuietBP combined with RobustBP and BiasedBP.

(Zitnick et al., 2004) provides the breakdancers dataset. Figure 7.4 presents a sample result from EnhancedBP. The focus of (Zitnick et al., 2004) was video view interpolation: given the many cameras, produce an image from a virtual camera located near the existing ones and have it look as good as possible. This was not the focus of EnhancedBP, so these comparisons should be taken with "a grain of salt" so to speak, *i.e.* their objectives were not my objectives, so these results are in no way intended to suggest one algorithm is better than another, as the application domains are different. Regardless, it is a high quality and useful dataset which I have used as an invaluable baseline reference in my research.

In this dataset, there is a region covering entire scanlines, across the top of the image, for which the ideal results should never change. Selecting the 40th scanline from the top of the image, Figure 7.5 shows the temporal consistency of the depth maps produced by (Zitnick et al., 2004) and EnhancedBP. Note that the ideal solution is a straight line which does not move. EnhancedBP both produces results which are nearer to planar and temporally more consistent.

Figure 7.6 shows the reprojection errors for ten frames. Clearly EnhancedBP reduces reprojection errors, but the reader needs to beware that the application domain and algorithm of (Zitnick et al., 2004) would have selected a nearer depth map for reprojections, so there was no priority for their reprojection errors to be minimized in this case.

One artifact of the building a background model for BiasedBP is that I now have a background model that could be used as an estimate for filling in holes in reprojections. This approach is not at all the object of my research. Further, this model is only useful in this way for scenes with low depth complexity because a scene with higher depth complexity would have reprojection holes which are correctly filled by objects other than the background. Still, for this dataset it could be a useful aid to rendering. Figure 7.9 compares two identical reprojections, one without using this background model to fill holes, and one using the background model to fill holes.

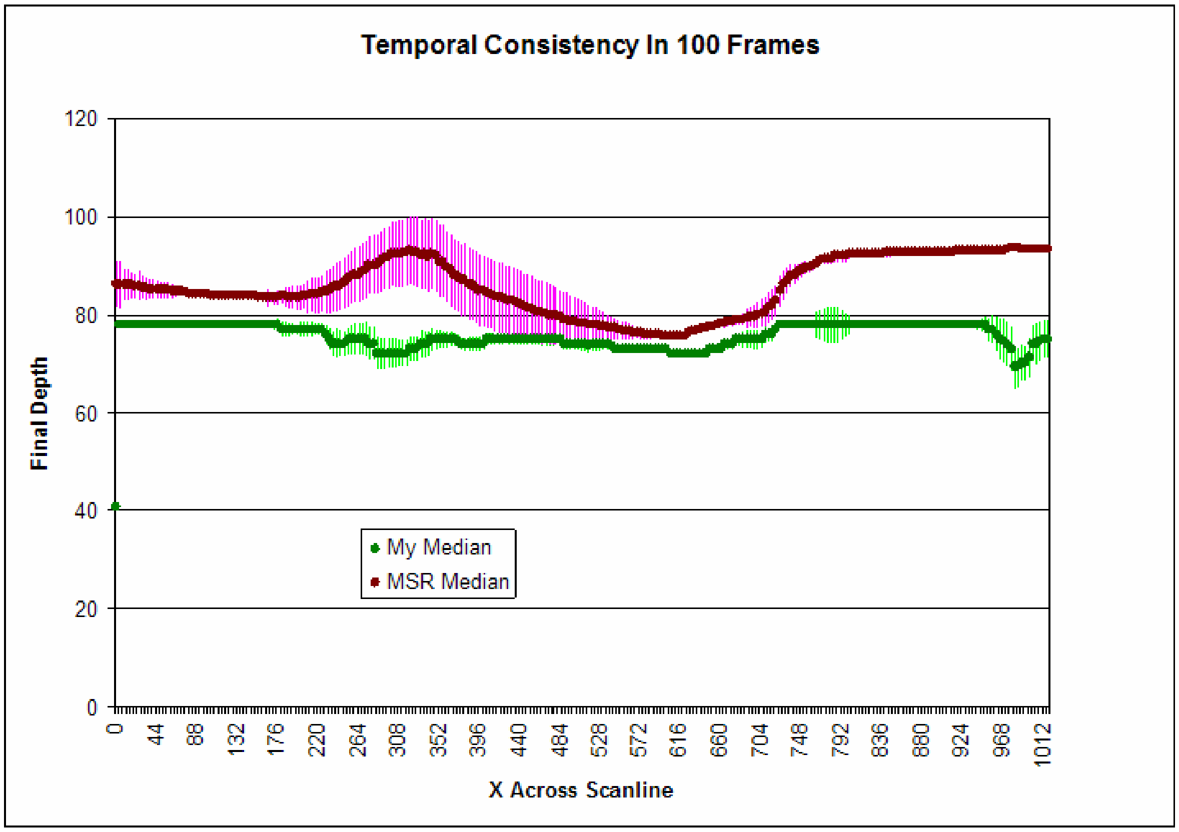
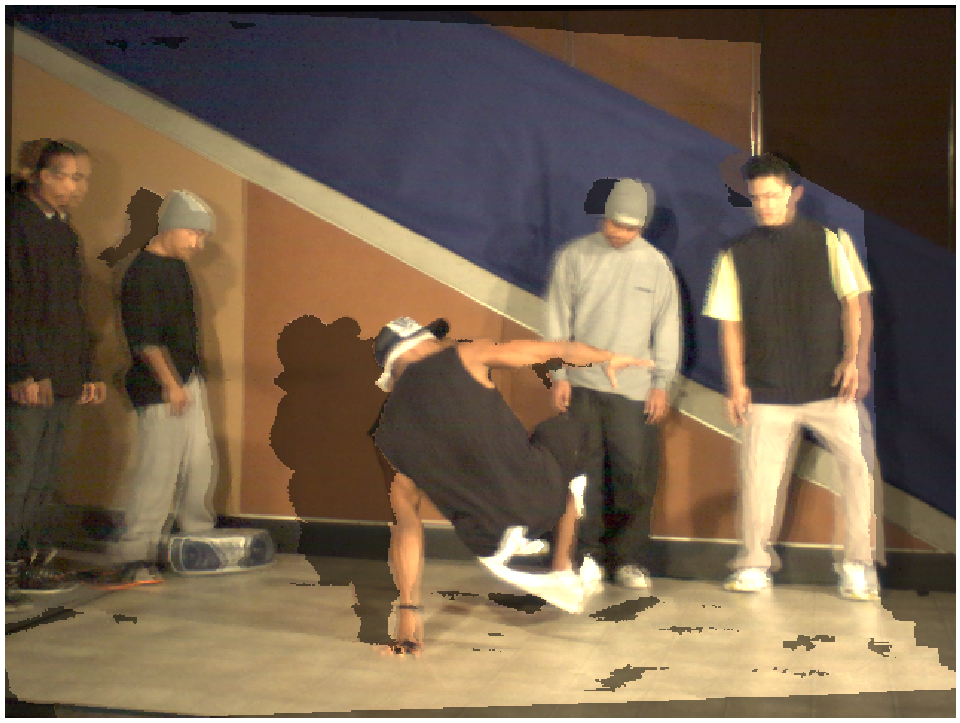


