Efficiently Synthesizing Virtual Video

Richard Radke*
Department of Electrical, Computer, and Systems Engineering
Rensselaer Polytechnic Institute
Troy, NY 12180

Peter Ramadge, Sanjeev Kulkarni
Department of Electrical Engineering
Princeton University
Princeton, NJ 08544

Tomio Echigo
IBM Research
Tokyo Research Laboratory
1623-14 Shimotsuruma
Yamato-shi, Kanagawa, Japan

December 17, 2001

Abstract

Given a set of synchronized video sequences of a dynamic scene taken by different cameras, we address the problem of creating a virtual video of the scene from a novel viewpoint. A key aspect of our algorithm is a method for recursively propagating dense and physically accurate correspondences between the two video sources. By exploiting temporal continuity and suitably constraining the correspondences, we provide an efficient framework for synthesizing realistic virtual video. The stability of the propagation algorithm is analyzed, and experimental results are presented.

Index terms: virtual video, virtual views, view synthesis, correspondence

1 Introduction

Given a set of synchronized video sequences of a dynamic scene taken by different cameras, the virtual video problem is to synthesize video of the scene from the viewpoint of a camera not in the original set. Approaches to the problem of synthesizing a virtual view from two still images have been discussed for some time in the computer graphics community [1, 2, 4, 3, 5]. Typically, these require an estimate of a dense set of corresponding points in each pair of real images in order to synthesize a virtual image. For some view synthesis methods, a certain class of virtual images can

*This research was partially supported by grants from the IBM Tokyo Research Laboratory, the New Jersey Center for Multimedia Research, the U.S. Army Research Office under contract number DAAD19-00-1-0466, and NSF KDI under contract number ECS-9873451.

be created from a correspondence estimate alone, without explicitly calibrated cameras. In order to synthesize physically accurate virtual images, the estimated correspondence must approximate physical reality. Hence, the fundamental problem that must be solved in order to create virtual video is the estimation of the correspondence between image planes induced by the camera and scene geometry at every point in time.

In theory, the virtual video problem can be solved as an independent sequence of virtual view problems over the length of the source videos. However, this approach is prohibitively time-consuming, since estimating dense correspondence between an image pair, especially when the underlying cameras are widely spaced with respect to the scene, generally requires human intervention. Treating each pair of frames independently does not exploit the temporal continuity of the input video. That is, assuming that the motion of the cameras and scene objects is small, we expect that the correspondence required to synthesize virtual images at adjacent frames is similar.

The main contribution of this paper is a framework for efficiently synthesizing virtual video using a recursive algorithm to propagate estimates of dense correspondence between image planes from one frame to the next. The resulting video can be constructed to resemble special effects recently seen in movies [6] or televised sporting events [7] in which a camera navigates through a scene along a trajectory that seems impossible to obtain with conventional cameras. While such effects are created using a highly specialized camera rig with tens or hundreds of cameras positioned along the desired camera path, our research demonstrates that similar results can be obtained using only a few real, uncalibrated video cameras and processing on normal desktop PCs. Aside from the aforementioned hardware solutions, the only other type of virtual video we know of prior to this work was created by moving a virtual camera through a static scene generated by a single pair of still images, so that objects seem to be frozen in time. In contrast, here we create true virtual video from a pair of source video sequences, in the sense that the virtual video evolves dynamically along with the scene.

We begin in Section 2 by presenting our modeling assumptions and formally stating the vir-

tual video problem. Section 3 describes how a physically accurate correspondence estimate is represented and discusses the estimation problem for the first frame pair. In Section 4 we present the main contribution of the paper, a framework for the recursive propagation of correspondences between frames of two video sequences. The propagation consists of a time update step and a measurement update step. The time update depends only on estimating the dynamics of the source cameras, while the measurement update uses local image detail to refine the correspondence. Using these results, the correspondence estimate relating each frame pair can be propagated and updated in a fraction of the time required to estimate correspondences anew at every frame. While virtual video is our motivating application, the recursive correspondence propagation framework applies to any two-camera video application in which correspondence is difficult and prohibitively time-consuming to estimate by processing frame pairs independently. Section 4 also includes a stability analysis of the presented algorithm.

We review how the virtual images are actually synthesized given a correspondence estimate in Section 5, and demonstrate our experimental results on real test video from a natural outdoor scene in Section 6. The scene is complex, with many moving objects, yet the synthetic virtual video looks realistic and conveys a convincing 3-D effect. The user need only provide a small set of point matches in the first frame pair and an algorithm to segment and track moving objects in the scene. A shorter version of this work originally appeared in [8].

2 Problem Formulation and Approach

Our goal is to synthesize physically correct virtual images created with well-founded geometric principles— the same images that would have been seen had an actual camera been present in the original environment. Furthermore, we will synthesize virtual images in situations where strong calibration (knowledge of 3-D location and orientation) of the source cameras is unavailable. In this paper, we will confine our attention to the case when images from exactly two source cameras

are available.

We consider a pair of rotating cameras, $\mathcal{C}_0$ and $\mathcal{C}_1$, taking images of a dynamic scene. The image taken by $\mathcal{C}_k$ at time i for $i = 0, 1, 2, \ldots$ is denoted by $\mathcal{I}_k(i)$, which lies on a coordinatized image plane $\mathcal{P}_k(i)$. We assume idealized pinhole cameras that produce images by perspective projection. Our goal is to synthesize the virtual image sequence $\{\mathcal{I}_s(i), i = 0, 1, 2, \ldots\}$ of the scene from the perspective of a moving virtual camera $\mathcal{C}_s$. We discuss how $\mathcal{C}_s$ is described with respect to the source cameras $(\mathcal{C}_0, \mathcal{C}_1)$ in Section 5.

We assume the cameras' centers of projection are not coincident, so that every pair of image planes $\mathcal{P}_0(i)$ and $\mathcal{P}_1(i)$ is related by a fundamental matrix $F(i)$. That is, there exists a matrix $F(i)$ of rank two such that for all correspondences $((x_0, y_0), (x_1, y_1)) \in \mathcal{P}_0(i) \times \mathcal{P}_1(i)$, $(x_1, y_1, 1)F(i)(x_0, y_0, 1)^T = 0$. For more information on fundamental matrices, see [9].

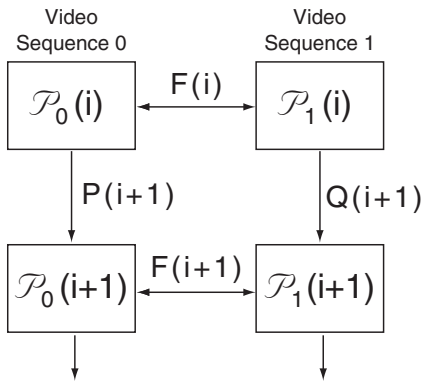

Figure 1: Relationships between image planes.

We also assume each camera's center of projection to be constant, which is reasonable in many applications where multiple cameras mounted on tripods simultaneously view a scene (e.g. sports video.) Hence, the plane coordinates of $\mathcal{P}_k(i-1)$ and $\mathcal{P}_k(i)$ are related by a projective transformation [9], denoted by $P(i)$ and $Q(i)$ for $k = 0, 1$ respectively. The various relationships between image planes are illustrated in Figure 1.

The problem of synthesizing a virtual image at time i is largely dependent on estimating a dense, physically accurate correspondence between $\mathcal{P}_0(i)$ and $\mathcal{P}_1(i)$. That is, for any point $w_0 \in$

$\mathcal{P}_0(i)$, we wish to estimate the projection of the corresponding underlying scene point in $\mathcal{P}_1(i)$, and decide whether or not the scene point is visible (i.e. unoccluded). The same applies to any point $w_1 \in \mathcal{P}_1(i)$. This is like stereo, but differs in that, in our setting, the cameras are widely separated compared with the typical stereo setup. This also differs from the problems of motion compensation in video, in which points are typically matched using photometric, not geometric, criteria, and tracking in computer vision, in which only a finite set of specific points is identified in every frame of a video sequence.

To simplify the notation, we will define $\chi^*(i)$ as the true (i.e. induced by physical reality) correspondence between the image plane pair $\mathcal{P}_0(i)$ and $\mathcal{P}_1(i)$. Our goal is to efficiently obtain an estimate of $\chi^*(i)$ at every time step. Let $\tilde{\chi}(i)$ be an estimate of $\chi^*(i)$ obtained by the application of a correspondence algorithm C^i. We assume that the application of the operator C^i is a time-consuming task, either because a lengthy search process or human intervention is required.

