A Multi-View Approach to Motion and Stereo

Richard Szeliski

July 19, 1999

Technical Report

MSR-TR-99-19

Microsoft Research

One Microsoft Way

Redmond, WA 98052

http://www.research.microsoft.com/

A shorter version of this paper appeared at the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'99), Fort Collins, CO, June 1999, pp. 157–163.

Abstract

This paper presents a new approach to computing dense depth and motion estimates from multiple images. Rather than computing a single depth or motion map from such a collection, we associate motion or depth estimates with multiple images in the collection. This has the advantage that the depth or motion of regions occluded in one image will still be represented in some other image. Thus, tasks such as novel view interpolation or motion-compensated prediction can be solved with greater fidelity. It also enables us to reason about occlusions by comparing estimates across multiple images. Furthermore, the natural variation in appearance between different images can be captured. To formulate motion and structure recovery, we cast the problem as a global optimization over the unknown motion or depth maps, and use robust smoothness constraints (i.e.. regularization or Bayesian prior models) to constrain the space of possible solutions. We develop and evaluate some motion and depth estimation algorithms based on this framework.

1 Introduction

Stereo and motion have long been central research problems in computer vision. Early work was motivated by the desire to recover depth maps and coarse shape and motion models for robotics and object recognition applications. More recently, depth maps obtained from stereo (or alternately dense correspondence maps obtained from motion) have been combined with texture maps extracted from input images in order to create realistic 3-D scenes and environments for virtual reality and virtual studio applications (McMillan and Bishop, 1995; Szeliski and Kang, 1995; Kanade *et al.*, 1996; Blonde *et al.*, 1996), as well as for motion-compensated prediction in video processing applications (Le Gall, 1991; Lee *et al.*, 1997; de Hann and Beller, 1998). Unfortunately, the quality and resolution of most of today's algorithms falls quite short of that demanded by these new applications, where even isolated errors in correspondence become readily visible when composited with synthetic graphical elements.

One of the most common errors made by these algorithms is a mis-estimation of depth or motion near occlusion boundaries. Traditional correspondence algorithms assume that every pixel has a

corresponding pixel in all other images. Obviously, in occluded regions, this is not so. Furthermore, if only a single depth or motion map is used, it is impossible to predict the appearance of the scene in regions which are occluded (Figure 1). Other problems include dealing with untextured or regularly textured regions, and with viewpoint-dependent effects such as specularities or shading.

One novel approach to tackling these problems is to build a *disparity space* or 3D volumetric model of the scene (Yang *et al.*, 1993; Intille and Bobick, 1994; Collins, 1996; Scharstein and Szeliski, 1998; Seitz and Dyer, 1997; Szeliski and Golland, 1998). The scene volume is discretized, often in terms of equal increments of disparity. The goal is then to find the voxels which lie on the surfaces of the objects in the scene. The benefits of such an approach include the equal and efficient treatment of a large number of images (Collins, 1996), the possibility of modeling occlusions (Intille and Bobick, 1994), and the detection of mixed pixels at occlusion boundaries (Szeliski and Golland, 1998). Unfortunately, discretizing space volumetrically introduces a large number of degrees of freedom and leads to sampling and aliasing artifacts. To prevent a systematic "fattening" of depth layers near occlusion boundaries, variable window sizes (Kanade and Okutomi, 1994) or iterative evidence aggregation (Scharstein and Szeliski, 1998) can be used. Sub-pixel disparities can be estimated by finding the the analytic minimum of the local error surface (Tian and Huhns, 1986; Matthies *et al.*, 1989) or using gradient-based techniques (Lucas and Kanade, 1981), but this requires going back to a single depth/motion map representation.

Another active area of research is the detection of parametric motions within image sequences (Wang and Adelson, 1993; Irani *et al.*, 1995; Sawhney and Ayer, 1996; Black and Jepson, 1996; Weiss and Adelson, 1996; Weiss, 1997). Here, the goal is to decompose the images into sub-images, commonly referred to as *layers*, such that the pixels within each layer move with a parametric transformation. For rigid scenes, the layers can be interpreted as planes in 3D being viewed by a moving camera, which results in fewer unknowns (Baker *et al.*, 1998). This representation facilitates reasoning about occlusions, permits the computation of accurate out-of-plane displacements, and enables the modeling of *mixed* or *transparent* pixels. Unfortunately, initializing such an algorithm and determining the appropriate number of layers is not straightforward, and may require sophisticated optimization algorithms such as expectation maximization (EM) (Torr *et al.*, 1999).

Figure 1: Slice through a motion sequence spatio-temporal volume. A standard estimation algorithm only estimates the motion at the center frame (⇒), whereas our multi-view approach produces several additional estimates (→). A layered motion model would use two (or more) layers to describe this motion, whereas a volumetric approach would assign one voxel to each "streak".

Thus, all current correspondence algorithms have their limitations. Single depth or motion maps cannot represent occluded regions not visible in the reference image and usually have problems matching near discontinuities. Volumetric techniques have an excessively large number of degrees of freedom and have limited resolution, which can lead to sampling or aliasing artifacts. Layered motion and stereo algorithms require combinatorial search to determine the correct number of layers and cannot naturally handle true three-dimensional objects (they are better at representing "cutout" scenes). Furthermore, none of these approaches can easily model the variation of scene or object appearance with respect to the viewing position.

In this paper, we propose a new representation which overcomes most of these limitations. Rather than estimating a single depth or motion map, we associate a depth or motion map with *each* input image (or some subset of them, Figure 1). Furthermore, we try to ensure consistency between these different estimates using a *motion compatibility* constraint, and reason about occlusion relationships by computing pixel *visibilities*.

Our new approach is motivated by several target applications. One application is *view interpolation*, where we wish to generate novel views from a collection of images with associated depth maps. The use of multiple depth maps and images allows us to model partially occluded regions and to model view-dependent effects (such as specularities) by blending images with taken from nearby viewpoints (Debevec *et al.*, 1996). Another application is *motion-compensated frame interpolation* (e.g., for video compression, rate conversion, or de-interlacing), where the ability to

predict bi-directionally (from both previous and future keyframes) yield better prediction results (Le Gall, 1991). A third application is as a low-level representation from which segmentation and layer extraction (or 3D model construction) can take place.

We begin this paper with a review of previous work in stereo matching and motion estimation. In Section 3, we present our basic multi-view image matching framework. In Section 4, we explain in more details the cost functions used to formulate our estimation problem, and present our new method for estimating visibility in multi-image sequences. Section 5 presents our estimation algorithm, which combines ideas from hierarchical estimation, correlation-style search, and sub-pixel motion estimation. We present some experimental results in Section 6. We conclude the paper with a discussion of these results, and a list of topics for future research.

2 Previous Work

Stereo matching (and the more general problem of stereo-based 3-D reconstruction) are fields with very rich histories (Barnard and Fischler, 1982; Dhond and Aggarwal, 1989). In this section, we focus only on previous work related to our central topics of interest: pixel-accurate matching with sub-pixel precision, the handling of occlusion boundaries, and the use of more than two images.

A useful framework for thinking about stereo algorithms is to subdivide the stereo matching process into three tasks: the initial computation of matching costs, the aggregation of local evidence, and the selection or computation of a disparity value for each pixel (Yang *et al.*, 1993; Scharstein and Szeliski, 1998).

The most fundamental element of any correspondence algorithm is a matching cost that measures the similarity of two or more corresponding pixels in different images. Matching costs can be defined locally (at the pixel level), e.g., as absolute (Kanade *et al.*, 1996) or squared intensity differences (Matthies *et al.*, 1989), using edges (Baker, 1980) or filtered images (Jenkin *et al.*, 1991; Jones and Malik, 1992). Robust matching costs can also be used (Black and Anandan, 1991; Zabih and Woodfill, 1994; Black and Rangarajan, 1996; Scharstein and Szeliski, 1998). Alternatively, matching costs may be defined over an area, e.g., using correlation (Ryan *et al.*, 1980) (this can be

viewed as a combination of the matching and aggregation stages).