Figure 7.5: Comparison of temporal consistency on a single (the 40th) scanline. Ideal is no variance and a linear depth segment. EnhancedBP more nearly approximates those two objectives than the results of (Zitnick et al., 2004). EnhancedBP focused more on temporal consistency than theirs, as measured by lower variance.


(a) Reprojection overlay from (Zitnick et al., 2004)

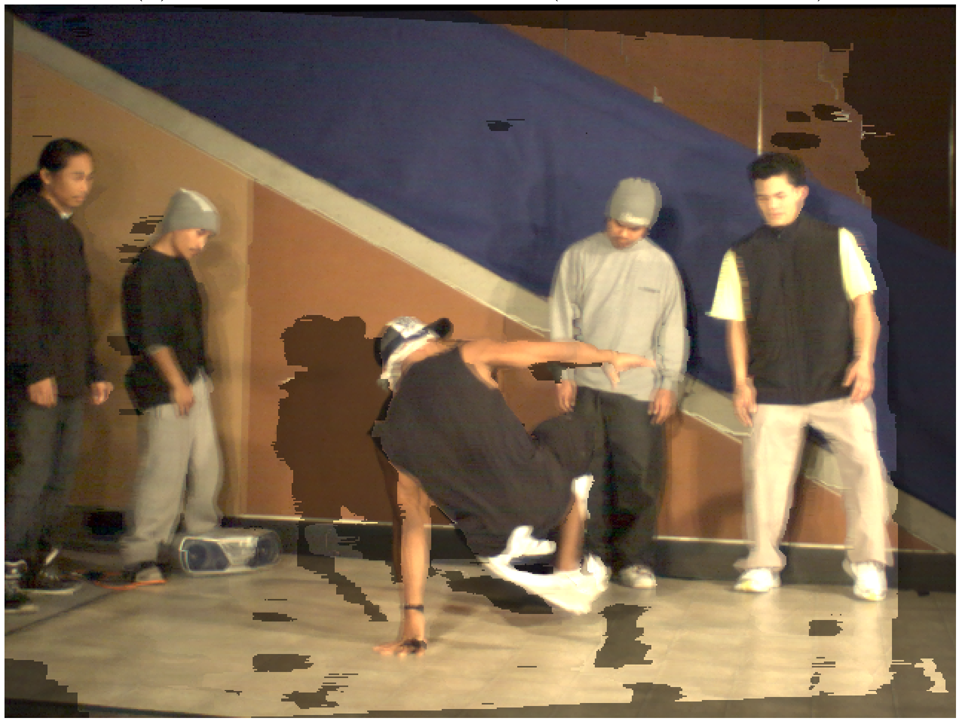
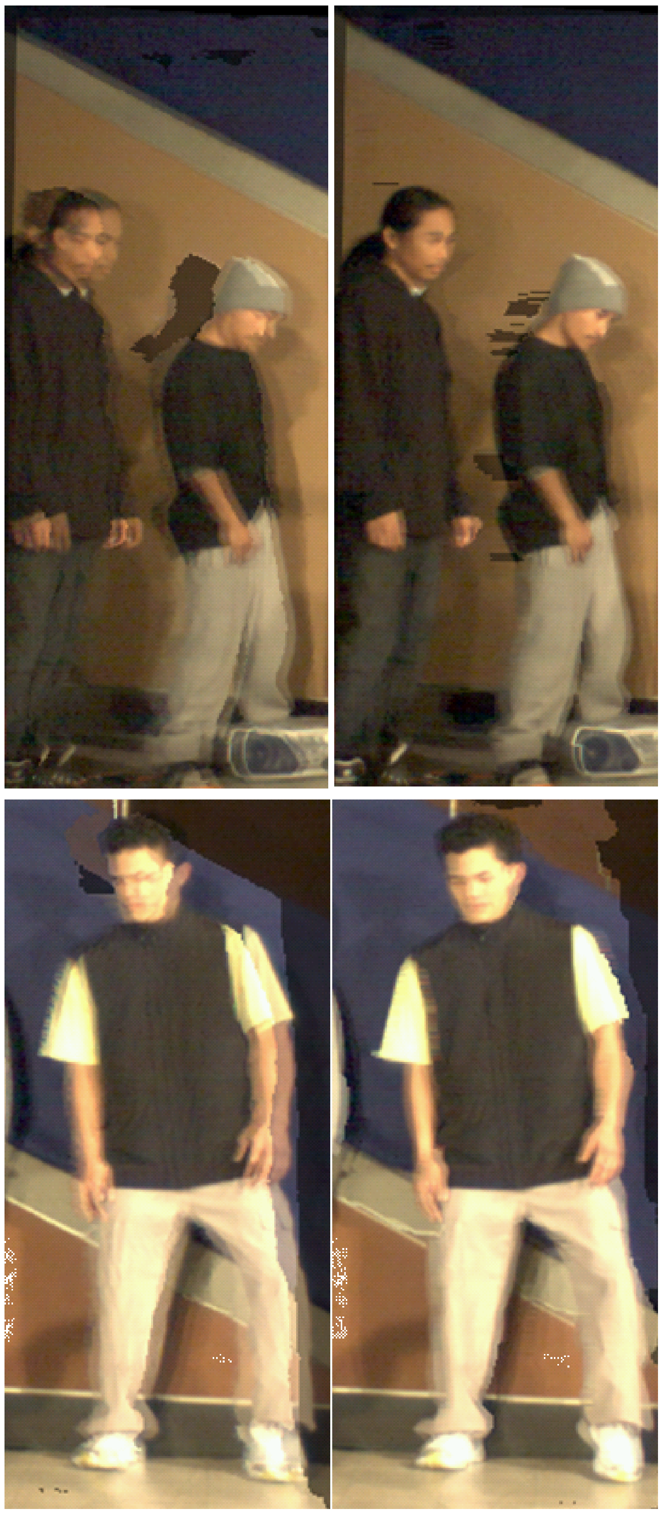