We wish to more efficiently estimate $\chi^*(i)$ at each time. We do so by exploiting the temporal continuity of the video, estimating the effect of camera motion, and using a computationally simpler approximation of C^i. Namely, let $\hat{\chi}(i \mid j)$ be an approximation of $\tilde{\chi}(i)$ based on information from time j, obtained by:

$$\begin{aligned} \hat{\chi}(0 \mid 0) &= \tilde{\chi}(0) \\ \hat{\chi}(i+1 \mid i) &= T^{i+1}(\hat{\chi}(i \mid i)) \\ \hat{\chi}(i+1 \mid i+1) &= M^{i+1}(\hat{\chi}(i+1 \mid i)) \end{aligned}$$

Here, T^{i+1} is a time update operator that propagates the correspondence estimate from time i to $i+1$, and M^{i+1} is a measurement update operator that refines the estimate using new information that has become available at time $i+1$. The time-dependence of the update operators arises from their dependence on the images $\mathcal{I}_0(i+1)$ and $\mathcal{I}_1(i+1)$.

To make this algorithm more concrete, we discuss the steps in more detail in the next three

sections. Section 3 specifies how a correspondence estimate is represented and discusses the initial step of estimating $\chi^*(0)$. Section 4 explains in detail the time and measurement updates and analyzes the stability of the recursive algorithm. Section 5 reviews how a virtual image is synthesized after the correspondence estimate has been obtained.

3 Initialization

Obtaining the initial correspondence estimate $\tilde{\chi}(0)$ from images taken by two uncalibrated cameras is a fundamental and difficult problem in computer vision. Many correspondence techniques are based on optical flow [10] or layered motion [11]. These techniques have the shortcoming that the estimates are typically obtained with photometric criteria that match points based entirely on the local variation of intensity between images. For the virtual view problem, as for stereo or structure from motion applications, we require a technique that attempts to find correspondence consistent with the underlying physical scene.

For an arbitrary image pair of the same scene, the only *a priori* constraint on the true correspondence is the well-known epipolar constraint [9]. The epipolar geometry can be estimated from a small number of point correspondences [12]. In theory, knowledge of the epipolar geometry reduces the correspondence problem to a series of 1-D matching problems. We mention several approaches to solving the correspondence problem in the context of conjugate epipolar lines in Appendix A. While these techniques are unstable in the small-baseline case, the virtual view problems we consider are in the wide-baseline setting, in which the cameras are widely spaced with respect to the scene.

Most epipolar-line-based correspondence algorithms make the assumption that scene points are projected onto conjugate epipolar lines in the same order. This is called the monotonicity assumption, and it is typically made so that polynomial-time algorithms can be used to efficiently obtain solutions. The result of the estimation for an epipolar line pair (ℓ_0, ℓ_1) can then be expressed

as a monotonic path through $\ell_0 \times \ell_1$, as illustrated in Figure 2. Occlusions are typically modeled as horizontal or vertical lines in the monotonic path, which is reasonable in the small-baseline case.

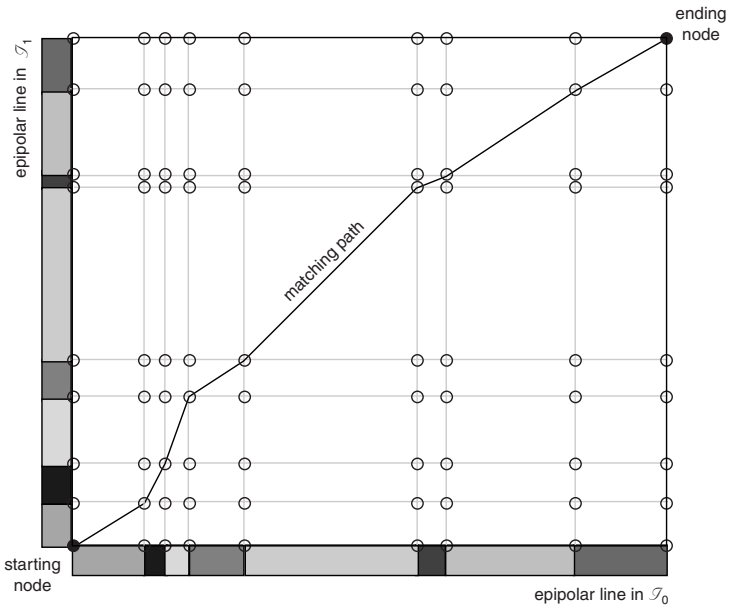

Figure 2: Matching graph for conjugate epipolar lines.

Unfortunately, the order of corresponding points along conjugate epipolar lines may not be invariant from image to image, so the monotonicity assumption is not generally valid. Figure 3 shows regions of two real images of the same scene in which the epipolar lines are horizontal and aligned. The numbered objects appear in different orders along conjugate epipolar lines due to the large perspective difference between the images. Each inconsistency in ordering generates a local violation of the monotonicity assumption in the affected conjugate epipolar lines. A monotonic path through a matching graph such as the one illustrated in Figure 2 cannot represent the correct matching. This phenomenon is sometimes called "the double nail illusion" in stereo.

Relaxing the monotonicity assumption to allow arbitrary matching of points between conjugate epipolar lines results in a problem of high combinatorial complexity. However, the set of correspondences that are physically realizable has a specific structure discussed in Appendix B and [13, 14]. We define the *correspondence graph* to be the set of all points that are visible in

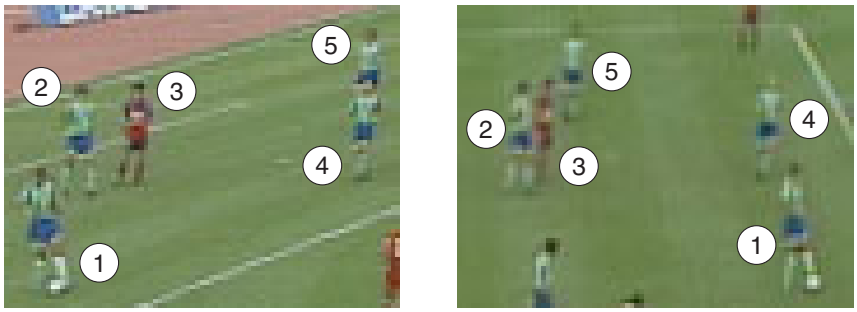

Figure 3: Violations of monotonicity.

two conjugate epipolar lines. Using the formalism of correspondence graphs, we can ensure that any estimated correspondence is consistent with a physical imaging system, which is especially important for geometric applications. The correspondence graph is generally a set of monotonic segments in $\ell_0 \times \ell_1$ and correctly takes the effects of occlusions into account (e.g. regions occluded in one epipolar line are not matched to intervals of zero length in the other.)

To illustrate the result of estimating the correspondence graph for a real image pair, consider the example illustrated in Figures 4a and 4b. These natural outdoor images of a soccer game were captured with high-quality digital video cameras. Objects which violate the monotonicity assumption were identified, segmented, and matched by hand (the segmentation is overlaid in a lighter color.) We have segmented and matched three soccer players, the soccer ball, and the uprights of the soccer goal. A typical pair of conjugate epipolar lines is displayed in white. Figure 4c shows the correspondence graph result for this line pair, comprised by the solid line segments that are unshadowed. The graph is created using a morphological operation illustrated by the "shadows" in the figure from 1) a model of the background correspondence (in this case, a projective transformation induced by the planar soccer field) and 2) the segmented objects. The dashed lines indicate regions visible in $\mathcal{I}_1$ but not in $\mathcal{I}_0$ because they are occluded or lie outside the field of view. The dotted lines indicate similar regions visible in $\mathcal{I}_0$ but not in $\mathcal{I}_1$. We can see that correspondence along this epipolar line pair is definitely not monotonic.

In summary, the initial correspondence estimate is constrained by 1) the estimated epipolar