Aggregating support is necessary to disambiguate potential matches. A support region can either be two-dimensional at a fixed disparity (favoring fronto-parallel surfaces), or three-dimensional in (x, y, d) space (allowing slanted surfaces). Two-dimensional evidence aggregation has been done using both fixed square windows (traditional) and windows with adaptive sizes (Arnold, 1983; Kanade and Okutomi, 1994). Three-dimensional support functions include limited disparity gradient (Pollard *et al.*, 1985), Prazdny's coherence principle (Prazdny, 1985) (which can be implemented using two diffusion processes (Szeliski and Hinton, 1985)), and iterative (non-linear) evidence aggregation (Scharstein and Szeliski, 1998).

The easiest way of choosing the best disparity is to select at each pixel the minimum aggregated cost across all disparities under consideration ("winner-take-all"). A problem with this is that uniqueness of matches is only enforced for one image (the *reference image*), while points in the other image might get matched to multiple points. Cooperative algorithms employing symmetric uniqueness constraints are one attempt to solve this problem (Marr and Poggio, 1976).

Hierarchical coarse-to-fine algorithms are commonly used to limit the amount of search required to find the best match, and to allow lower-frequency information to influence areas where the local evidence is ambiguous. These algorithms often work by allowing finer levels in a multiresolution representation to correct the estimates computed at coarser levels (Quam, 1984; Terzopoulos, 1986a; Anandan, 1989; Bergen *et al.*, 1992).

Occlusion is another very important issue in generating high-quality stereo maps. Many approaches ignore the effects of occlusion; others try to minimize them by using a cyclopean disparity representation (Barnard, 1989), or try to recover occluded regions after the matching by cross-checking (Fua, 1993). Several authors have addressed occlusions explicitly, using Bayesian models and dynamic programming (Arnold, 1983; Ohta and Kanade, 1985; Belhumeur and Mumford, 1992; Cox, 1994; Geiger *et al.*, 1992; Intille and Bobick, 1994). However, such techniques require the strict enforcement of *ordering constraints* (Yuille and Poggio, 1984), and are thus limited to two input images.

Sub-pixel (fractional) disparity estimates, which are essential for applications such as view

interpolation, can be computed by fitting a curve to the matching costs at the discrete disparity levels (Lucas and Kanade, 1981; Tian and Huhns, 1986; Matthies *et al.*, 1989; Kanade and Okutomi, 1994). This provides an easy way to increase the resolution of a stereo algorithm with little additional computation. However, to work well, the intensities being matched must vary smoothly, or the disparity space must be sampled finely enough.

More than two images are used in multiframe stereo to increase stability of the algorithm (Bolles *et al.*, 1987; Matthies *et al.*, 1989; Okutomi and Kanade, 1993; Kang *et al.*, 1995; Collins, 1996). This can help disambiguate potential matches (Okutomi and Kanade, 1993) and reduce the error in the estimates (Matthies *et al.*, 1989). There is also a long tradition of computing optical flow from spatio-temporal derivatives (Heeger, 1988) that can use multiple frames of temporal support. However, no previous algorithm has suggested simultaneously estimating a consistent set of depth maps or motion estimates from multiple images.

3 The multi-view framework

As we mentioned before, our multi-view framework is motivated by several requirements. These include the ability to accurately predict the appearance of novel views or in-between images and the ability to extract higher-level representations such as layered models or surface-based models. Therefore, our goal is to estimate a collection of motion or depth field associated with several images, such that other images in the input collection can be predicted based on these estimates.

Assume that we are given a collection of images $\{I_t(\mathbf{x}_t)\}$, where I_t is the image at time or location t, and $\mathbf{x}_t = (x_t, y_t)$ indexes pixels in image I_t. A simple way to formulate a multi-view matching criterion is

$$\mathcal{C}(\{\mathbf{u}_s\}) = \sum_{s \in S} \sum_{t \in \mathcal{N}(s)} w_{st} \sum_{\mathbf{x}_s} \rho\left(I_s(\mathbf{x}_s) - I_t(\mathbf{x}_t)\right). \tag{1}$$

The images I_s form the set S of *keyframes* (or *key-views*) for which we will estimate a motion or depth estimate $\mathbf{u}_s(\mathbf{x}_s)$ (see Section 3.1 for a description of our motion models). The decision as to which images are keyframes is problem-dependent, much like the selection of I and P frames in

video compression (Le Gall, 1991). For 3D view interpolation, one possible choice of keyframes would be a collection of *characteristic views*.

Images $I_t, t \in \mathcal{N}(s)$ are *neighboring frames* (or views), for which we require that corresponding pixel brightnesses (or colors) agree. The pixel coordinate $\mathbf{x}_t$ corresponding to a given keyframe pixel $\mathbf{x}_s$ with flow/depth $\mathbf{u}_s$ can be computed according to the motion model (Section 3.1). The constants w_{st} are the *inter-frame weights* which dictate how much neighboring frame t will contribute to the estimate of $\mathbf{u}_s$. [1] [2]

Corresponding pixel brightness or color differences are passed through a robust penalty function ρ, which is discussed in more detail in Section 3.2. In the case of color images, we currently pass each color channel separately through the robust penalty function. A better approach would be to compute a reasonable color-space distance between pixels, and pass this through a robust penalty (since typically either all bands are affected by outlier events such as occlusions or specularities, or none of them are).

Below, we discuss possible motion models and robust penalty functions in more detail. A more detailed cost function (the one we actually use) is described in Section 4.

3.1 Motion models

Given our basic matching criterion, a variety of motion models can be used, depending on the imaging/acquisition setup and the problem at hand. Bergen *et al.* (Bergen *et al.*, 1992) present a variety of instantaneous (infinitessimal) motion models in a unified estimation framework, ranging from global parametric motion, through rigid motion, to general 2-D flow. Szeliski and Coughlan (Szeliski and Coughlan, 1997) present a similar set of motion models for finite (larger) motion. In this paper, we focus on two motion models presented in (Szeliski and Coughlan, 1997): *constant flow* (uniform velocity), and *rigid body motion*. Each motion model provides us with a definition of $\mathbf{x}_t$ as a function of $\mathbf{x}_s$ and $\mathbf{u}_s(\mathbf{x}_s)$.

[1] Note that we could set $w_{st} = 0$ for $t \notin \mathcal{N}(s)$ and replace $\sum_{t \in \mathcal{N}(s)}$ with $\sum_t$.

[2] A more geometrically plausible weighting would reflect the degree of similarity between the viewing rays on a pixel-by-pixel basis (Gortler *et al.*, 1996), but we have not implemented this idea.

The constant flow motion model assumes a (locally) constant velocity,

$$\mathbf{x}_t = \mathbf{x}_s + (t-s)\mathbf{u}_s(\mathbf{x}_s). \tag{2}$$

This model is appropriate when processing regular video with a relatively small sliding window of analysis. Note that this model does *not* require constant flow throughout the whole video. Rather, it assumes that within the window $t \in \mathcal{N}(s)$, the constant flow model is a reasonably good approximation to the true velocity.

The rigid motion model assumes that the camera is moving in a rigid scene or observing a single rigidly moving object, but does *not* assume a uniform temporal sampling rate (e.g., the pictures can be taken by different cameras). The motion model is given by

$$\mathbf{x}_t = \mathcal{P}\left(\mathbf{M}_{ts}\hat{\mathbf{x}}_s + \mathbf{e}_{ts}d_s(\mathbf{x}_s)\right), \tag{3}$$

where $\hat{\mathbf{x}}_s = (x_s, y_s, 1)$, $\mathbf{M}_{ts}$ is a *homography* describing a global parametric motion, $d_s(\mathbf{x}_s)$ is a per-pixel displacement[3] that adds some motion towards the *epipole* $\mathbf{e}_{ts}$, and $\mathcal{P}(x, y, z) = (x/z, y/z)$ is the perspective projection operator. This formulation is most often associated with *plane plus parallax* representations of motion (Sawhney, 1994; Kumar *et al.*, 1994; Szeliski and Coughlan, 1997; Baker *et al.*, 1998). However, it is equally applicable when no dominant planar motion exists.