(b) Reprojection overlay from EnhancedBP

Figure 7.6: Subjective presentation of reprojection errors. (a) and (b) reproject from camera 3 (a central camera) to camera 7 (an extremal camera). The reprojected image and the ground truth image are overlayed, illustrating errors. Figure 7.7 shows more details from the same two images.



Previous work of (Zitnick et al., 2004)

My result

Figure 7.7: Zoomed in detail comparisons for the images in Figure 7.6.

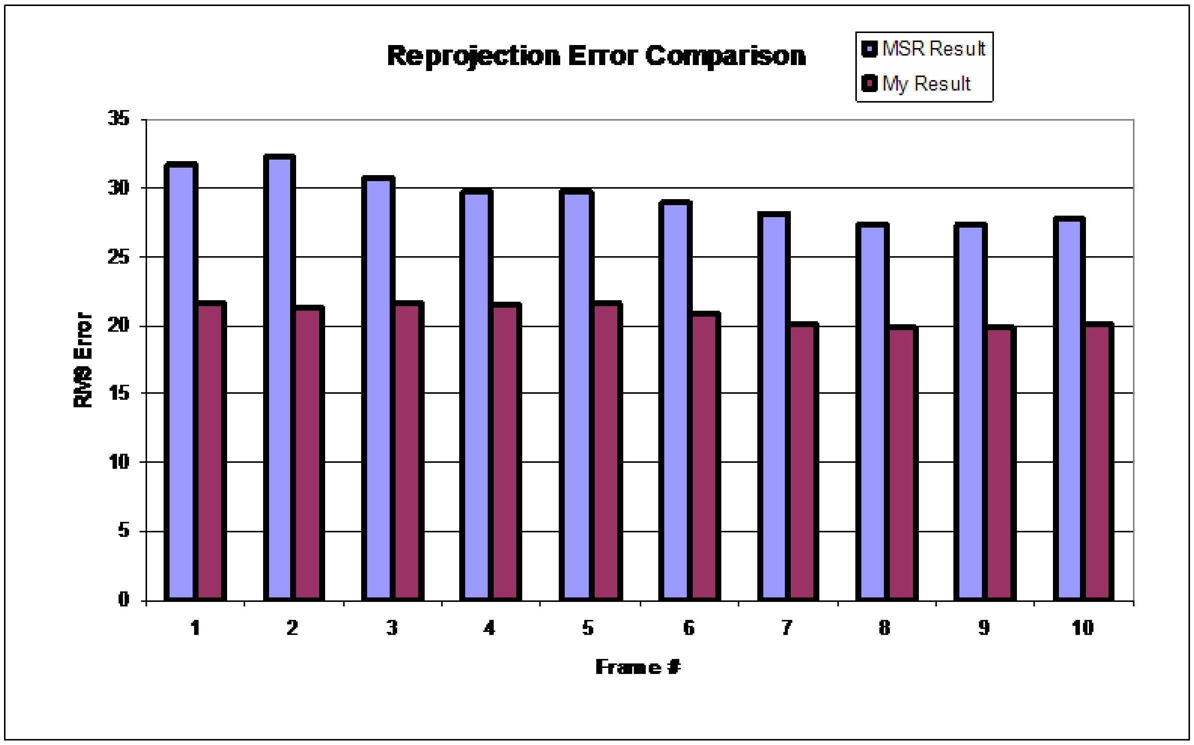


Figure 7.8: Objective comparison of reprojection errors. Total reprojection error measured as the sum of the absolute difference in each of the red, green, and blue channels.