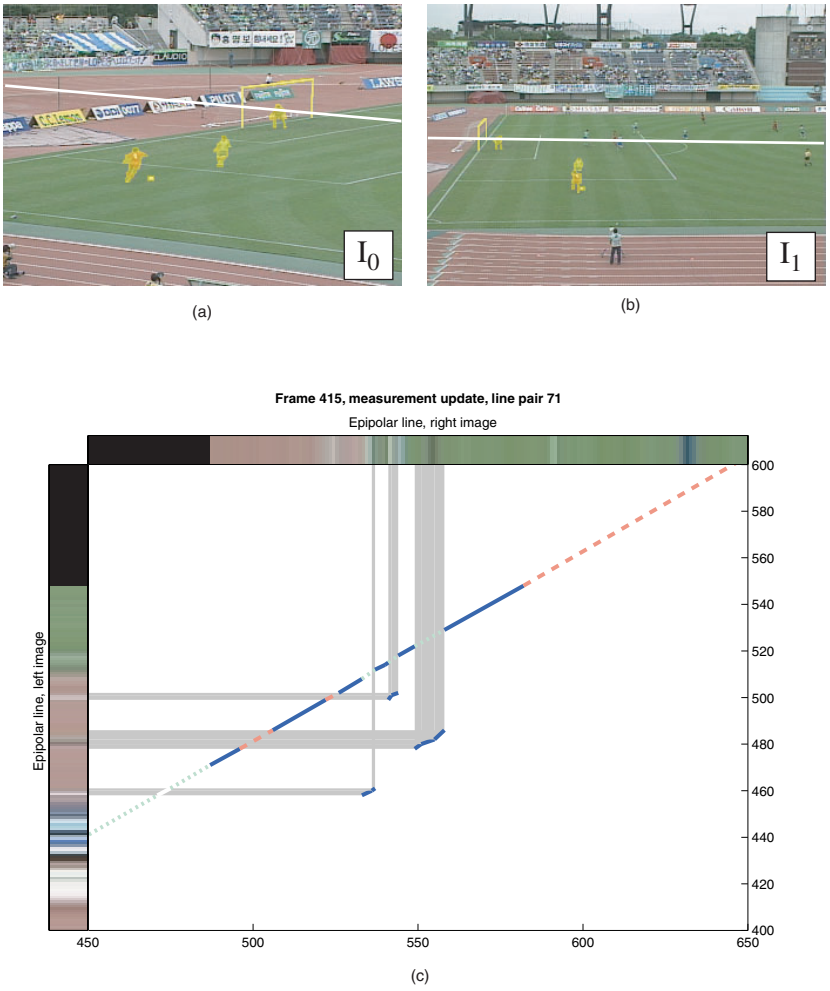

Figure 4: (a) and (b) Image pair, with overlaid segmentation and a pair of conjugate epipolar lines. (c) The estimated correspondence graph associated with the epipolar line pair is indicated by the dark line segments.

geometry, represented by the partitioning of the image planes into pairs of conjugate epipolar lines, and 2) occlusions, represented by a correspondence graph for each conjugate epipolar line pair. In the following, when we use the symbol χ to represent an estimate of correspondence, we assume that χ is a set of correspondence graphs, one for each pair of estimated conjugate epipolar lines.

4 Recursive Propagation of Correspondences

4.1 Time Update

Given a static scene and complete knowledge of the camera motion of Figure 1, the new positions at time $i+1$ of a pair of corresponding points $(w_0(i), w_1(i)) \in \mathcal{P}_0(i) \times \mathcal{P}_1(i)$ is simply $(w_0(i+1), w_1(i+1)) = (P(i+1)w_0(i), Q(i+1)w_1(i))$. That is, if the only difference between the frames at times i and $i+1$ is due to motion of the cameras, the coordinates of $\mathcal{P}_k(i)$ and $\mathcal{P}_k(i+1)$, $k = 0, 1$, are globally related by a projective transformation.

Since we describe correspondence along conjugate epipolar lines, it is often desirable to work with rectified image planes $(\bar{\mathcal{P}}_0(i), \bar{\mathcal{P}}_1(i))$ in which conjugate epipolar lines are horizontal and aligned. Rectified image planes are produced by a (nonunique) pair of projective transformations $(G(i), H(i))$ representing underlying rotations of the cameras $\mathcal{C}_0(i)$ and $\mathcal{C}_1(i)$ to new cameras $\bar{\mathcal{C}}_0(i)$ and $\bar{\mathcal{C}}_1(i)$ such that the new cameras have the same optical centers as the old ones, but whose image planes $\bar{\mathcal{P}}_0(i)$ and $\bar{\mathcal{P}}_1(i)$ lie on a plane in $\mathbb{R}^3$ parallel to the baseline. Any pair of projective transformations $(G(i), H(i))$ with this property is said to be rectifying. It can be shown (see [5]) that a necessary and sufficient condition for $(G(i), H(i))$ to be a rectifying pair is that $H(i)^{-T}F(i)G(i)^{-1} = F_*$, with $F_* = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix}$.

The time update for rectified image planes can be expressed in a particularly simple form. Suppose $(G(i), H(i))$ rectify $(\mathcal{P}_0(i), \mathcal{P}_1(i))$, so that $H(i)^{-T}F(i)G(i)^{-1} = F^*$. We would like to

choose a pair of projective transformations $(G(i+1), H(i+1))$ that rectify $(\mathcal{P}_0(i+1), \mathcal{P}_1(i+1))$. It is easily shown that one such pair is given by:

$$G(i+1) = G(i)P(i+1)^{-1} \quad (1)$$

$$H(i+1) = H(i)Q(i+1)^{-1} \quad (2)$$

Using this special rectifying pair, a point match $(\bar{w}_0(i), \bar{w}_1(i))$ from the rectified images $\bar{\mathcal{P}}_0(i) \times \bar{\mathcal{P}}_1(i)$ is propagated to the rectified images $\bar{\mathcal{P}}_0(i+1) \times \bar{\mathcal{P}}_1(i+1)$ by

$$\begin{aligned}(\bar{w}_0(i+1), \bar{w}_1(i+1)) &= (G(i+1)P(i+1)G(i)^{-1}\bar{w}_0(i), H(i+1)Q(i+1)H(i)^{-1}\bar{w}_1(i)) \\ &= (\bar{w}_0(i), \bar{w}_1(i))\end{aligned}$$

That is, the propagating transformation is simply the identity. Given that we use the rectifying pair in (1)-(2), this leads us to define the time update operator T^{i+1} that operates on a correspondence estimate χ to simply be $T^{i+1}(\chi) = \chi$. This is well-defined since the coordinates of $\bar{\mathcal{P}}_j(i)$ and $\bar{\mathcal{P}}_j(i+1)$, $j = 0, 1$ agree by construction of the rectifying projective transformations.

Of course, the various projective transformations are generally estimated using a regression algorithm, so that the estimated rectifying projective transformations $(\hat{G}(i+1), \hat{H}(i+1))$ are compositions of other estimates given by $(\hat{G}(i+1), \hat{H}(i+1)) = (\hat{G}(i)\hat{P}(i+1)^{-1}, \hat{H}(i)\hat{Q}(i+1)^{-1})$. Since there is error in these estimates, the true time update is a perturbation from the identity. In practice, we neglect this perturbation and approximate T^{i+1} by the identity; that is, we use $\hat{T}^{i+1}(\chi) = \chi$. In Section 4.3 we will analyze the implications of this approximation.

An initial rectifying pair $(G(0), H(0))$ can be obtained using one of several known methods [5, 15, 16] and a small number of point correspondences between $\mathcal{P}_0(0)$ and $\mathcal{P}_1(0)$.

4.2 Measurement Update

Let C^i be the operator that takes as input an image pair $(\mathcal{I}_0(i), \mathcal{I}_1(i))$ and produces an estimate $\tilde{\chi}(i)$ of the set of correspondence graphs for each pair of conjugate epipolar lines. Without any *a priori* knowledge besides the boundaries of segmented and matched occluding regions[1], a correspondence algorithm would need to solve a set of monotonic matching problems over a series of rectangular domains (see Figure 5).

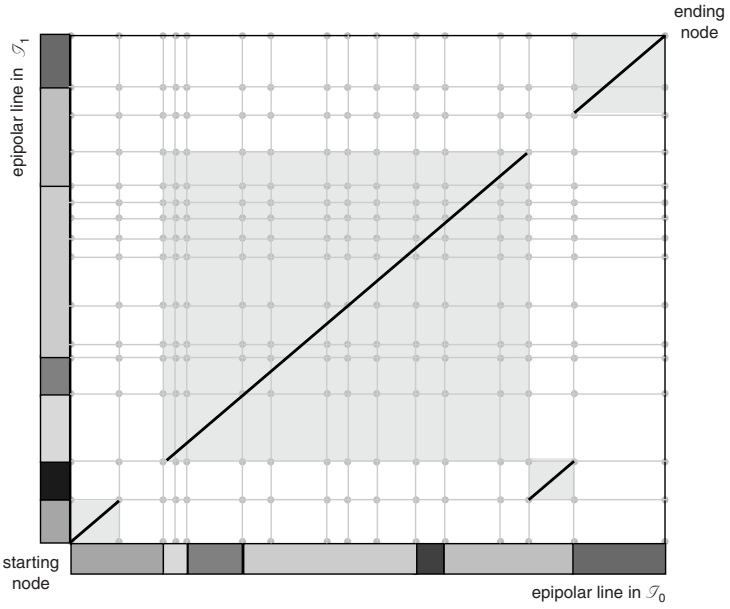

Figure 5: The set of rectangular domains searched by the correspondence operator C^i given basic correspondence graph topology for one epipolar line pair.

However, at times $i > 0$, we possess the set of time-updated correspondence graphs from time $i-1$, which we assume to be a good estimate of the set of correspondence graphs at time i. Hence, given an estimate of correspondence $\hat{\chi}(i+1 \mid i)$, we define the measurement update operator $M^{i+1}(\hat{\chi}(i+1 \mid i))$ to be C^{i+1} restricted to an ε-ball around $\hat{\chi}(i+1 \mid i)$. This is illustrated in Figure 6 for one epipolar line pair.