For a calibrated camera with intrinsic viewing matrix $\mathbf{V}_t$, we have $\mathbf{M}_{ts} = \mathbf{V}_t\mathbf{R}_t\mathbf{R}_s^{-1}\mathbf{V}_s^{-1}$ and $\mathbf{e}_{ts} = \mathbf{V}_t\mathbf{R}_t(\mathbf{c}_s - \mathbf{c}_t)$, where $\mathbf{R}_t$ is the camera's orientation and $\mathbf{c}_t$ is its position in space. In this case, $\mathbf{M}_{ts}$ is the homography corresponding to the plane at infinity (Baker *et al.*, 1998). If all of the cameras live in the same plane with their optical axes perpendicular to the plane, d_s is the *inverse depth* (sometimes called the *disparity* (Kanade and Okutomi, 1994)) of a pixel. In principle, we could estimate $\{\mathbf{M}_{ts}, \mathbf{e}_{ts}\}$ at the same time as we estimate $d_s(\mathbf{x}_s)$ (Bergen *et al.*, 1992; Szeliski and Coughlan, 1997), but we have not yet implemented this part of the algorithm. Instead, we assume that these global parameters are computed ahead of time by tracking some feature points and doing a projective reconstruction of the camera ego-motion.

[3] In the remainder of the paper, we use the notation $\mathbf{u}_s$ to indicate the unknown per-pixel motion parameter, even when it is actually a scalar displacement d_s.

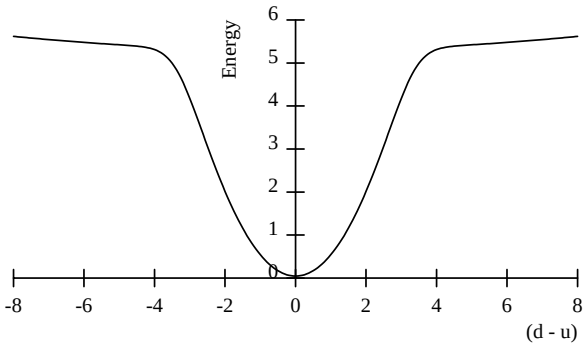


Figure 2: Contaminated Gaussian robust penalty function

3.2 Robust penalty functions

In order to account for *outliers* among the pixel correspondences (e.g., because pixels might be occluded in some images), we use a robust matching criterion. Black and Rangarajan (Black and Rangarajan, 1996) provide a nice survey of robust statistics applied to image matching and image segmentation problems. They not only prove the equivalence between robust continuity constraints and line processes, but also provide a useful catalog of commonly used robust estimators.

In our work, we use a *contaminated Gaussian* distribution (Szeliski, 1990; Scharstein and Szeliski, 1998), which is a mixture of a Gaussian distribution and a uniform distribution. The probability function is

$$p(x;\sigma,\epsilon) = \frac{1}{Z}(1-\epsilon)\exp(-\frac{x^2}{2\sigma^2}) + \epsilon, \tag{4}$$

where σ in the standard deviation of the *inlier* process, ϵ is the probability of finding an *outlier*, and Z is a normalizing constant. The robust penalty function is the negative log likelihood,

$$\rho(x;\sigma,\epsilon) = -\log\left((1-\epsilon)\exp(-\frac{x^2}{2\sigma^2}) + \epsilon\right). \tag{5}$$

The shape of this function (Figure 2) is qualitatively similar to Leclerc's inverted Gaussian function (Leclerc, 1989), and is equivalent to the robust function derived by Geiger and Girosi (Geiger and Girosi, 1991), although it is derived based on different assumptions.

Our main motivation for using a contaminated Gaussian is to explicitly represent and reason about the inlier and outlier processes separately. For example, in Section 4.3 we propose a robust controlled smoothness constraint where the strength of the constraint depends on the neighboring

pixel color similarity. Pixels which are more similar in color should have a higher penalty for motion/depth discontinuities (Gamble and Poggio, 1987), but how do we encode this? Do we simply scale up the penalty function for similar pixels, or do we change the size of the quadratic penalty region? In Appendix A we show that pixel color similarity affects the outlier probability but not the inlier variance. Using a contaminated Gaussian gives us a principled way to incorporate these effects.

Is the contaminated Gaussian the best possible model? Probably not. In fact, it should be possible to estimate the probability distribution of corresponding pixel or disparity differences and of neighboring disparity values by analyzing these residuals after a solution is obtained. Performing this analysis is something we would like to do in the future.

4 Optimization criteria

The actual cost function we use consists of three terms,

$$\mathcal{C} = \mathcal{C}_I + \mathcal{C}_T + \mathcal{C}_S, \tag{6}$$

where $\mathcal{C}_I$ measures the *brightness compatibility*, i.e., the raw differences in corresponding pixel intensities or colors, $\mathcal{C}_T$ measures the temporal *flow compatibility*, i.e., the agreement between flow estimates in different frames, and $\mathcal{C}_S$ measures the *flow smoothness*. Below, we give more details on each of these three terms.

4.1 Brightness compatibility

The brightness compatibility term measures the degree of agreement in brightness or color between corresponding pixels,

$$\mathcal{C}_I(\{\mathbf{u}_s\}) = \sum_{s \in S} \sum_{t \in \mathcal{N}(s)} w_{st} \sum_{\mathbf{x}_s} v_{st}(\mathbf{x}_s) e_{st}(\mathbf{x}_s), \tag{7}$$

where

$$e_{st}(\mathbf{x}_s) = \rho_I \left(I_s(\mathbf{x}_s) - \gamma_{st} I_t(\mathbf{x}_t) - \beta_{st}; \sigma_I, \epsilon_I \right). \tag{8}$$

Compared with Equation (1), we have added a visibility factor $v_{st}(\mathbf{x}_s)$, which encodes whether pixel $\mathbf{x}_s$ is *visible* in image I_t (rules for determining this visibility factor are discussed in Section 4.4). We have also generalized the robust penalty to allow for a global bias (β_{st}) and gain (γ_{st}) change (Gennert, 1988).[4] We could, if desired, also make σ_I depend on the local intensity variation in order to account for small motion errors such as jitter (Simoncelli *et al.*, 1991) and re-sampling errors (Birchfield and Tomasi, 1998), but we have not yet done this.

4.2 Flow compatibility

The controlled flow compatibility constraint,

$$\mathcal{C}_T(\{\mathbf{u}_s\}) = \sum_{s \in S} \sum_{t \in \mathcal{N}(s) \cap S} w_{st} \sum_{\mathbf{x}_s} v_{st}(\mathbf{x}_s) c_{st}(\mathbf{x}_s), \tag{9}$$

with

$$c_{st}(\mathbf{x}_s) = \rho_T \left(|\mathbf{u}_s(\mathbf{x}_s) - \mathbf{u}_t(\mathbf{x}_t)|; \sigma_T, \epsilon_T, \right) \tag{10}$$

enforces *mutual consistency* between flow estimates at different neighboring keyframes.

For the constant flow motion model, the variance σ_T can be used to account for drift in the velocities (acceleration). For a rigid scene, we don't expect any drift. However, the d_s's may actually be related by a projective transformation (Shade *et al.*, 1998). For a scene with object far enough away or for cameras arranged in a plane perpendicular to their optical axes, the inverse depths to corresponding pixels are close enough that this is not a problem.

4.3 Flow smoothness

The final cost term we use is a controlled flow smoothness constraint,

$$\mathcal{C}_S(\{\mathbf{u}_s\}) = \sum_{s \in S} \sum_{\mathbf{x}_s} f_s(\mathbf{x}_s), \tag{11}$$

with

$$f_s(\mathbf{x}_s) = \sum_{\mathbf{x}' \in \mathcal{N}_4(\mathbf{x})} \rho_S(|\mathbf{u}_s(\mathbf{x}) - \mathbf{u}_s(\mathbf{x}')|; \sigma_S, \epsilon_S(\mathbf{x}, \mathbf{x}')). \tag{12}$$

[4] We could further generalize the bias and gain fields to be slowly varying functions across the image (Wells *et al.*, 1994).

The value of the outlier probability is based on the brightness/color difference between neighboring pixels

$$\epsilon_S(\mathbf{x}, \mathbf{x}') = \Psi(I_s(\mathbf{x}) - I_s(\mathbf{x}')).$$

Appendix A derives the form of this Ψ function and justifies the dependence of the outlier probability on the local intensity variation.

4.4 Determining visibility

One of the most novel aspects of our multi-view matching framework is the explicit use of visibility to prevent the matching of pixels into areas which are occluded. This kind of visibility computation is commonly used in a number of computer graphics algorithms, e.g., algorithms for computing shadows and algorithms for recovering texture maps from images. It is also used in some more recent stereo matching algorithms (Seitz and Dyer, 1997; Szeliski and Golland, 1998; Baker *et al.*, 1998).