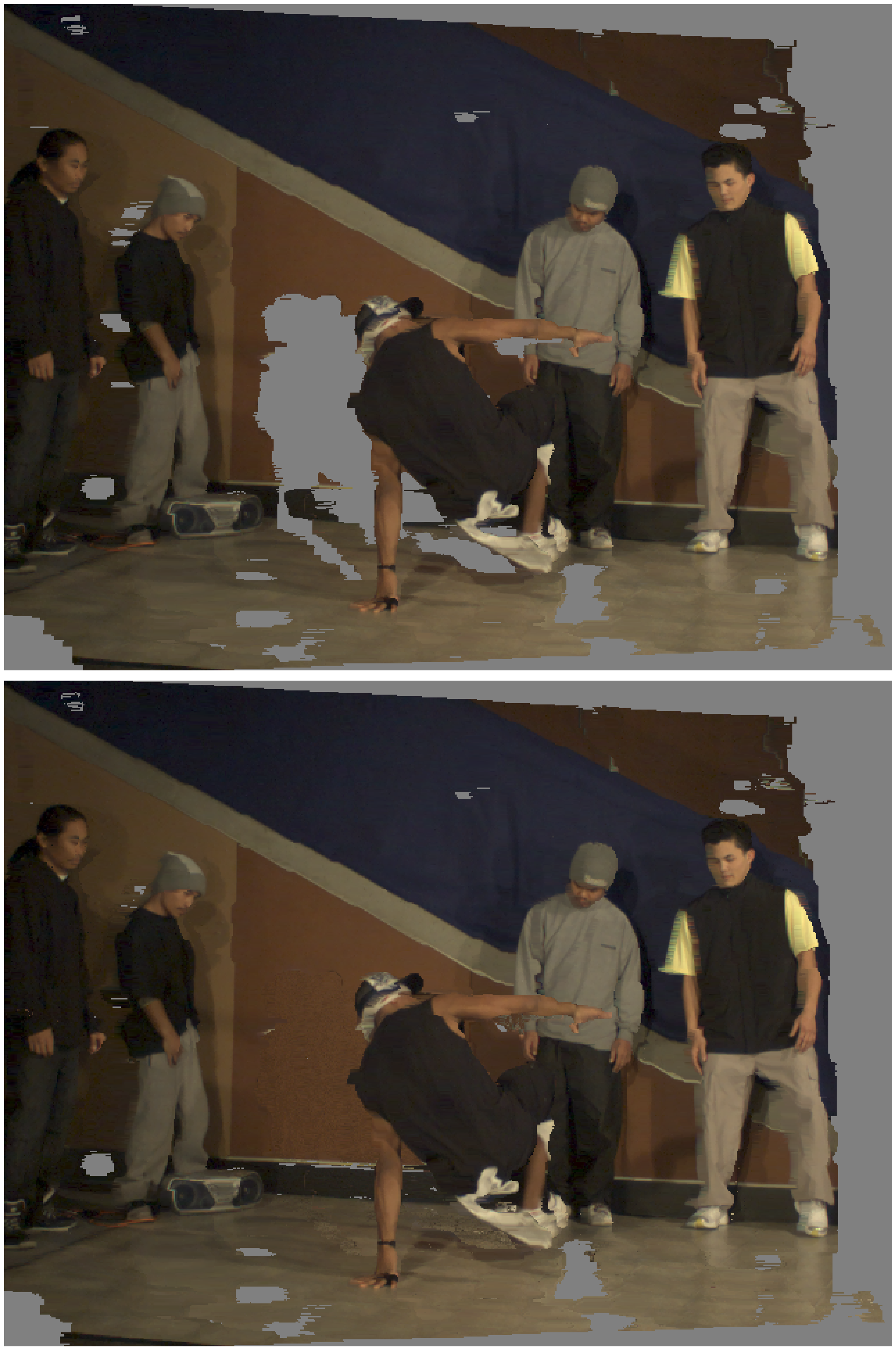


Figure 7.9: The first image is a pure reprojection, leaving holes due to the depth map being 2D for a camera at a different location. The second image is the same reprojection, but the holes are filled by the background model which is information available from the BiasedBP algorithm.

Chapter 8

Conclusions and Future Work

To arrive at any goal, we need to:

1. have a clear picture of where we are, and

2. have a clear picture of where we want to be.

And history teaches us that it also helps to have a picture of where we have been.