[1]Obtaining these boundaries is a well-studied and difficult computer vision problem that needs to be solved separately, and is not the focus of our paper. In future work we would like to incorporate automated methods for the segmentation problem in-line with our virtual video algorithm.

To be more concrete, consider the correspondence graph X that is one element of $\hat{\chi}(i+1 \mid i)$ for a given conjugate epipolar line pair. The endpoints of X delimit a set D of rectangular domains that constitute the search neighborhood for the correspondence algorithm C^{i+1}. The measurement update operator searches the subset B of D given by

$$B = D \cap X^{(\varepsilon)} \tag{3}$$

Here $X^{(\varepsilon)}$ is the ε-ball around X defined by a metric d on $\mathbb{R}^2$ and the dilation operator $X^{(\varepsilon)} = \{x \in \mathbb{R}^2 | \inf_{y \in X} d(x,y) < \varepsilon\}$. We intersect the ε-ball with D so that the output of the measurement update operator $M^{i+1}(\hat{\chi}(i+1 \mid i))$ is still a set of monotonic line segments with the same endpoints as $\hat{\chi}(i+1 \mid i)$.

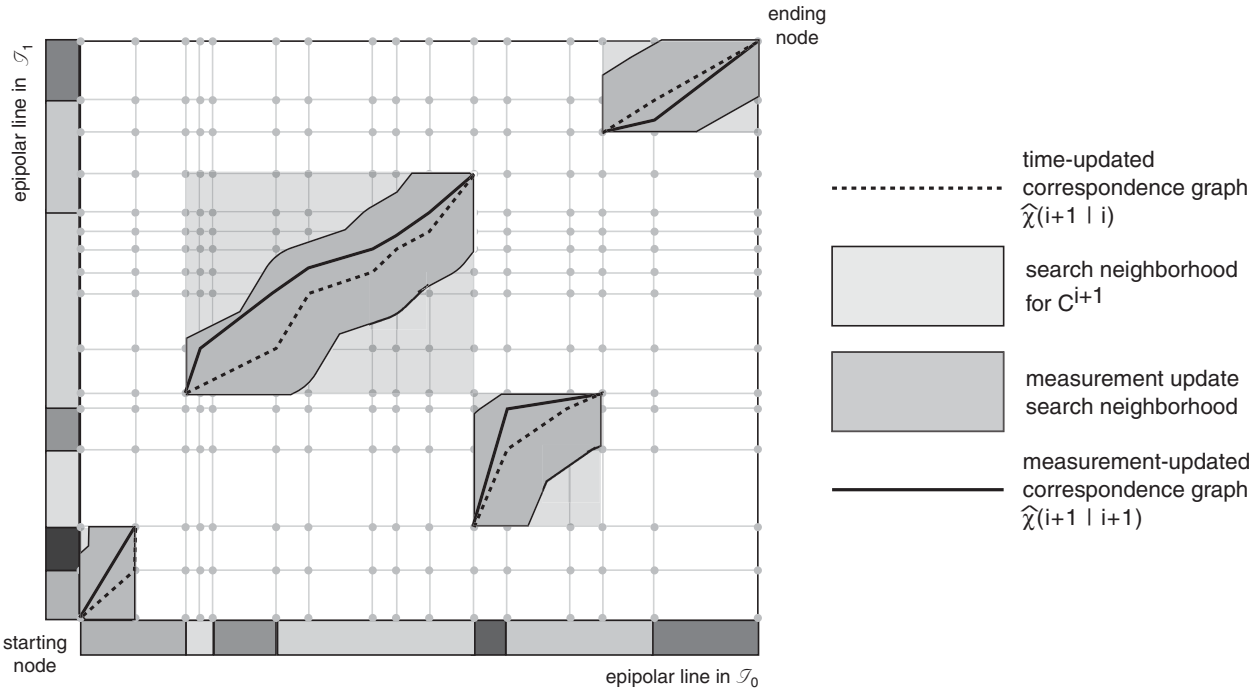

Figure 6: Measurement update by searching a local neighborhood around the time-updated estimate.

By construction, $B \subset D$, and if ε is small the area of B can be substantially smaller than the area of D. Specifically, if D is the union of K rectangles with dimensions $M_k \times N_k$, $k = 1, \ldots, K$,

then the ratio r of the area of B to the area of D is approximately $r = \sum_{k=1}^{K} \frac{2\varepsilon(M_k+N_k)-\varepsilon^2}{M_k N_k}$ if the ε-ball is based on the L_1 norm, and $r = \sum_{k=1}^{K} \frac{2\varepsilon\sqrt{M_k^2+N_k^2}-\varepsilon^2\frac{M_k^2+N_k^2}{M_k N_k}}{M_k N_k}$ if the ε-ball is based on the L_2 norm. In either case, if $\varepsilon \ll M_k, N_k$, r becomes quite small. The measurement update M^{i+1} can be computed more efficiently than the full correspondence operator C^{i+1}, since the computation required to solve the correspondence estimation problem over a domain is proportional to the area of that domain. In our experiments we used the Ohta and Kanade algorithm (see Appendix A and [17]) as out operator C^{i+1}.

4.3 Error Analysis

We use the recurrence $\hat{\chi}(i+1 \mid i+1) = M^{i+1}\hat{T}^{i+1}(\hat{\chi}(i \mid i))$, where $\hat{T}^{i+1}$ is an estimate of the true T^{i+1} induced by camera dynamics. We are interested in bounding the difference between the output of the $(\hat{T}, M)$ algorithm and the true correspondence $\chi^*(i+1)$. To this end, we consider the estimation error $\varepsilon_{TM}(i)$ at each time i, defined in the following way.

For a given point $w_0 \in \mathcal{P}_0(i)$, let $w_1^{*i}(w_0)$ be the true correspondence in $\mathcal{P}_1(i)$, if it exists (i.e. is unoccluded), and let $\hat{w}_1^i(w_0)$ be the estimate of $w_1^{*i}(w_0)$ given by the $(\hat{T}, M)$ algorithm at time i, if it exists (i.e. is unoccluded.) Then let $W_0(i) \subset \mathcal{P}_0(i)$ be the set of all $w_0 \in \mathcal{P}_0(i)$ for which both $w_1^{*i}(w_0)$ and $\hat{w}_1^i(w_0)$ are defined. Similarly, we define the quantities $w_0^{*i}(w_1)$ and $\hat{w}_0^i(w_1)$ associated with a point $w_1 \in \mathcal{P}_1(i)$ and the related subset $W_1(i) \subset \mathcal{P}_1(i)$. Then we define the estimation error $\varepsilon_{TM}(i)$ as:

$$\varepsilon_{TM}(i) = \max\left\{ \sup_{w_0 \in W_0(i)} \|\hat{w}_1^i(w_0) - w_1^{*i}(w_0)\|, \sup_{w_1 \in W_1(i)} \|\hat{w}_0^i(w_1) - w_0^{*i}(w_1)\| \right\} \tag{4}$$

$\varepsilon_{TM}(i)$ is simply the maximal difference (for visible points where an estimate exists) between where a correspondence truly is at time i and where it was estimated to be. We can describe the accumulation of error between time i and time $i+1$ in terms of several factors. In the following, we fix an arbitrary scene point P, and let $w_j^*(i)$ be the projection of P onto $\mathcal{P}_j(i)$. We let $\hat{w}_j(i)$ be

the estimate of $w_j^*(i)$ given by the $(\hat{T}, M)$ algorithm.

1. Errors in the epipolar geometry estimate due to errors in the projective transformation estimates. As mentioned in Section 4.1, in practice we approximate the projective transformations $(P(i), Q(i))$ that relate temporally adjacent frames by estimates $(\hat{P}(i), \hat{Q}(i))$. We assume that the accuracy of each projective transformation estimate is bounded in the sense that:

$$\begin{aligned} \|\hat{P}(i)w_0^*(i-1) - P(i)w_0^*(i-1)\| &\leq \gamma \\ \|\hat{Q}(i)w_1^*(i-1) - Q(i)w_1^*(i-1)\| &\leq \gamma \end{aligned}$$

   A finite γ exists because of the finite extent of the image planes. γ is also a function of the underlying rotation and zoom of the cameras and the estimation algorithm that is used. For an accurate algorithm, we expect γ to be less than a few pixel widths.

2. The magnitude of the scene dynamics between adjacent frames. We assume that δ is the maximum distance that the projections of scene points can move after compensating for camera motion. That is, we assume

$$\|w_0^*(i) - P(i)w_0^*(i-1)\| \leq \delta \quad (5)$$

$$\|w_1^*(i) - Q(i)w_1^*(i-1)\| \leq \delta \quad (6)$$

3. Errors in the correspondence along estimated conjugate epipolar lines in the measurement update. Even when the epipolar geometry estimate is very accurate, changes in illumination, occlusions, and deviation from modeling assumptions can prevent the measurement update operator from accurately estimating the correspondence along a conjugate epipolar line pair. In the case of inaccurate epipolar geometry, the lines along which the measurement update

associates points do not match in physical reality, and the estimated correspondence will be noticeably incorrect in non-smooth areas of the images.