When working with rigid motion and depth/disparity estimates, the visibility computation is fairly straightforward. Consider two images, I_s and I_t. We wish to compute whether pixel $\mathbf{x}_s$ in image I_s is visible at location $\mathbf{x}_t$ in image I_t. If $\mathbf{x}_s$ is visible, the values of $d_s(\mathbf{x}_s)$ and $d_t(\mathbf{x}_t)$ should be the same.[5] If $\mathbf{x}_s$ is occluded, then $d_t(\mathbf{x}_t) > d_s(\mathbf{x}_s)$ (assuming $d = 0$ at infinity and positive elsewhere in front of the camera). We therefore have

$$v_{st}(\mathbf{x}_s) = ((d_t(\mathbf{x}_t) - d_s(\mathbf{x}_s)) \leq \delta), \tag{13}$$

where δ is a threshold to account for errors in estimation and warping. Note that v is generally not commutative, e.g., $v_{st}(\mathbf{x}_s)$ may not be the same as $v_{ts}(\mathbf{x}_t)$, since $\mathbf{x}_t$ may map to a different pixel $\mathbf{x}'_s$ if it is an occluder. Note that we should also never have $(d_s(\mathbf{x}_s) - d_t(\mathbf{x}_t)) > \delta$, since an occluder will not map to the same pixel as its occludee. Fortunately, the flow compatibility constraint should ensure that this does not happen.

[5] See the discussion in Section 4.2 of how disparities may have to be re-mapped between images in certain camera configurations.

In cases where not all frames are keyframes, we may have images I_t without associated $\mathbf{u}_t$ estimates. In this case, we can forward warp motion estimates from neighboring keyframes, using z-buffering to resolve ambiguities when several pixels map to the same destination. A more detailed explanation of such a warping algorithm is given in (Shade *et al.*, 1998).

When we have general 2-D flow, the situation is more complicated. In general, we cannot determine whether an occluding layer will be moving slower or faster than a pixel in a occluded layer. (For a translating camera, we can always pan the camera is such a way that either case occurs.) Therefore, the best we can do is to simply compare the flow estimates, and infer that a pixel may be invisible if the two velocities disagree,

$$v_{st}(\mathbf{x}_s) = (\|\mathbf{u}_s(\mathbf{x}_s) - \mathbf{u}_t(\mathbf{x}_t)\| \leq \delta). \tag{14}$$

Regardless of the motion model, we also set $v_{st}(\mathbf{x}_s) = 0$ whenever the corresponding pixel $\mathbf{x}_t$ is outside the boundaries of I_t, i.e., $\mathbf{x}_t \notin I_t$. (We can think of the camera body as being the occluder, in this case.)

5 Estimation algorithm

With our cost framework in place, we now describe our estimation algorithm.

In order to determine the best possible algorithm characteristics and to compare different design choices and algorithm components, we have developed a general-purpose framework which combines ideas from hierarchical estimation (Quam, 1984; Bergen *et al.*, 1992), correlation-style search (Matthies *et al.*, 1989; Kanade and Okutomi, 1994), and sub-pixel motion/disparity estimation (Lucas and Kanade, 1981; Matthies *et al.*, 1989). We have also added a few new twists, which we describe below.

Our algorithm operates in two phases. During an initialization phase, we estimate the flows independently for each keyframe. Since we do not yet have any good motion estimates for other frames, the flow compatibility term $\mathcal{C}_T$ is ignored, and no visibilities are computed (i.e., $v_{st} = 1$). In the second phase, we enforce flow compatibility and compute visibilities based on the current collection of flow estimates $\{\mathbf{u}_s\}$.

5.1 Computing initial estimates

Our algorithm is hierarchical, i.e., the matching can occur at any level in a multi-resolution pyramid, and results from coarser levels can be used to initialize estimates at a finer level.[6] Hierarchical matching results in a more efficient algorithm, since fewer pixels are examined at coarser levels, and potentially results in better quality estimates, since a wider range of motions can be searched and a better local minimum can be found.

Within each level, we can use one of two techniques to improve the current estimates: explicit correlation-style search, or Lucas-Kanade style gradient descent.

5.1.1 Correlation-style search

In correlation-style search, we evaluate several motion or disparity hypotheses at once, and then locally pick the one which results in the lowest local cost function. To rank the hypotheses, we evaluate the local error function $e_{st}(\mathbf{x}_s, \hat{\mathbf{u}}_s)$ given in Equation (8) (the dependence on $\hat{\mathbf{u}}_s$ is made explicit). The flow hypotheses $\hat{\mathbf{u}}_s$ are obtained from

$$\hat{\mathbf{u}}_s = \mathbf{u}_s + \boldsymbol{\Delta}\mathbf{u}_s,$$

where $\mathbf{u}_s$ is the current estimate, $\boldsymbol{\Delta}\mathbf{u}_s = (iS, jS)$, S is a step size, and $i = -N \ldots N, j = -N \ldots N$ is a $(2N+1) \times (2N+1)$ search window. For rigid motion, only a 1-D search window of size $(2N+1)$ over possible d values is used.[7]

In other words, we take the current flow field $\mathbf{u}_s$ and add a fixed step in (u, v) before performing the re-sampling (warping) of image I_t (see Table 1). This is similar to the iterative re-warping algorithms described in (Bergen *et al.*, 1992; Szeliski and Coughlan, 1997), as opposed to algorithms based on shifting a square correlation window (Lucas and Kanade, 1981; Matthies *et al.*, 1989; Kanade and Okutomi, 1994). Note, however, that if the initial (current) flow estimate is 0, the behavior of this part of the algorithm is the same as that of a simple correlation window or of a

[6] In the case of 2-D flow, we double the velocities when transferring to a finer level. For rigid motion, we adjust the $\mathbf{M}_{ts}$ and $\mathbf{e}_{ts}$ global parameters for each level.

[7] Furthermore, only non-negative disparities are ever evaluated, since negative disparities lie behind the viewer, assuming that $\mathbf{M}_{st}$ is the plane at infinity.

plane sweep algorithm (Collins, 1996; Scharstein and Szeliski, 1998; Szeliski and Golland, 1998). The advantage of iterative warping is that it results in better matches (and hence, more accurate estimates) in regions with severe foreshortening or inhomogeneous motion.

The values of the local error function $e_{st}(\mathbf{x}_s, \hat{\mathbf{u}}_s)$ are usually not sufficient to reliably determine a winning $\hat{\mathbf{u}}_s$ at each pixel. Traditionally, two approaches have been used to overcome this problem. The first is to aggregate evidence spatially, using square windows (of potentially variable size) (Kanade and Okutomi, 1994), convolution, pyramid-based smoothing (Bergen *et al.*, 1992), or non-linear diffusion (Scharstein and Szeliski, 1998). Our current implementation uses spatial convolution

$$\tilde{e}_{st}(\mathbf{x}_s, \hat{\mathbf{u}}_s) = e_{st}(\mathbf{x}_s, \hat{\mathbf{u}}_s) * W(\mathbf{x}), \tag{15}$$

where $W(\mathbf{x})$ is a convolution kernel. In our current implementation, we use iterated convolution with a separable $(1/4, 1/2, 1/4)$ kernel.[8]

The other major approach to local ambiguity is the use of smoothness constraints (Horn and Schunck, 1981; Terzopoulos, 1986b). Our algorithm uses the smoothness constraints described in Section 4.3. In our current implementation, we disable smoothness constraints when performing the initial estimate (with $\mathbf{u}_s = 0$), and then enable them and reduce the amount of spatial aggregation. Note that for the smoothness constraint to be meaningful, we evaluate $f_s(\mathbf{x}_s, \hat{\mathbf{u}}_s)$ with the neighboring values of $\mathbf{u}_s$ set to their current (rather than hypothesized) values. Also, in order to prevent the introduction of oscillations due to the coupled nature of the smoothness constraint, we add an extra cost term $\alpha\rho_s(\hat{\mathbf{u}}_s, \mathbf{u}_s)$, with $0 \leq \alpha \leq 4$.