In this chapter I discuss my view of the lay ahead: my view of future BP research opportunities for 3D reconstruction. I then discuss another light in which we should be critiquing both the future of this work and the current situation, focusing on the current state. Finally I summarize the contributions I have made through this dissertation, bringing us to the current state of the art. With this vision, before us lies the task of making the future.

8.1 Future Work Avenues

I begin by proposing topics for future exploration. In no particular order:

Generic extension of my BP modifications: The enhancements I make in this dissertation to BP are focused on both the problem at hand and the many other applications of BP that are outside the scope of this dissertation. Exploration of the general theoretical value of these enhancements would be a sufficient topic for at least an entire dissertation by itself. That work might allow usage of enhanced BP in the many areas where it currently enjoys usage and also might enable its successful extension into new application domains.

Multi-Resolution BP: Solving the depth extraction problem at the pixel level induces an erroneous artificial dependency on the camera's resolution and sensing system. There are many very compelling motivations for higher resolution images, yet this will tend to push the pixel level information farther and farther away from the higher level salient components of the image. For example: in a low resolution image, the center of an object may be N pixels from the object's silhouette, but in a high resolution it may be more like M*N pixels away. On the other hand, higher resolution means there is much more information which could be used. I and many other researchers are seeing strong sensitivities in our results to the resolution of the input images. What is the right scale in which to work? Clearly the answer to that question varies in different parts of the scene. The implication is that we need to determine better ways to work in different parts of the scene at different scales. A BP algorithm which could natively work across multiple scales is a solution which would address this and is worth pursuing.

Better occlusion models: Concurrently with this dissertation, many researchers have been working on adjusting the equations to explicitly model occlusions and their effects (e.g. (Sun et al., 2003; Kolmogorov et al., 2003)). Their work leads to increasingly complex formulations, but explores an important space and achieves significant improvements. For my work, I dealt with occlusions indirectly via RobustBP. Occlusion handling remains a very hard problem, but one worth working on. I have opted to stay with simple models while hopefully leaving open the door for integration of more advanced occlusion models. The work of others has shown that occlusion modeling explicitly provides great benefits, and the "current best" way to do this is changing very rapidly: Occlusion modeling is a fertile area for continued research.

Blending of patch- and pixel- based BP algorithms: The work of (Zitnick et al., 2004) takes the approach of segmenting their images followed by correlation at the patch level, while my approach was pixel-based. When deciding to use patch-based or pixel-based graphs, there currently are engineering trade-offs that must be considered. But what if that decision did not have to be made? If one built a BP graph so as to connect nodes in both approaches, what would the behavior be? A single, unified system which would operate at both levels seems like a worthy path to explore. I have an expectation that a hybrid system would produce better results, for many of the same reasons as the related topic earlier in this section on "Multi-Resolution BP."

More real world datasets: Lack of readily available multi-camera video data of real world quality is a hindrance to the researcher. Acquiring and sharing better datasets for this application would be a significant aid to furthering the state of the art, and should be a priority for the research community.

Better inter-camera interactions: I attempted to solve the depth extraction problem for all the pixels in a single reference image, as most researchers do. But some are building systems which simultaneously solve for depths in multiple images, as illustrated by (Sun et al., 2005). Particularly, I would encourage the researcher to explore mechanisms which explicitly formulate the interactions and express the equations in terms of all involved images. This encourages thinking about how to use the available information and solve the problem of interest. This breaks away from simplifications that are needed less now than they used to be, as compute power has increased.

Better Robustness algorithms: My approaches to outlier detection, used in RobustBP, covered different approaches but were not necessarily well founded theoretically. That is to say, the theoretical foundations for them were generally in significantly different settings and it is not clear that those assumptions hold here, *e.g.* many statistical outlier detection algorithms assume a relatively high number of samples, and I only had at most seven at any given node which is too few for most statistics. On the other hand, my technique worked. A rigorous foundation for robustness in this setting would be valuable.

Mathematically-founded solution for QuietBP: The propagation damping shown experimentally in Chapter 6 and dealt with via QuietBP has not been shown mathematically. The vast majority of the mathematical conclusions in the literature in regards to BP are only applicable to the non-loopy cases (*i.e.* graphs that have no loops). The math involved in the loopy setting is significantly more complex, but my work has shown that there are important behaviors that have not been previously seen or explored. As BP continues to be favored in the research communities, exploring the complex systems induced by the loops will be of significant value to the community.

Smarter compatibility functions: The "compatibility functions" (the specification of how PDF smoothing occurs, as a function of external features of the image, *e.g.* "edge strength") used by most researchers have generally been fairly straightforward "smoothing" functions of simple low-level image features. There are two places to improve here. First, is the image features. One could imagine using segmentation boundaries, texture data, and even the current depth estimates as features for parameterizing the compatibility functions. Second, the effect of the function could be more advanced. For instance, instead of Gaussian smoothing, one could imagine modifying the filter as a function of the local image features, particularly imposing depth discontinuity estimates of information from other images simultaneously being solved (*e.g.* in "Better inter-camera interactions").

I want to single out one topic which is different from the others, "Sparse representations and dimensions compression". There are two reasons for singling this out: 1) It is primarily an implementation issue, while the other proposals are more theoretical in nature, and 2) Progress in almost *all* of the other topics require the tractability improvements that this one may provide. While I say this is primarily an implementation topic, I should emphasize that there are many theoretical issues that need to be addressed for successful contributions in even this topic.