Inaccuracies in matching along an estimated pair of conjugate epipolar lines are bounded by the radius ε of the measurement update operator, discussed in Section 4.2. That is,

$$\begin{aligned} \|\hat{w}_0(i) - \hat{P}(i)\hat{w}_0(i-1)\| &\leq \varepsilon \\ \|\hat{w}_1(i) - \hat{Q}(i)\hat{w}_1(i-1)\| &\leq \varepsilon \end{aligned}$$

In practice, we should choose $\varepsilon \approx \delta$.

4. Relative contraction or expansion induced by the projective transformations. At each iteration i, we assume that

$$\begin{aligned} \|P(i)w_0^*(i-1) - P(i)\hat{w}_0(i-1)\| &\leq \alpha\|w_0^*(i-1) - \hat{w}_0(i-1)\| \\ &= \alpha\varepsilon_{TM}(i-1) \\ \|Q(i)w_1^*(i-1) - Q(i)\hat{w}_1(i-1)\| &\leq \alpha\|w_1^*(i-1) - \hat{w}_1(i-1)\| \\ &= \alpha\varepsilon_{TM}(i-1) \end{aligned}$$

The constant α is a function of the underlying camera rotation and zoom represented by the transformations $P(i)$ and $Q(i)$. For closely spaced images, $\alpha \approx 1$.

Figure 7 illustrates the error at time 1 in frame $\mathcal{P}_1(1)$, which can be seen to satisfy $\varepsilon_{TM}(1) \leq \delta + \gamma + \varepsilon$. Inductively, we can show

$$\varepsilon_{TM}(i) \leq (\delta + \gamma + \varepsilon)\sum_{n=0}^{i-1}\alpha^n \tag{7}$$

$$= (\delta + \gamma + \varepsilon)\frac{1-\alpha^i}{1-\alpha} \tag{8}$$

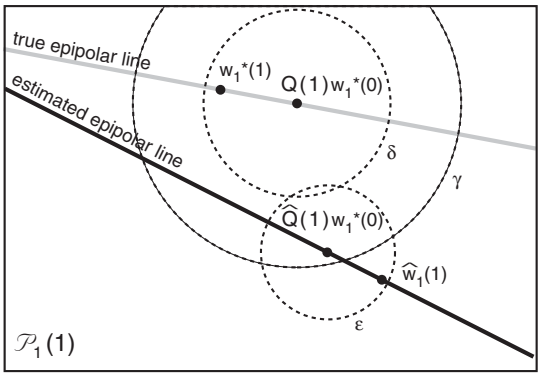

Figure 7: Sources of error at time 1.

Hence, we can guarantee the error remains bounded by a given number of pixel widths p when the time index satisfies

$$i < \frac{\log\left(1 + \frac{(1-\alpha)p}{\delta+\gamma+\varepsilon}\right)}{\log \alpha} \tag{9}$$

After this point, to keep the error within p pixel widths, a reinitialization of the epipolar geometry would be required to re-establish the accuracy of the correspondence estimates. We briefly address the problem of periodically re-estimating the epipolar geometry in Section 7.

5 Virtual View Synthesis Given Correspondence

To illustrate how a certain class of virtual images can be synthesized from a dense correspondence estimate, we briefly review the view morphing algorithm [5] for a pair of still images. There are many other approaches to image-based view synthesis in the computer graphics literature, including plenoptic modeling [2], the light field [3] and the lumigraph [4]. Other researchers (e.g. Laveau and Faugeras [18] and Avidan and Shashua [19]) have discussed using images from more than two cameras to create virtual still images. A recent paper by Ma et. al [20] characterizes the set of physically correct virtual images that can be obtained from a finite number of real images.

Consider the camera configuration of Figure 8a, in which the two image planes $\mathcal{P}_0$ and $\mathcal{P}_1$ are parallel to each other and to the baseline (i.e. they are rectified). Let f_0 and f_1 be the focal lengths

of $\mathcal{C}_0$ and $\mathcal{C}_1$, respectively. Without loss of generality, we can fix the origins of the cameras to be $O_0 = (0,0,0)$ and $O_1 = (1,0,0)$. Fix a third camera $\mathcal{C}_s$ with origin $O_s = \left(\frac{sf_1}{(1-s)f_0+sf_1}, 0, 0\right)$, focal length $f_s = (1-s)f_0 + sf_1$, and image plane $\mathcal{P}_s$ parallel to $\mathcal{P}_0$ and $\mathcal{P}_1$. If we fix a scene point P and consider its projections w_0,w_1, and w_s onto $\mathcal{P}_0$, $\mathcal{P}_1$, and $\mathcal{P}_s$ respectively, then Chen and Williams [1] noted that

$$w_s = (1-s)w_0 + sw_1 \tag{10}$$

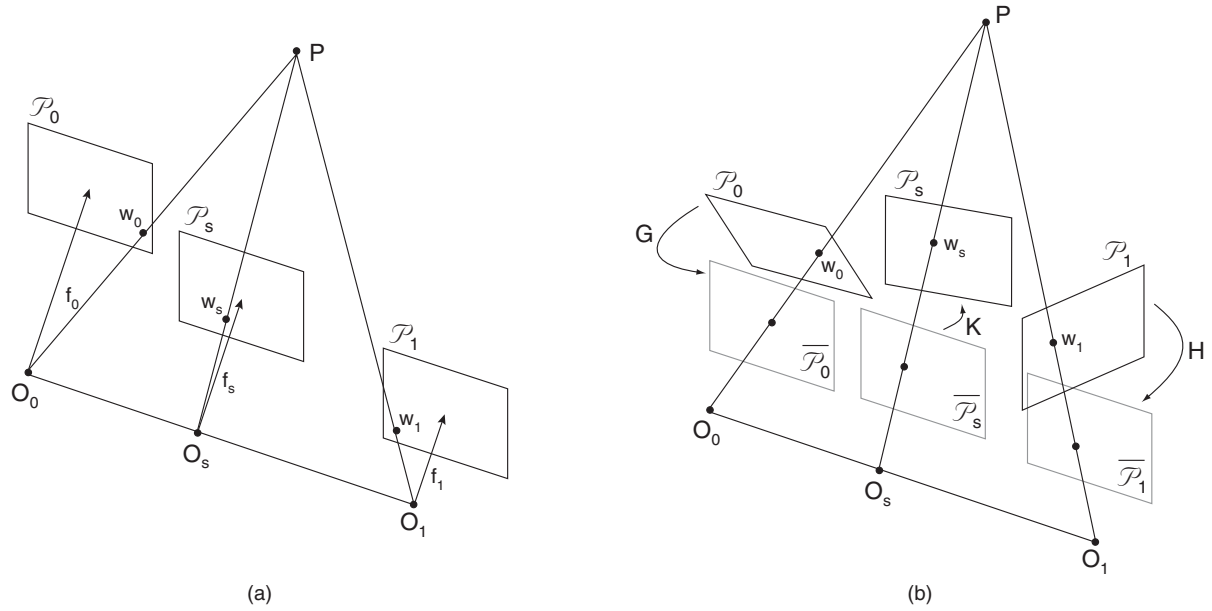

Figure 8: (a) View interpolation. (b) View morphing.

That is, interpolating the image coordinates of the projections of P gives the same result as projecting P onto the image plane of the "interpolated" camera $\mathcal{C}_s$. Hence, a new projection of the scene onto $\mathcal{P}_s$ can be obtained without knowledge of the three-dimensional locations of cameras or scene points. Provided that given any point $w_0 \in \mathcal{P}_0$, its correspondence $w_1 \in \mathcal{P}_1$ can be estimated, the correspondence $w_s \in \mathcal{P}_s$ can be computed through (10). Chen and Williams called this result view interpolation.

In the more general camera configuration of Figure 8b, the orientations of the image planes and focal lengths of the cameras are unconstrained. However, the image planes can be rectified by

an appropriate pair of projective transformations (G, H). Since $\bar{\mathcal{P}}_0$ and $\bar{\mathcal{P}}_1$ are in the configuration necessary for view interpolation, a new projection can be synthesized from the perspective of a camera $\bar{\mathcal{C}}_s$ whose origin O_s lies at $(s, 0, 0)$ and whose image plane $\bar{\mathcal{P}}_s$ is coplanar with $\bar{\mathcal{P}}_0$ and $\bar{\mathcal{P}}_1$. The image plane $\mathcal{P}_s$ of an arbitrary camera $\mathcal{C}_s$ with origin O_s can be obtained from $\bar{\mathcal{P}}_s$ by application of an appropriate projective transformation K that effectively rotates the image plane from $\bar{\mathcal{P}}_s$ to $\mathcal{P}_s$. If we fix a scene point P and consider its projections w_0,w_1, and w_s onto $\mathcal{P}_0$, $\mathcal{P}_1$, and $\mathcal{P}_s$ respectively, we have the central equation

$$w_s = K^{-1}((1-s)G(w_0) + sH(w_1)) \tag{11}$$

This result was first obtained by Seitz and Dyer [5], who called the process view morphing. Observe that the camera $\mathcal{C}_s$ is described relative to $\mathcal{C}_0$ and $\mathcal{C}_1$, not in reference to absolute 3-D coordinates.