To find the best flow hypothesis at each pixel, we sum up the spatially aggregated error function for each temporal neighbor and add in the smoothness term to obtain a local cost function

$$\mathcal{C}_L(\mathbf{x}_s, \hat{\mathbf{u}}_s) = \sum_{t \in \mathcal{N}(s)} w_{st}\tilde{e}_{st}(\mathbf{x}_s, \hat{\mathbf{u}}_s) + f_s(\mathbf{x}_s, \hat{\mathbf{u}}_s), \tag{16}$$

Based on this set of cost estimates, $\tilde{C}_L(\mathbf{x}_s, \hat{\mathbf{u}}_s), \hat{\mathbf{u}}_s \in \mathcal{H}$, where $\mathcal{H}$ is our set of new motion hypotheses, we pick the $\hat{\mathbf{u}}_s$ with the lowest (best) cost at each pixel. (This corresponds to the "winner-take-all" step of many stereo algorithms.)

[8] If the number of iterations is 0, no spatial aggregation is performed.

To obtain motion estimates with better accuracy, we compute a *fractional* motion estimate by fitting a quadratic cost function to the cost function values around the minimum and analytically computing its minimum (Matthies *et al.*, 1989). For 2-D flow, we use the minimum cost hypothesis along with 5 of its 8 $\mathcal{N}_8$ neighbors to fit the quadratic. We also disable fractional disparity fitting if the distance of the analytic minimum from the discrete minimum is more than a $1/2$ step.

We can also estimate local confidence by finding the minimum eigenvalue (curvature or second derivative) of the local quadratic fit and dividing it by the minimum error, which is an estimate of the noise variance (Matthies *et al.*, 1989). This is useful when generating "sparse" or "confidence-weighted" motion/disparity estimates, e.g., for the later integration of multiple depth maps. It is also useful when trying to initialize a layered representation on the basis of these (noisy and uncertain) depth estimates (Torr *et al.*, 1999).

5.1.2 Gradient descent

An alternative to explicit correlation-style search over discrete motion hypothesis is to perform gradient descent on the local cost function (Lucas and Kanade, 1981; Bergen *et al.*, 1992; Szeliski and Coughlan, 1997). Derivatives of the terms in the local cost function are taken with respect to infinitessimal changes in both the horizontal and vertical motion components (for 2-D flow) or in disparity (rigid motion). These terms involve image gradients, differences between neighboring or corresponding flow estimates (for f_s and c_{st}), and derivatives of the robust functions (Black and Rangarajan, 1996; Sawhney and Ayer, 1996). Outer products of these derivatives are taken and aggregated spatially and temporally, just as in correlation-style search. Finally a 2×2 linear system is solved at each pixel to determine the local change in motion. We omit details of these steps, since they are readily derivable from the equations already presented (see, e.g., (Bergen *et al.*, 1992)).

5.2 Multi-View Estimation

Once we have computed an initial set of motion estimates $\{\mathbf{u}_s\}$, we can now compute visibilities $v_{st}(\mathbf{x}_s)$ and add in the motion compatibility constraint $\mathcal{C}_T$. For those pixels which are not visible in a given frame, i.e., $v_{st}(\mathbf{x}_s) = 0$, what cost function should we assign? This issue arises not

only when performing multi-view estimation, but even in the initial independent motion estimation stage, whenever pixels are mapped outside the boundaries of an image, i.e., $\mathbf{x}_t \notin I_t$.[9]

One possibility is to not pay any penalty, i.e., to set $e_{st}(\mathbf{x}_s, \hat{\mathbf{u}}_s)$ (and c_{st}) to 0 whenever $v_{st}(\mathbf{x}_s) = 0$. Unfortunately, this encourages pixels near image borders to have large, outward-going flows. Another possibility is to set $e_{st}(\mathbf{x}_s, \hat{\mathbf{u}}_s) = \rho(\infty)$. Unfortunately, this encourages pixels near image borders to have inward-directed flows.

The solution we have devised is to use the visibility field v_{st} as a mask for a morphological fill operation. In other words, we replace entries in $\mathcal{C}_L(\mathbf{x}_s, \hat{\mathbf{u}}_s)$ with their neighbors' values whenever $v_{st}(\mathbf{x}_s) = 0$. The actual filling algorithm we use is a variant of the multiresolution push/pull algorithm described in (Gortler *et al.*, 1996). Thus, to truly reflect our current implementation, Equation (7) (and similarly, (9)) should replace the $v_{st}e_{st}$ term with a $\hat{e}_{st}$ term that denotes the *filled* error function.

The multi-view estimation algorithm can be repeated several times, at each iteration obtaining better estimates of motion and visibility. We currently perform this *sweeping* through the keyframes, instead of performing a single global estimation, because it is easier to implement and requires less memory. The alternative of performing a full spatio-temporal regularization is discussed in Section 7.

5.3 Algorithm summary

Table 1 summarizes a single sweep of our multi-view estimation algorithm. The algorithm iterates in a coarse-to-fine hierarchy, using the results from the coarser level or previous sweep to initialize the flow estimates (0 flow is assumed to start with). The algorithm can also be run without a coarse-to-fine hierarchy, i.e., with $L_{\max} = L_{\min}$.

Within each level, a number of iterations is performed. Each iteration consists of generating some motion increment hypotheses. These correspond to a block (correlation-style) search in motion estimation, or the small motion refinement in a coarse-to-fine algorithm (Anandan, 1989). Alternatively, a complete plane sweep can be evaluated at this step.

[9] We glossed over this fact in the previous subsection, but we will rectify this omission now.

for $l = L_{\max}$ down to $L_{\min}$

  for all $s \in S$

    initialize flow $\mathbf{u}_s$ (previous frame, level, or 0)

    for $iter$ = 1 to N_{iter}

      for $\mathbf{\Delta u}_s \in \mathcal{H}$

        for $t \in \mathcal{N}(s)$

          Warp $\tilde{I}_{ts}(\mathbf{x}_s) \leftarrow I_t(\mathbf{x}_t(\mathbf{x}_s; \mathbf{u}_s + \mathbf{\Delta u}_s))$

          Warp $\tilde{\mathbf{u}}_{ts}(\mathbf{x}_s) \leftarrow \mathbf{u}_t(\mathbf{x}_t(\mathbf{x}_s; \mathbf{u}_s + \mathbf{\Delta u}_s))$

          Compute visibility based on $\mathbf{u}_s - \tilde{\mathbf{u}}_{ts}$

          Compute robust error $\rho_I(|I_s - \tilde{I}_{ts}|)$

          Add flow compatibility $\rho_T(|\mathbf{u}_s + \mathbf{\Delta u} - \tilde{\mathbf{u}}_{ts}|)$

          Optionally compute gradients and Hessians

          Aggregate spatially

        Accumulate (weighted) over all t

        Add flow smoothness $\rho_S(|\nabla(\mathbf{u}_s + \mathbf{\Delta u})|; |\nabla I_s|)$

      next $\mathbf{\Delta u}_s$

      Pick best $\mathbf{\Delta u}_s^*$ at each pixel

      Compute sub-pixel $\widehat{\mathbf{\Delta u}_s}$

      Update $\mathbf{u}_s \leftarrow \mathbf{u}_s + \widehat{\mathbf{\Delta u}_s}$

    next $iter$

  next s

next l

Table 1: Algorithm summary (single sweep)

For each hypothesis, the neighboring frames (both intensities and flows) are warped using the new motion hypothesis. These warped values are compared to the current frame's values, and robust penalties are computed. These penalties are selectively disabled based on the visibility computation that results from comparing the warped and current flow estimates. The penalties are then optionally aggregated spatially. Finally, smootheness penalties are optionally added.

The best Δu_s hypothesis is then picked independently at each pixel, and that pixel's motion is updated. A fractional update can be performed using the costs associated with surrounding hypotheses. The process of evaluating and picking motion update hypotheses is continued for a number of iterations, and eventually executed for all keyframes, in a complete coarse-to-fine hierarchy.

6 Experiments

We have applied our multi-view matching algorithm to a number of image sequences, both where the camera motion is known (based on tracking points and computing structure from motion), and where the flow is uniform over time (video sequences). Figures 3 and 4 show some representative results and illustrate some of the features of our algorithm.