Sparse representations and dimensions compression: One theoretical strength of BP is its maintaining multiple state candidates at all stages (*e.g.* for depths, it estimates the likelihoods for all depth candidates at all stages of the algorithm). This allows the algorithm to find solutions in initially weak areas, delay decisions between ambiguous states, etc. But, this strength implies significant resource requirements, both memory and computation. Often there exists a set of candidate states which are clearly not of interest, but the algorithm continues to require resources for these states. I would suggest investigating the representation of PDFs as compressed or sparse PDFs, and/or dynamic local adjustments of the dimension of the problem ("dimension" refers to the candidate states and their cardinality). What I have in mind is an algorithm which could start with a specific granularity of detail, iterate for while, then select the most likely N states and refine the granularity. This would allow a fixed resource budget to be used while allowing for very specific refinements. With resources for only N states at each node, one could hopefully use this technique to achieve the results at a level of detail that would have only been attainable with N*M states.

8.2 In the Bigger Picture

The light we should be using to look at how this dissertation fits into the bigger picture is typified by this statement that my machine learning professor emphasized to his class over and over and over: "The optimization algorithm is not the equations being optimized, is not the problem being solved" (Martinez, 1997). This complex statement summarizes the relationship between three distinct entities:

The Problem being Solved: This is the actual real-world problem that we are trying to solve. In our case this is the problem of estimating 3D position of points in a scene given only images of the scene.

The Equations being Optimized: This is our formulation of the problem that we want the computer to solve. Formulation of the problem is an absolutely essential component of using computers to solve problems. It is essential not only because it is needed in order to program the computer, but because it forces the researcher to think clearly and formally about the problem and its solutions. But, as all practitioners know, the equations are not always a perfect model for the actual problem being solved: assumptions are made, simplifications are made, real-world data is noisy, and so forth.

The Optimization Algorithm: Having determined a formulation, the researcher must next determine which algorithm will be used to find an optimal solution to the equations. There are many algorithms in existence, and more continue to appear. Each algorithm makes its own tradeoffs in search of an optimal solution, and the practitioner must always be aware of the implications of these assumptions.

And finally, this statement says that we must not only always remember that these are three different entities, but that we must be intimately aware of the implications of their differences. If our objective is to solve the problem (as it has been in this dissertation), then we must prioritize that about the others when making decisions. If our priority were theoretical optimization, as is the focus of many mathematicians, then we must prioritize that above the other two.

(Kolmogorov, 2006) presented an optimization algorithm which is well suited to the equations I have used. Their algorithm finds solutions to the equations that have lower energy (therefore better solutions to the equations) compared to BP, yet their solutions solve the original computer vision problem with lower quality. My research arguably yields higher quality results than either approach, through the alteration of the underlying algorithm. The wise researcher is cautious in this approach though, because one must simultaneously retain as much theoretical strength as one is able or else the solutions will end up too *ad hoc*. I am cautiously optimistic that I chose wisely, but have tried to remain as theoretically sound as possible. This is evidenced by my repeated comments throughout this dissertation pointing to potential generalization. I believe that I have retained the right balance, while prioritizing the solving of the class of real problems I have focused on.

My intuition is that Belief Propagation is a better optimization algorithm than others for the computer vision problems I am focusing on. This intuition derives from the message passing aspects of the algorithm, making use of local information while retaining theoretical underpinnings towards global solutions, simultaneous progress of the algorithm along all paths instead of early selections of promising routes, and the specifiable interactions between multiple current hypothesis. Thus this dissertation has focused on retaining the strengths of BP.

The last component of the inequality is the model: the equations being optimized, the formal expression of the problem to solve. Attempts to extend this model for solving the problem of temporally consistent reconstruction in a full 3D space, as opposed to the 2 1/2-D space shown in the previous chapters, have so far been fruitless. Part of that comes from the fact that I found it hard to actually write the equations – the various components of the equations would invariably induce nearly infinite recursive and very complex interactions with many of the other components of the equations. Perhaps there exists a way to push my solutions into the model specification aspect, but I have so far been unable to come up with a good formulation. Meanwhile, the alternative – keep the same simple equations used by other researchers and instead alter the algorithm in sound ways so as to better solve the problem – was the path I choose and is simple to formalize and achieves the goals quite well.

8.3 The State of the Art

There are two angles from which to consider the work in this dissertation: 1) from that of the driving application, and 2) from that of the theoretical algorithms underlying the solutions. The driving goal of this research has been to improve BP for multi-camera, dynamic video sequences. This research identifies specific weaknesses of BP and details new solutions for improving overall reconstruction quality.

I have successfully developed three new enhancements (Larsen et al., 2007) to the classic BP algorithm. RobustBP provides robustness to statistically outlying neighborhood connections – the benefits of this are seen most dramatically in dynamic video sequences. BiasedBP provides automatic self-adaptive use of higher level algorithm results – these benefits are dramatically seen in occluded regions, in both static and dynamic settings. Finally, QuietBP addresses a previously unidentified weakness of BP, that of poor convergence characteristics in largely uniform regions due to the effects of the loopy nature of the connectivity graph – this is seen both in temporal scenes and in untextured regions of the scene.