The view morphing equation (11) is a statement only about the positions of corresponding points in the image planes, not about their colors. Here we proceed from the Lambertian assumption that scene points have the same color regardless of the viewing angle, and that the color of an image point is the same as the color of a single corresponding scene point. To compensate for deviations from these assumptions in real images, we will color points in the virtual images by a weighted average:

$$\mathcal{I}_s(w_s) = (1-s)\mathcal{I}_0(w_0) + s\mathcal{I}_1(w_1) \tag{12}$$

We illustrate an example of view morphing using the two images in Figures 9a and 9b. These natural, outdoor images come from widely separated cameras viewing a soccer game. A dense correspondence of all points which appear in both images was estimated using the correspondence graph formalism described above, and the view morphing equations (11)-(12) were used to create the virtual image in Figure 9c using $s = 0.5$. We used the algorithm suggested in Seitz's original paper [5] to obtain rectifying transformations (G, H), and let $K = I$ in (11). While the rendered

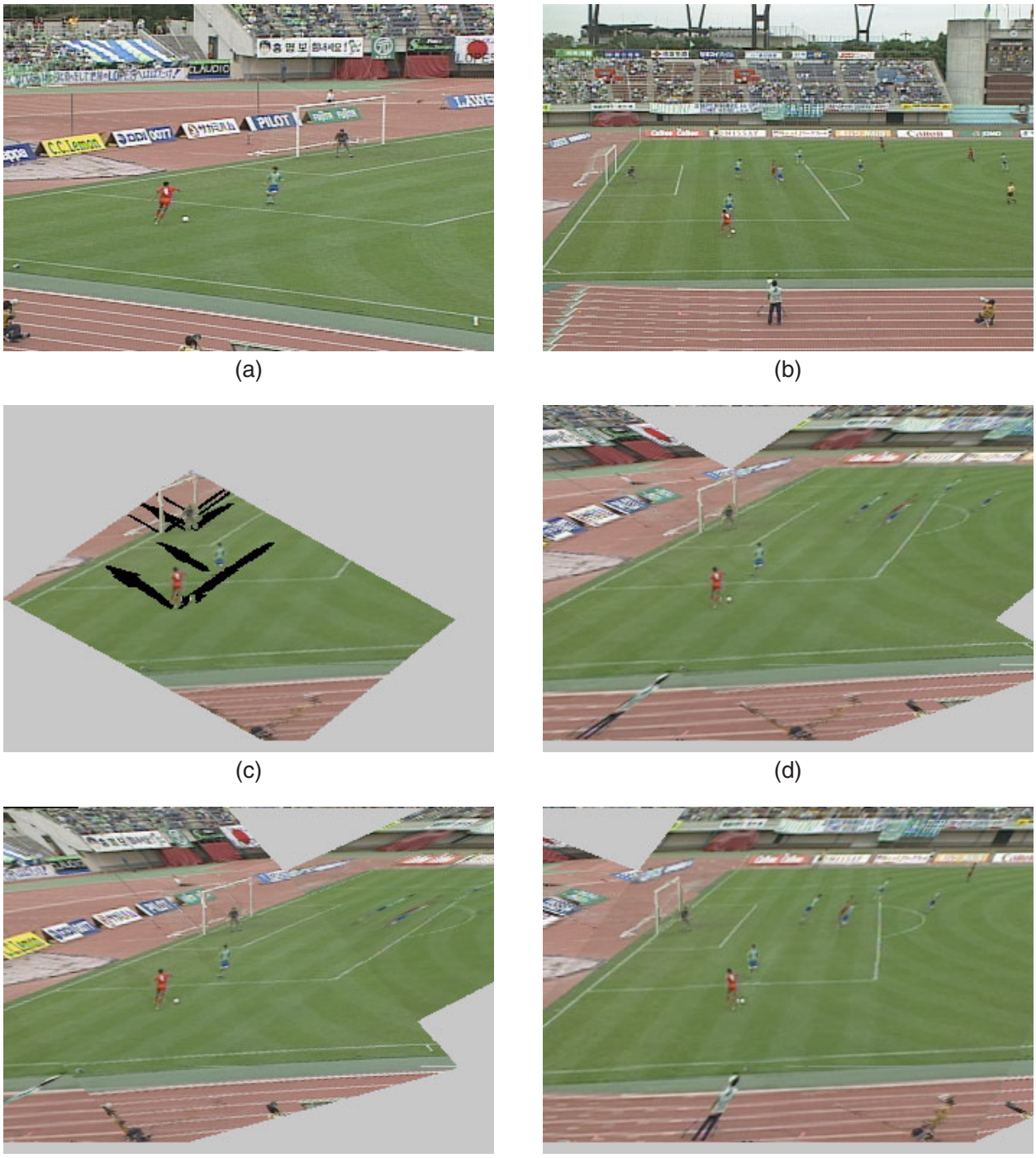

Figure 9: (a) and (b) Original images, corresponding to $s = 0$ and $s = 1$. (c) and (d) Synthesized virtual image $\mathcal{I}_s$ at $s = 0.5$, with and without filling of occluded regions. (e) and (f) Interpolated virtual images $\mathcal{I}_s$ at $s = 0.25$ and $s = 0.75$, respectively.

pixels appear realistic, the eye is drawn to the limited extent of the virtual image compared to the originals, and the black regions in the virtual image plane that correspond to pixels visible in only one of the images $(\mathcal{I}_0, \mathcal{I}_1)$.

In this example, we can alleviate both of the above problems by supposing that the scene points visible in only one of the images lie on a planar surface. Consider a scene point P that is visible in $\mathcal{I}_0$ at w_0 but is not visible in $\mathcal{I}_1$. We compute an estimate $\tilde{w}_1$ that is the image of w_0 under the projective transformation induced by the planar surface [21]. Then $(w_0, \tilde{w}_1)$ can be treated as a correspondence, and the projection w_s of P in $\mathcal{P}_s$ can be estimated as $w_s = (1-s)G(w_0)+sH(\tilde{w}_1)$. However, in this case we should only use the color of the point in the image where it is visible- that is, $\mathcal{I}_s(w_s) = \mathcal{I}_0(w_0)$. We take a similar tactic for points that are visible in $\mathcal{I}_1$ but not in $\mathcal{I}_0$. Of course, there may be regions that are visible in neither image due to occlusions by multiple objects. A correspondence estimate $(\tilde{w}_0, \tilde{w}_1)$ can be obtained for such a point from the planar assumption, but there is no color information for this point. In this case, we can interpolate the colors from either side of the missing piece, or use a default color.

The result of filling in occluded regions by the planar assumption is illustrated in Figure 9d. Since the planar assumption is valid over many occluded pixels, the virtual image is much more realistic. Distortion is visible in several regions where the planar assumption is invalid, such as the stands in the upper left corner, and the soccer players at the upper right. However, the virtual image is a convincing rendition of the scene from a viewpoint that is halfway between the unknown optical centers of the original cameras. Interpolated views with $s = 0.25$ and $s = 0.75$ are illustrated in Figures 9e and 9f, respectively. Note that we can see arrangements of objects in the virtual images (e.g. the position of the goalie with respect to the goalposts in Figure 9d) that never occurred in the original frames. In the case of more than two cameras, virtual views can be constructed whose camera centers lie in the convex hull of the centers of the original cameras.

6 Experimental Results

Here we demonstrate the results of the recursive propagation framework in the context of creating virtual video. Our test sequence is 43 frames long and constitutes a single event from a soccer game (a player attempts to kick the ball and is tripped). The 24-bit color frames are 340 $\times$ 240 pixels, and come from a high-quality digital video camera.

Our current implementation produces virtual video at about 20 frames per minute. User intervention is required to provide a sparse set of point correspondences in the initial frame pair (used to estimate the epipolar geometry and the projective transformation relating the planar surface in the image pair), and segmentation and tracking information for moving objects in each frame (used to construct correct correspondence graphs). In this example, this information was obtained by hand. Again, in future work we hope to incorporate a segmentation and tracking algorithm in-line with the video synthesis algorithm, but this is not our focus here. A fully automatic system could use active contours [22, 23] or conditional density estimation [24] for tracking.

The projective transformations $P(i)$ and $Q(i)$ were estimated using the efficient algorithm described in [25], using point matches extracted by the automatic feature selection algorithm described in [26]. The measurement update used an 8-pixel search neighborhood about the time-updated estimate and the Ohta-Kanade cost function.