In both sets of figures, images (a–c) show the first, middle, and last image in the sequence (we used the first 4 even images from the *flower garden* sequence and 5 out of 40 images from the *symposium* sequence). The depth maps estimated by the initial, independent analysis algorithm (Section 5.1) are shown in images (e–g). The final results of applying our multi-view estimation algorithm (Section 5.2) with flow smoothness, flow compatibility, and visibility estimation are shown in images (i–k). Notice the improved quality of the estimates obtained with the multi-view estimation algorithm, especially in regions that are partially occluded. For example, in Figure 3, since the tree is moving from right to left, the occluded region is to the left of the tree in the first image, and to the right of the tree in the last one. Notice how the opposite edge of the trunk (where disocclusions are occurring) looks "crisp".

Image (d) in both Figures shows the results of warping one image based on the flow computed

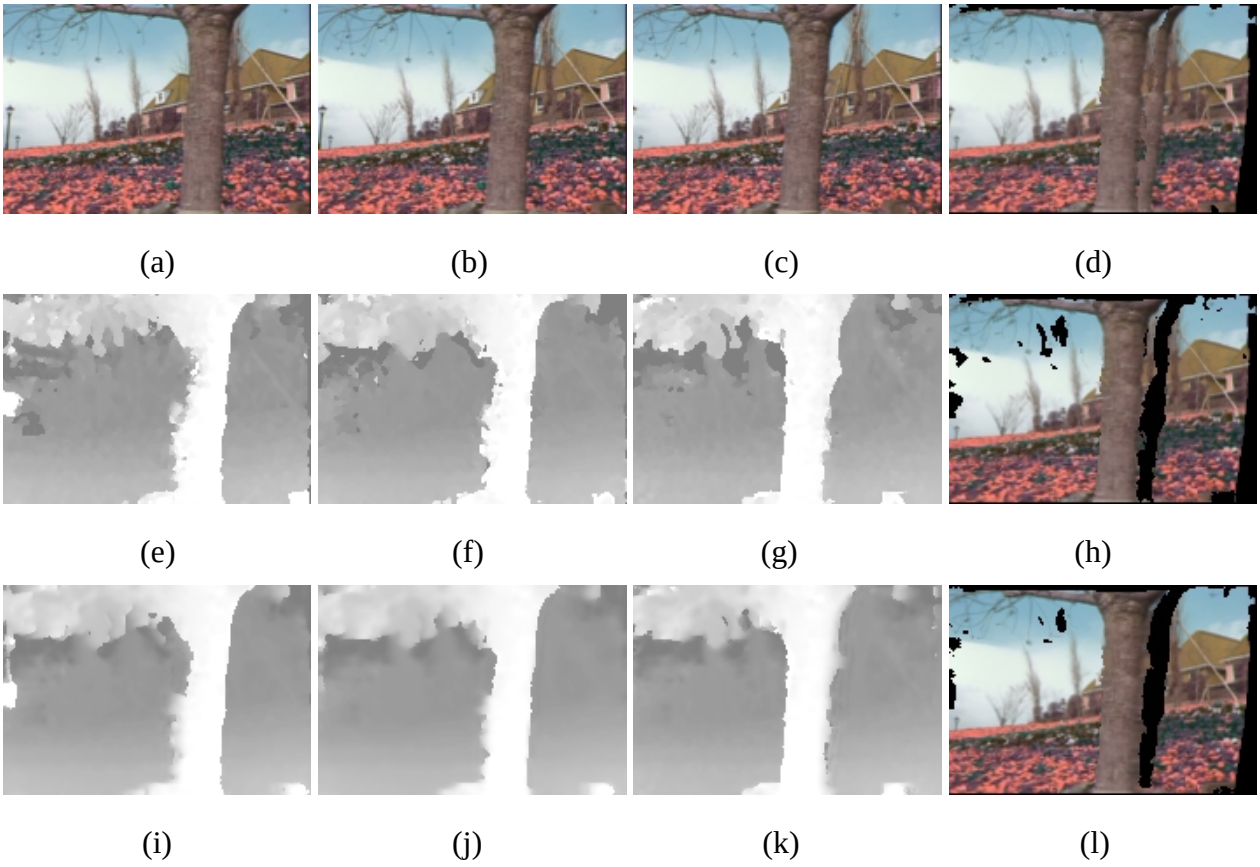
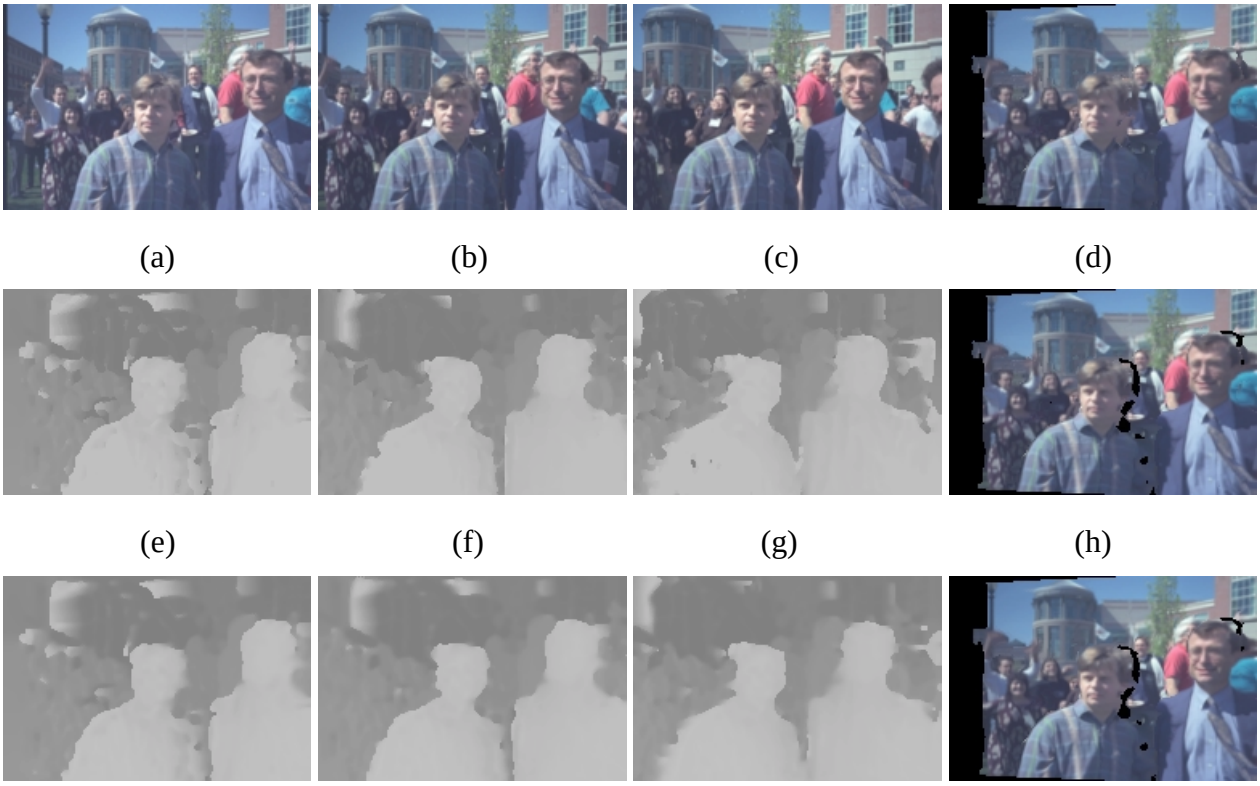


Figure 3: Results on the *flower garden* sequence: (a–c) first, second, and fourth (last) frame; (e–g) initial depth estimates; (i–k) refined (multi-view) depth estimates. Warped (resampled) images: (d) after initial estimate; (h) with visibility computation; (l) with refined estimates.

in another image. Displaying these warped images as the algorithm progresses is a very useful way to debug the algorithm and to assess the quality of the motion estimates.[10] Without visibility computation, image (d) shows how the pixels in occluded regions draw their colors somewhere from the foreground regions (e.g., the tree trunk in Figure 3 and the people's heads in Figure 4).

Images (h) and (i) show the warped images with invisible pixels flagged as black (the images were generated after the initial and final estimation stages, and hence correspond to the flow fields shown to their left). Notice how the algorithm correctly labels most of the occluded pixels, es-

[10]For the *Yosemite* sequence, it shows that the *correct flows* supplied with the sequence fail to completely stabilize the images.