Finally, throughout this dissertation, I have attempted to discuss the theoretical aspects of these enhancements in a light of, and hope for, future extension to more general applications of BP – those beyond the scope of this dissertation. In light of the past and potential future, I believe these enhancements are the tip of the iceberg in regards to opening up the full potential of solving bigger and better problems, and obtaining higher quality solutions to those problems.

Bibliography

Baumgart, B. G. (1974). *Geometric Modeling for Computer Vision.* PhD thesis, Stanford University, Stanford Artificial Intelligence Laboratory, Memo no. AIM-249. 37

Birchfield, S. and Tomasi, C. (1998). A pixel dissimilarity measure that is insensitive to image sampling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(4). 43

Broadhurst, A., Drummond, T., and Cipolla, R. (2001). A probabilisitic framework for the Space Carving algorithm. In *Proceedings of the 8th International Conference of Computer Vision*, IEEE International Conferenece on Computer Vision, pages 388–393. IEEE Computer Society, IEEE Computer Society Press. 37

Caire, G., Shamai, S., and Verdu, S. (2003). Lossless data compression with error correcting codes. In *Proceedings, IEEE International Symposium on Information Theory.* IEEE. 20, 39

Carceroni, R. and Kutulakos, K. (2002). Multi-view scene capture by surfel sampling: From video streams to non-rigid 3d motion, shape and reflectance. *International Journal on Computer Vision (IJCV)*, 49(2-3):175–214. 30, 32, 37, 38, 60

Castleman, K. R. (1996). *Digital Image Processing.* Prentice Hall. 27

Comaniciu, D. and Meer, P. (2002). Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 24(5):603–619. 50

Felzenszwalb, P. F. and Huttenlocher, D. P. (2004). Efficient belief propagation for early vision. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR04)*, IEEE Computer Society Conference on Computer Vision. IEEE Computer Society. 41, 66

Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24:381–395. 50, 66

Freeman, W., Pasztor, E. C., and Carmichael, O. T. (2000). Learning low-level vision. *International Journal of Computer Vision*, 40(1):25–47. 39

Galvin, B., McCane, B., Novins, K., Mason, D., and Mills, S. (1998). Recovering motion fields: An evaluation of eight optical flow algorithms. In *Proceedings of the 9th British Machine Vision Conference*, pages 195–204. 61

Goldluecke, B. and Magnor, M. (2004). Space-time isosurface evolution for temporally coherent 3d reconstruction. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 350–355. 30, 31, 60

Gong, M. (2006). Enforcing temporal consistency in real-time stereo estimation. In *European Conference on Computer Vision (ECCV)*, pages 564–577. 30, 31, 32, 60

Hemayed, E. E. (2003). A survey of camera self-calibration. In *Proceedings of the IEEE Conference on Advanced Video and Signal Based Surveillance (AVSS)*, page 351. 34

Klaus, A., Sormann, M., and Karner, K. (2006). Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure". In *ICPR*. x, 1, 3, 44

Kolmogorov, V. (2006). Convergent tree-reweighted message passing for energy minimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 28(10):1568–1583. 107

Kolmogorov, V., Zabih, R., and Gortler, S. (2003). Generalized multi-camera scene reconstruction using graph cuts. In *EMMCVPR*. 102

Kutulakos, K. and Seitz, S. (2000). A theory of shape by space carving. *International Journal of Computer Vision (IJCV)*, 38(3):199–218. 30, 37

Larsen, E. S., Mordohai, P., Polleyfeys, M., and Fuchs, H. (2007). Temporally consistent reconstruction from multiple video streams using enhanced belief propagation. *IEEE 11th International Conference on Computer Vision (ICCV)*. 108

Laurentini, A. (1994). The visual hull concept for silhouette-based image understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 150–162. 37

Liu, Y., Chen, G., and Max, N. (2004). Visual hull rendering with multi-view stereo refinement. *Journal of Winter School of Computer Graphics (WSCG)*, 12. 37

MacQueen, J. B. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297, Berkeley, California. University of California Press. 55

Marr, D. and Poggio, T. (1976). Cooperative computation of stereo disparity. *Science*, 194:283–287. 24

Martinez, T. (1997). Private correspondence. 106

Matusik, W., Buehler, C., Raskar, R., Gortler, S. J., and McMillan, L. (2000). Image-based visual hulls. *Proceedings of SIGGRAPH*. 37

McCane, B., Novins, K., Crannitch, D., and Galvin, B. (2001). On benchmarking optical flow. *Computer Vision and Image Understanding*, 84(1):126–143. 61

McEliece, R. J., MacKay, D. J. C., and Cheng, J. F. (1998). Turbo decoding as an instance of pearl's 'belief propagation' algorithm. In *IEEE Journal on Selected Areas in Communications*, volume 16, pages 140–152. 20, 39

Nakamura, Y., Matsuura, T., Satoh, K., and Ohta, Y. (1996). Occlusion detectable stereo-occlusion patterns in camera matrix. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'96)*, pages 371–378, San Francisco, California. xii, 2, 51, 52, 53

Neumann, J. and Aloimonos, Y. (2002). Spatio-temporal stereo using multi-resolution subdivision surfaces. *International Journal of Computer Vision (IJCV)*, 47(1-3):181–193. 30, 60

Pearl, J. (1998). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible-Inference*. Kaufmann, San Francisco, CA, 2nd edition. 4, 20, 39