Figure 10 illustrates the results of the algorithm on conjugate epipolar line 105 for the first and second frames of video (labeled Frame 0 and Frame 1). The upper left hand corner of each figure is the basic correspondence graph for Frame 0 induced by the planar assumption and object segmentation. The upper right hand corner is the refined correspondence graph for Frame 0 obtained by applying the measurement update operator to the basic correspondence graph. The lower left hand corner is the correspondence graph for Frame 1 obtained by the time update, and the lower right hand corner is the correspondence graph for Frame 1 obtained by the measurement update.

The correspondence graphs all seem rather similar (which is the point of the algorithm). How-

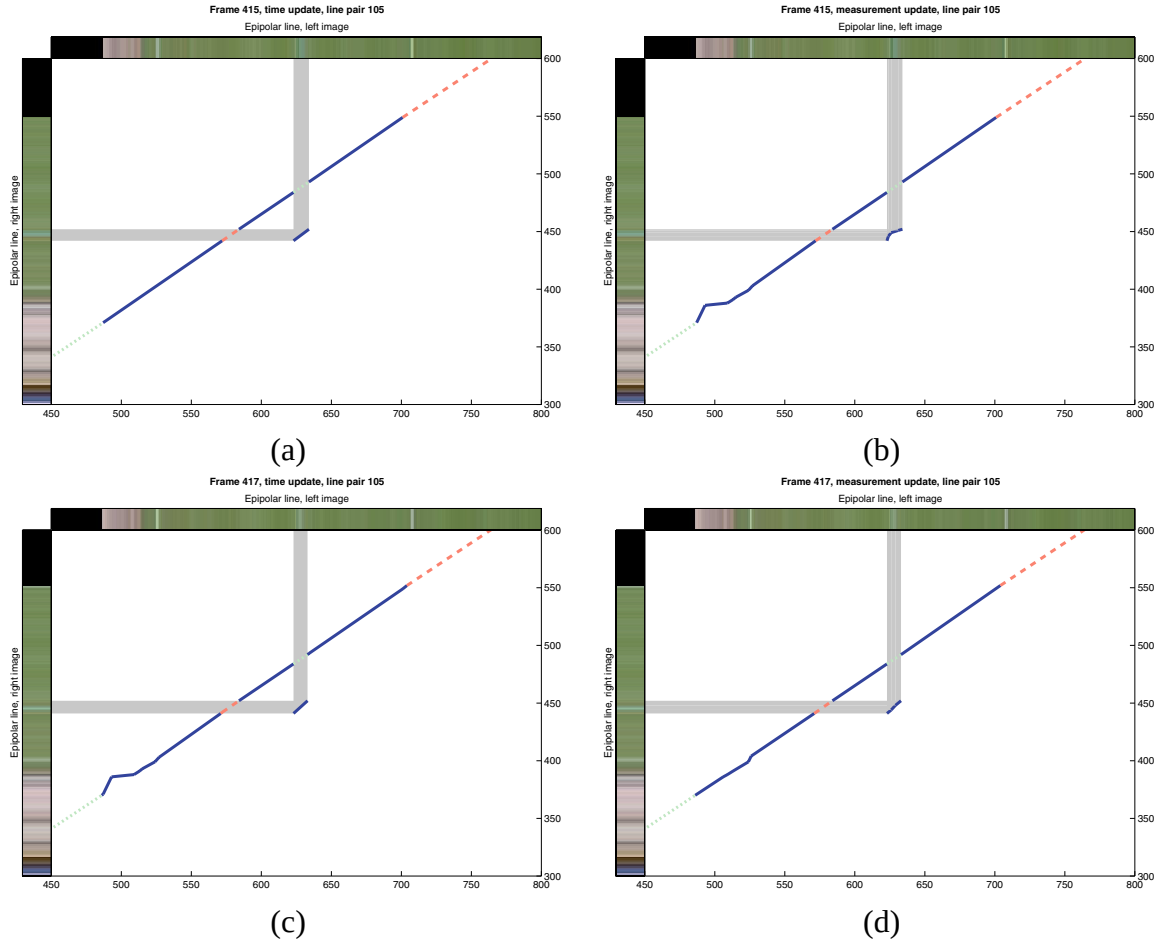

Figure 10: Correspondence graphs, line 105. (a) Frame 0 initialization. (b) Frame 0 measurement update. (c) Frame 1 time update. (d) Frame 1 measurement update.

ever, it can be seen clearly from Figure 10b and 10c that the background correspondence from Frame 0 is time-updated to the same location in Frame 1 (note the "elbow" at the lower left end of the long piece). This correspondence is refined by the measurement update (and the elbow disappears in Figure 10d.)

More compelling are the virtual video frames rendered using this correspondence. Six such frames are illustrated in the middle column of Figure 11. The left and right columns are real images $\{\mathcal{I}_0(i), \mathcal{I}_1(i)\}$ seen at various times i, corresponding to locations along the baseline of $s = 0$ and $s = 1$. The middle column is a rendition from a moving virtual camera whose optical center moves at constant speed from $s = 0$ to $s = 1$. Over the course of the video shot, camera $\mathcal{C}_0$ undergoes a slow pan to the right, while camera $\mathcal{C}_1$ slowly zooms in. The virtual camera has dynamics observed in neither of the source video clips, and moves very quickly and smoothly, at approximately 15 m/s. This is an example of a virtual camera being used in a situation where physical limitations preclude the use of a conventional physical camera.

Unfortunately, it is difficult to convey the three-dimensional feeling of the rendered video from these still images. We refer the reader to `http://www.ecse.rpi.edu/homepages/rjradke/pages/vvid/vvid.html` for several virtual video examples using the images in this paper.

In later frames of the video, the error growth of the algorithm results in minor but visible artifacts. Notably, some of the soccer players seem to "lose their heads"- the head of the player appears several pixels away from the correct location on top of the body. This is especially visible at Frame 34. This is largely due to the accumulation of errors in the estimation of the projective transformations $P(i)$ and $Q(i)$, which in turn affect the accuracy of the estimated epipolar geometry. At this point, correspondence is not being estimated along true epipolar lines. Though our projective transformation estimation algorithm is generally quite accurate, after i iterations, the projective transformations $\hat{G}(i)$ and $\hat{H}(i)$ applied to $\mathcal{P}_0(i)$ and $\mathcal{P}_1(i)$ are compositions of i estimated transformations. In this video sequence, when i is more than about 25, $(\hat{G}(i), \hat{H}(i))$ are no