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

(e) (f) (g) (h)

(i) (j) (k) (l)

Figure 4: Results on the *symposium* sequence: (a–c) first, third, and fifth (last) frame; (e–g) initial depth estimates; (i–k) refined (multi-view) depth estimates. Warped (resampled) images: (d) after initial estimate; (h) with visibility computation; (l) with refined estimates.

pecially after the final estimation. Notice, also, that some regions without texture such as the sky sometimes erroneously indicate occlusion. Using more smoothing or adding a check that occluder and occludees have different colors could be used to eliminate this problem (which is actually harmless, if we are using our matcher for view interpolation or motion prediction applications).

7 Discussion

The experimental results we have obtained so far are encouraging, but still leave room for improvement. In particular, the smoothness of the final estimates and the *sharpness* of the motion

discontinuities is not as high as that obtainable with layered motion estimates (Wang and Adelson, 1993; Weiss, 1997; Baker *et al.*, 1998). This is particularly true in occluded regions: layered models will apply the layer's motion to the occluded regions, while we use a weak smoothness constraint.

In future work, we would like to apply our multi-view framework to view interpolation and motion-based prediction. Since a novel view does not come with an associated $\mathbf{u}_t$ map, we must first forward warp motion estimates from neighboring keyframes, using z-buffering to resolve ambiguities when several pixels map to the same destination (Shade *et al.*, 1998). The novel view can then be generated as a *blend* of original views, taking into account pixel visibilities.

We would also like to add super-resolution and de-noising to our multi-view estimation framework. Rather than optimizing our original brightness compatibility constraint, we would minimize $\rho(\hat{I}_s(\mathbf{x}_s) - I_t(\mathbf{x}_t))$ where $\hat{I}_s$ is a de-noised and possibly super-resolved image (texture) estimate.[11] We would also like to investigate explicitly representing discontinuities, since these are where many of the errors occur.

8 Conclusions

In this paper, we have developed a novel multi-view framework for estimating dense motion and correspondence maps. Our framework simultaneously produces estimates for a subset of the input images, thereby representing motion or depth in partially occluded regions and explicitly modeling the variation in appearance between different views. Our framework is based on minimizing a three part cost function, which consists of a brightness compatibility, a motion compatibility, and a motion smoothness term. Our novel motion smoothness terms uses the presence of color/brightness discontinuities to modify the probability of motion smoothness violation (outlier). We have also developed some novel techniques for determining the visibility of pixels in neighboring images, and used this visibility to affect the values of the brightness and motion compatibility constraints.

While our preliminary experimental results look encouraging, there remains much work to be done in developing truly accurate and robust correspondence algorithms. We believe that the

[11] If the images are at different resolutions, proper downsampling of $\hat{I}_s$ should be applied.

development of such algorithms will be crucial in promoting a wider use of image-based modeling in novel applications such as virtual environments, image-based rendering, and video processing.

References

Anandan, P. 1989. A computational framework and an algorithm for the measurement of visual motion. *International Journal of Computer Vision, 2(3)*, 283–310.

Arnold, R. D. 1983. *Automated Stereo Perception*. Technical Report AIM-351, Artificial Intelligence Laboratory, Stanford University.

Baker, H. H. 1980. Edge based stereo correlation. In Baumann, L. S., editor, *Image Understanding Workshop*, pages 168–175, Science Applications International Corporation.

Baker, S., Szeliski, R., and Anandan, P. 1998. A layered approach to stereo reconstruction. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'98)*, pages 434–441, Santa Barbara.

Barnard, S. T. 1989. Stochastic stereo matching over scale. *International Journal of Computer Vision, 3(1)*, 17–32.

Barnard, S. T. and Fischler, M. A. 1982. Computational stereo. *Computing Surveys, 14(4)*, 553–572.

Belhumeur, P. N. and Mumford, D. 1992. A Bayesian treatment of the stereo correspondence problem using half-occluded regions. In *Computer Vision and Pattern Recognition*, pages 506–512, Champaign-Urbana, Illinois.

Bergen, J. R., Anandan, P., Hanna, K. J., and Hingorani, R. 1992. Hierarchical model-based motion estimation. In *Second European Conference on Computer Vision (ECCV'92)*, pages 237–252, Springer-Verlag, Santa Margherita Liguere, Italy.

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)*, 401–406.

Black, M. and Anandan, P. 1991. Robust dynamic motion estimation over time. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'91)*, pages 296–302, IEEE Computer Society Press, Maui, Hawaii.

Black, M. J. and Jepson, A. D. 1996. Estimating optical flow in segmented images using variable-order parametric models with local deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *18(10)*, 972–986.

Black, M. J. and Rangarajan, A. 1996. On the unification of line processes, outlier rejection, and robust statistics with applications in early vision. *International Journal of Computer Vision*, *19(1)*, 57–91.

Blonde, L. *et al.*. 1996. A virtual studio for live broadcasting: The Mona Lisa project. *IEEE Multimedia, 3(2)*, 18–29.

Bolles, R. C., Baker, H. H., and Marimont, D. H. 1987. Epipolar-plane image analysis: An approach to determining structure from motion. *International Journal of Computer Vision, 1*, 7–55.

Collins, R. T. 1996. A space-sweep approach to true multi-image matching. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'96)*, pages 358–363, San Francisco, California.

Cox, I. J. 1994. A maximum likelihood n-camera stereo algorithm. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'94)*, pages 733–739, IEEE Computer Society, Seattle, Washington.

de Hann, G. and Beller, E. B. 1998. Deinterlacing—an overview. *Proceedings of the IEEE, 86(9)*, 1839–1857.

Debevec, P. E., Taylor, C. J., and Malik, J. 1996. Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach. *Computer Graphics (SIGGRAPH'96)*, , 11–20.

Dhond, U. R. and Aggarwal, J. K. 1989. Structure from stereo—a review. *IEEE Transactions on Systems, Man, and Cybernetics, 19(6)*, 1489–1510.

Fua, P. 1993. A parallel stereo algorithm that produces dense depth maps and preserves image features. *Machine Vision and Applications, 6*, 35–49.

Gamble, E. and Poggio, T. 1987. *Visual integration and detection of discontinuities: the key role of intensity edges*. A. I. Memo 970, Artificial Intelligence Laboratory, Massachusetts Institute of Technology.

Geiger, D. and Girosi, F. 1991. Parallel and deterministic algorithms for MRF's: Surface reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 13(5)*, 401–412.

Geiger, D., Ladendorf, B., and Yuille, A. 1992. Occlusions and binocular stereo. In *Second European Conference on Computer Vision (ECCV'92)*, pages 425–433, Springer-Verlag, Santa Margherita Liguere, Italy.

Gennert, M. A. 1988. Brightness-based stereo matching. In *Second International Conference on Computer Vision (ICCV'88)*, pages 139–143, IEEE Computer Society Press, Tampa, Florida.

Gortler, S. J., Grzeszczuk, R., Szeliski, R., and Cohen, M. F. 1996. The lumigraph. In *Computer Graphics Proceedings, Annual Conference Series*, pages 43–54, ACM SIGGRAPH, Proc. SIGGRAPH'96 (New Orleans).

Heeger, D. J. 1988. Optical flow using spatiotemporal filters. *International Journal of Computer Vision, 1*, 279–302.

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

Intille, S. S. and Bobick, A. F. 1994. Disparity-space images and large occlusion stereo. In *Proc. Third European Conference on Computer Vision (ECCV'94)*, Springer-Verlag, Stockholm, Sweden.

Irani, M., Anandan, P., and Hsu, S. 1995. Mosaic based representations of video sequences and their applications. In *Fifth International Conference on Computer Vision (ICCV'95)*, pages 605–611, Cambridge, Massachusetts.

Jenkin, M. R. M., Jepson, A. D., and Tsotsos, J. K. 1991. Techniques for disparity measurement. *CVGIP: Image Understanding, 53(1)*, 14–30.

Jones, D. G. and Malik, J. 1992. A computational framework for determining stereo correspondence from a set of linear spatial filters. In *Second European Conference on Computer Vision (ECCV'92)*, pages 397–410, Springer-Verlag, Santa Margherita Liguere, Italy.