Perrone, M. P. (1993). *Improving regression estimation: Averaging methods for variance reduction with extensions to general convex measure optimization*. PhD thesis, Brown University. 66

Petrovic, N., Frey, B. J., Koetter, R., and Huang, T. (2001). Enforcing integrability for surface reconstruction algorithms using belief propagation in graphical models. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 743–748. IEEE Computer Society. 39

Proesmans, M., Van Gool, L., Pauwels, E., and Oosterlinck, A. (1994). Determination of optical flow and its discontinuities using non-linear diffusion. In *IEEE European Conference on Computer Vision (ECCV)*, pages B:295–304. xiii, 69, 90

Scharstein, D. and Szeliski, R. (2002). A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1-3):7–42. xi, xv, xvi, 13, 14, 15, 26, 68, 84, 85, 91, 92, 93

Seitz, S., Curless, B., Diebel, J., Scharstein, D., and Szeliski, R. (2006). A comparison and evaluation of multi-view stereo reconstruction algorithms. In *Proceedings of the IEEE Computer Society Computer Vision and Pattern Recognition (CVPR)*. 32

Shoup, R. (2001). Superpaint: An early frame buffer graphics system. In *IEEE Annals of the History of Computing*, volume 23, pages 32 –37. 54

Slabaugh, G., Schafer, R., and Hans, M. (2002). Multi-resolution space carving using level set methods. In *ICIP02*, pages II: 545–548. 37, 38

Sonka, M., Hlavac, V., and Boyle, R. (1999). *Image Processing, Analysis, and Machine Vision*. PWS Publishing, 2nd edition. 27

Sonnenwald, D. H., Maurin, H., Cairns, B., Manning, J., Freid, E., Welch, G., and Fuchs, H. (2006). Experimental comparison of the use of 2D and 3D telepresence technologies in distributed emergency medical situations. *Proceedings of the American Society of Information Science and Technology.* x, 9, 10

Sun, J., Li, Y., Kang, S., and Shum, H. (2005). Symmetric stereo matching for occlusion handling. In *International Conference on Computer Vision and Pattern Recognition*, pages II: 399–406. 103

Sun, J., Zheng, N.-N., and Shum, J.-Y. (2003). Stereo matching using belief propagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(7):787–800. x, 1, 3, 39, 41, 43, 44, 102

Svoboda, T., Martinec, D., and Pajdla, T. (2005). A convenient multi-camera self-calibration for virtual environments. *PRESENCE: Teleoperators and Virtual Environments*, 14:407–422. 34

Tao, H., Sawhney, H., and Kumar, R. (2001). Dynamic depth recovery from multiple synchronized video streams. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 118–124. 31

Tappen, M. F. and Freeman, W. T. (2003). Comparison of graph cuts with belief propagation for stereo, using identical MRF parameters. In *Proceedings, 2003 IEEE International Conference on Computer Vision.* 39

Trucco, E. and Verri, A. (1998). *Introductory Techniques for 3-D Computer Vision.* Prentice-Hall, Inc. 24

Vedula, S., Baker, S., Rander, P., Collins, R., and Kanade, T. (1999). Three-dimensional scene flow. In *International Conference on Computer Visions (ICCV)*, pages 722–729. 30, 60

Vedula, S., Baker, S., Rander, P., Collins, R., and Kanade, T. (2005). Three-dimensional scene flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 27(3):475–480. 30, 31, 60

Vedula, S., Baker, S., Seitz, S., and Kanade, T. (2000). Shape and motion carving in 6d. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 592–598. 30, 31

Webpage: 3DTV (2007). 3DTV network of excellence, `https://www.3dtv-research.org/publicSwLibrary.php`. xi, xvi, 10, 14, 18, 94

Webpage: Multi-View Stereo (2007). Mutli-view stereo evaluation webpage, `http://vision.middlebury.edu/mview`. xi, 17, 19

Webpage: Stereo (2007). Stereo evaluation webpage, `http://vision.middlebury.edu/stereo`. 14, 68

Welch, G., Noland, M., and Bishop, G. (2007). Complementary tracking and two-handed interaction for remote 3d medical consultation with a PDA. *Proceedings of Trends and Issues in Tracking for Virtual Environments, Workshop at the IEEE Virtual Reality Conference*. 10

Yedidia, J. S., Freeman, W. T., and Weiss, Y. (2001). Understanding belief propagation and its generalizations. In *International Joint Conference on Artificial Intelligence (IJCAI 2001)*, Distinguished Papers Track. 41, 43, 72

Yoon, K. and Kweon, I. (2005). Locally adaptive support-weight approach for visual correspondence search. In *CVPR*, pages II: 924–931. xvi, 44, 91, 94

Zhang, Y. and Kambhamettu, C. (2003). On 3-d scene flow and structure recovery from multiview image sequences. *IEEE Transactions on Systems, Man and Cybernetics*, 33(4):592–606. 30, 31, 32, 60

Zhang, Z. (2000). A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334. 34

Zitnick, C., Kang, S., Uyttendaele, M., Winder, S., and Szeliski, R. (2004). High-quality video view interpolation using a layered representation. *ACM Transactions on Graphics*, 23(3):600–608. xi, xii, xiii, xvi, 11, 14, 16, 46, 54, 56, 62, 63, 75, 84, 95, 96, 97, 98, 103