Kanade, T. and Okutomi, M. 1994. A stereo matching algorithm with an adaptive window: Theory and experiment. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 16(9)*, 920–932.

Kanade, T. *et al.*. 1996. A stereo machine for video-rate dense depth mapping and its new applications. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'96)*, pages 196–202, San Francisco, California.

Kang, S. B., Webb, J., Zitnick, L., and Kanade, T. 1995. A multibaseline stereo system with active illumination and real-time image acquisition. In *Fifth International Conference on Computer Vision (ICCV'95)*, pages 88–93, Cambridge, Massachusetts.

Kumar, R., Anandan, P., and Hanna, K. 1994. Direct recovery of shape from multiple views: A parallax based approach. In *Twelfth International Conference on Pattern Recognition (ICPR'94)*, pages 685–688, IEEE Computer Society Press, Jerusalem, Israel.

Le Gall, D. 1991. MPEG: A video compression standard for multimedia applications. *Communications of the ACM, 34(4)*, 44–58.

Leclerc, Y. G. 1989. Constructing simple stable descriptions for image partitioning. *International Journal of Computer Vision*, *3(1)*, 73–102.

Lee, M.-C. *et al.*. 1997. A layered video object coding system using sprite and affine motion model. *IEEE Transactions on Circuits and Systems for Video Technology*, *7(1)*, 130–145.

Lucas, B. D. and Kanade, T. 1981. An iterative image registration technique with an application in stereo vision. In *Seventh International Joint Conference on Artificial Intelligence (IJCAI-81)*, pages 674–679, Vancouver.

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

Matthies, L. H., Szeliski, R., and Kanade, T. 1989. Kalman filter-based algorithms for estimating depth from image sequences. *International Journal of Computer Vision*, *3*, 209–236.

McMillan, L. and Bishop, G. 1995. Plenoptic modeling: An image-based rendering system. *Computer Graphics (SIGGRAPH'95)*, , 39–46.

Ohta, Y. and Kanade, T. 1985. Stereo by intra- and inter-scanline search using dynamic programming. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *PAMI-7(2)*, 139–154.

Okutomi, M. and Kanade, T. 1993. A multiple baseline stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *15(4)*, 353–363.

Pollard, S. B., Mayhew, J. E. W., and Frisby, J. P. 1985. PMF: A stereo correspondence algorithm using a disparity gradient limit. *Perception*, *14*, 449–470.

Prazdny, K. 1985. Detection of binocular disparities. *Biological Cybernetics*, *52*, 93–99.

Quam, L. H. 1984. Hierarchical warp stereo. In *Image Understanding Workshop*, pages 149–155, Science Applications International Corporation, New Orleans, Louisiana.

Ryan, T. W., Gray, R. T., and Hunt, B. R. 1980. Prediction of correlation errors in stereo-pair images. *Optical Engineering*, *19(3)*, 312–322.

Sawhney, H. S. 1994. 3D geometry from planar parallax. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'94)*, pages 929–934, IEEE Computer Society, Seattle, Washington.

Sawhney, H. S. and Ayer, S. 1996. Compact representation of videos through dominant multiple motion estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *18(8)*, 814–830.

Scharstein, D. and Szeliski, R. 1998. Stereo matching with nonlinear diffusion. *International Journal of Computer Vision*, *28(2)*, 155–174.

Seitz, S. M. and Dyer, C. M. 1997. Photorealistic scene reconstrcution by space coloring. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'97)*, pages 1067–1073, San Juan, Puerto Rico.

Shade, J., Gortler, S., He, L.-W., and Szeliski, R. 1998. Layered depth images. In *Computer Graphics (SIGGRAPH'98) Proceedings*, pages 231–242, ACM SIGGRAPH, Orlando.

Simoncelli, E. P., Adelson, E. H., and Heeger, D. J. 1991. Probability distributions of optic flow. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'91)*, pages 310–315, IEEE Computer Society Press, Maui, Hawaii.

Szeliski, R. 1990. Bayesian modeling of uncertainty in low-level vision. *International Journal of Computer Vision*, *5(3)*, 271–301.

Szeliski, R. and Coughlan, J. 1997. Hierarchical spline-based image registration. *International Journal of Computer Vision*, *22(3)*, 199–218.

Szeliski, R. and Golland, P. 1998. Stereo matching with transparency and matting. In *Sixth International Conference on Computer Vision (ICCV'98)*, pages 517–524, Bombay.

Szeliski, R. and Hinton, G. 1985. Solving random-dot stereograms using the heat equation. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'85)*, pages 284–288, IEEE Computer Society Press, San Francisco, California.

Szeliski, R. and Kang, S. B. 1995. Direct methods for visual scene reconstruction. In *IEEE Workshop on Representations of Visual Scenes*, pages 26–33, Cambridge, Massachusetts.

Terzopoulos, D. 1986a. Image analysis using multigrid relaxation methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-8(2)*, 129–139.

Terzopoulos, D. 1986b. Regularization of inverse visual problems involving discontinuities. *IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-8(4)*, 413–424.

Tian, Q. and Huhns, M. N. 1986. Algorithms for subpixel registration. *Computer Vision, Graphics, and Image Processing, 35*, 220–233.

Torr, P. H. S., Szeliski, R., and Anandan, P. 1999. An integrated Bayesian approach to layer extraction from image sequences. In *Seventh International Conference on Computer Vision (ICCV'98)*, Kerkyra, Greece.

Wang, J. Y. A. and Adelson, E. H. 1993. Layered representation for motion analysis. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'93)*, pages 361–366, New York, New York.

Weiss, Y. 1997. Smoothness in layers: Motion segmentation using nonparametric mixture estimation. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'97)*, pages 520–526, San Juan, Puerto Rico.

Weiss, Y. and Adelson, E. H. 1996. A unified mixture framework for motion segmentation: Incorporating spatial coherence and estimating the number of models. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'96)*, pages 321–326, San Francisco, California.

Wells, W., Kikinis, R., Grimson, W., and Jolesz, F. 1994. Statistical intensity correction and segmentation of magnetic resonance image data. In *Third Conference on Visualization in Biomedical Computing*, Rochester.

Yang, Y., Yuille, A., and Lu, J. 1993. Local, global, and multilevel stereo matching. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'93)*, pages 274–279, IEEE Computer Society, New York, New York.

Yuille, A. L. and Poggio, T. 1984. *A Generalized Ordering Constraint for Stereo Correspondence*. A. I. Memo 777, Artificial Intelligence Laboratory, Massachusetts Institute of Technology.

Zabih, R. and Woodfill, J. 1994. Non-parametric local transforms for computing visual correspondence. In *Proc. Third European Conference on Computer Vision (ECCV'94)*, pages 151–158.

A An image-dependent flow smoothness constraint

To develop our constraint, assume that we know the prior probability p_D that two neighboring pixels straddle a motion discontinuity (i.e., that they live on different surfaces). The distribution of the brightness or color differences between two neighboring pixels depends on the event D that they live on different surfaces, i.e., we have two distributions $p(I_s(\mathbf{x}) - I_s(\mathbf{x}')|\overline{D})$ and $p(I_s(\mathbf{x}) - I_s(\mathbf{x}')|D)$. These distributions can either be guessed (say as contaminated Gaussians, with the probability of outliers much higher in the case of D), or estimated from labelled image data.

Given these distributions and the prior probability p_D, we can apply Bayes' Rule to calculate $\Psi(I_s(\mathbf{x}) - I_s(\mathbf{x}')) = p(D|I_s(\mathbf{x}) - I_s(\mathbf{x}'))$.

$$p(D|I_s(\mathbf{x}) - I_s(\mathbf{x}')) = \frac{p_D p(I_s(\mathbf{x}) - I_s(\mathbf{x}')|D)}{p_D p(I_s(\mathbf{x}) - I_s(\mathbf{x}')|D) + (1 - p_D)p(I_s(\mathbf{x}) - I_s(\mathbf{x}')|\overline{D})}$$

(This function will typically start at some small probability ϵ_0 for small color differences, and increase to a final value ϵ_1 for large differences.) This posterior probability of a motion discontinuity can then be plugged in as the local value of ϵ_S in our controlled motion continuity constraint (12).