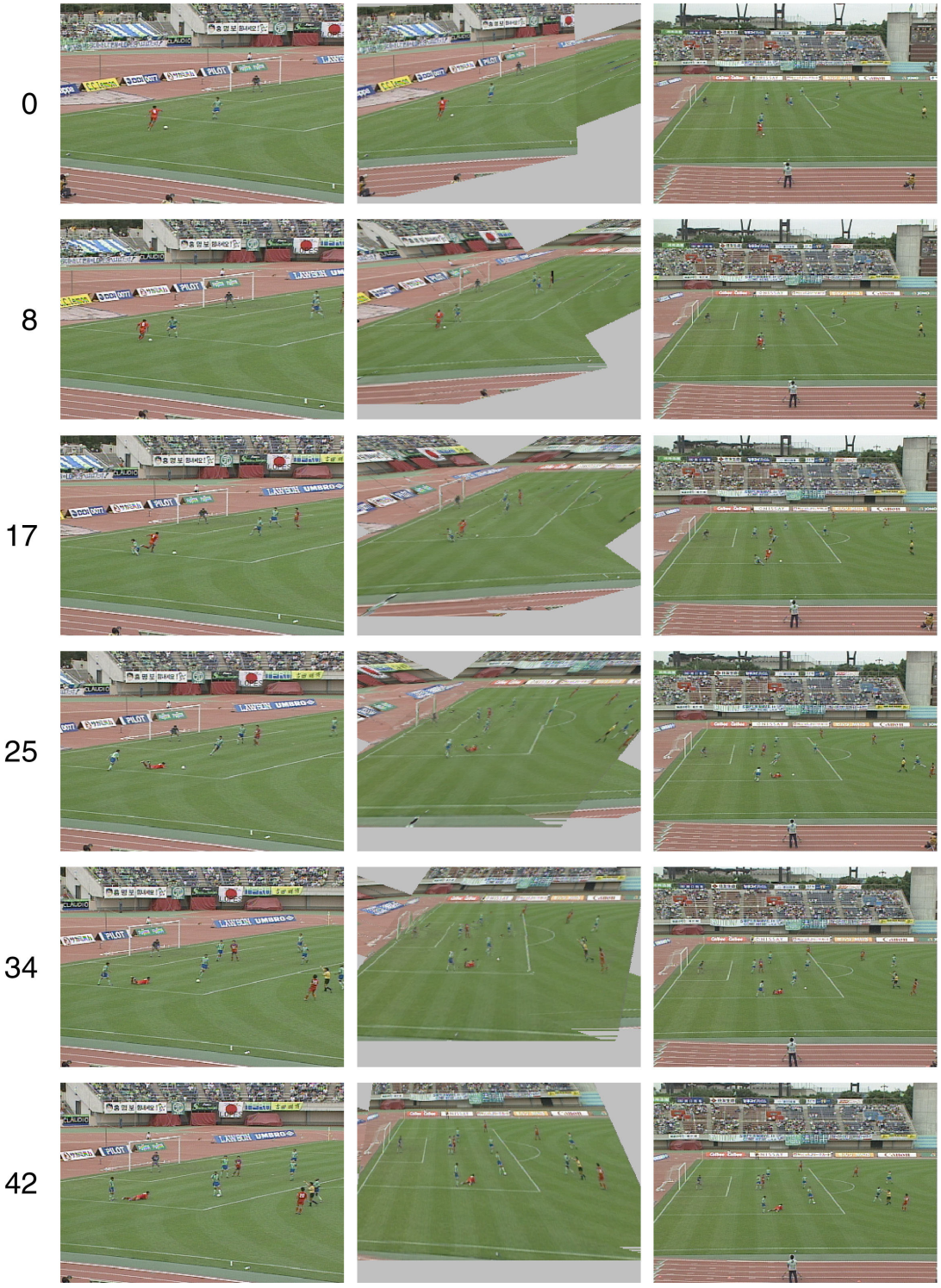

Figure 11: Virtual images, frames 0, 8, 17, 25, 34, 42. Left column: original $\mathcal{C}_0$ frame,$s = 0$. Right column: original $\mathcal{C}_1$ frame, $s = 1$. Middle column: virtual $\mathcal{C}_s$ frame, $s = 0, 0.2, 0.4, 0.6, 0.8, 1.0$

longer close to a rectifying pair. This problem should be alleviated by a periodic re-estimation of the epipolar geometry, as discussed below.

7 Conclusions

Each of the virtual images in our experiment is a convincing rendition of the dynamic scene from an intermediate viewpoint. We emphasize that the effects exhibited here are similar to those produced by specialized multicamera hardware. However, here we only require two uncalibrated cameras and no 3-D scene modeling. These results show that understanding the relationship between image correspondence and camera motion can be a powerful tool.

There are many directions for future work in the area of virtual video, both in improving the stability of the estimation algorithm and in rendering the synthetic images. As addressed in the text, the propagation process eventually destabilizes, due to accumulation of errors in the estimation of the projective transformations. A reinitialization of the epipolar geometry is required. However, since this estimation requires the selection and matching of feature points between images with a substantial perspective difference, user intervention is generally required to obtain reliable results. Since some matching points are selected by the user for the first frame pair, one approach is to track these points through each image sequence, using a measure of feature similarity that is invariant to perspective distortion (e.g. based on corners). Periodically, the algorithm could be restarted with a new estimate of the fundamental matrix and rectifying projective transformations obtained from these tracked points. Automatically detecting that restarting is necessary and maintaining continuity of the rectifying transformations and virtual images across the restarted frame would be problems to overcome. Alternately, we are exploring a recursive algorithm for quickly, incrementally improving the epipolar geometry estimate at each frame using a small amount of new information.

As noted, while not the central point of this paper, segmentation and tracking of objects in the

video sequences is an important and difficult issue that we hope to address in the future. Integrating robust, automated methods for these tasks and assessing their performance is an interesting direction to pursue for further work, and would be crucial in order to put a practical virtual video system in place.

In terms of rendering quality, the virtual images are slightly blurry compared to the original video frames. This is caused by several steps of image resampling in our current implementation and could be alleviated by removing the dependence of our rendering algorithm on explicitly rectified images. Additionally, post-processing techniques (e.g. unsharp masking, texture-mapping of surfaces) could be used to improve the perceptual quality of the virtual video.

References

[1] S.E. Chen and L. Williams. View Interpolation for Image Synthesis. *Computer Graphics (SIGGRAPH '93)*, pp. 279–288, July 1993.

[2] L. McMillan and G. Bishop. Plenoptic Modeling: An Image-Based Rendering System. *Computer Graphics (SIGGRAPH '95)*, pp. 39–46, August 1995.

[3] M. Levoy and P. Hanrahan. Light Field Rendering. *Computer Graphics (SIGGRAPH '96)*, pp. 31–42, August 1996.

[4] S.J. Gortler, R. Grzeszczuk, R. Szeliski, and M.F. Cohen. The Lumigraph. *Computer Graphics (SIGGRAPH '96)*, pp. 43–54, August 1996.

[5] S.M. Seitz and C.R. Dyer. View Morphing. *Computer Graphics (SIGGRAPH '96)*, pp. 21–30, August 1996.

[6] *The Matrix*, Warner Brothers, 1999.

[7] *Superbowl XXXV*, CBS Sports, January 28, 2001.

[8] R. Radke, P. Ramadge, S. Kulkarni, T. Echigo, and S. Iisaku. Recursive Propagation of Correspondences with Applications to the Creation of Virtual Video. In *Proc. ICIP 2000*, Vancouver, Canada, September 2000.

[9] O.D. Faugeras. *Three-Dimensional Computer Vision: A Geometric Viewpoint.* MIT Press, 1993.

[10] J.L. Barron, D.J. Fleet, and S.S. Beauchemin. Performance of Optical Flow Techniques. *International Journal of Computer Vision*, Vol. 12, No. 1, pp. 43–77, 1994.

[11] J.Y.A. Wang and E.H. Adelson. Representing Moving Objects with Layers. *IEEE Transactions on Image Processing Special Issue: Image Sequence Compression*, Vol. 3, No. 5, pp. 625–638, September 1994.

[12] Z. Zhang. Determining the Epipolar Geometry and its Uncertainty: A Review. *International Journal of Computer Vision*, vol. 27, no. 2, pp. 161-195, 1998.

[13] R. Radke, V. Zagorodnov, S. Kulkarni and P. Ramadge. Estimating Correspondence in Digital Video. In *Proc. ITCC 2001*, Las Vegas, Nevada, April 2001.

[14] R. Radke. *Estimation Problems in Digital Video.* Ph.D. Thesis, Department of Electrical Engineering, Princeton University, 2001.

[15] R.I. Hartley. Theory and Practice of Projective Rectification. *International Journal of Computer Vision*, Vol. 35, No. 2, pp. 115–127, November 1999.

[16] F. Isgrò and E. Trucco. Projective Rectification without Epipolar Geometry. In *Proc. CVPR '99*, June 1999.

[17] Y. Ohta and T. Kanade. Stereo by Intra- and Inter-Scanline Search Using Dynamic Programming. *IEEE PAMI*, Vol. 7, No. 2, pp. 139–154, March 1985.

[18] S. Laveau and O.D. Faugeras. 3-D Scene Representation as a Collection of Images and Fundamental Matrices. Technical Report 2205, INRIA-Sophia Antipolis, February 1994.

[19] S. Avidan and A. Shashua. Novel View Synthesis by Cascading Trilinear Tensors. *IEEE Transactions on Visualization and Computer Graphics*, vol. 4, no. 4, October-December 1998.

[20] Y. Ma, S. Soatto, J. Košecká, and S. Sastry. Euclidean Reconstruction and Reprojection up to Subgroups. *International Journal of Computer Vision*, Vol. 38, No. 3, pp. 219–229, 2000.

[21] R.Y. Tsai and T.S. Huang. Estimating the Three-Dimensional Motion Parameters of a Rigid Planar Patch. *IEEE Trans. ASSP*, vol. 25, no. 6, pp. 1147–1152, December 1981.

[22] C. Xu and J.L. Prince. Snakes, Shapes, and Gradient Vector Flow. *IEEE Transactions on Image Processing*, vol. 7, no. 3, pp. 359–369, March 1998.

[23] A. Tsai, A. Yezzi, Jr., and A.S. Willsky. Curve Evolution, Boundary-Value Stochastic Processes, the Mumford-Shah Problem, and Missing Data Applications. In *Proc. ICIP 2000*, Vancouver, Canada, September 2000.

[24] J. MacCormick and A. Blake. A Probabilistic Exclusion Principle for Tracking Multiple Objects. *International Journal of Computer Vision*, vol. 39, no. 1, pp. 57–71, 2000.

[25] R. Radke, P. Ramadge, T. Echigo, and S. Iisaku. Efficiently Estimating Projective Transformations. In *Proc. ICIP 2000*, September 2000.

[26] Y.P. Tan, S. Kulkarni, and P. Ramadge. Extracting Good Features for Motion Estimation. *Proc. ICIP 1996*, vol. 1, pp. 117–120, 1996.

[27] P.N. Belhumeur. A Bayesian Approach to Binocular Stereopsis. *International Journal of Computer Vision*, Vol. 19, No. 3, pp 237–260, 1996.

[28] I.J. Cox, S.L. Hingorani, and S.B. Rao. A Maximum Likelihood Stereo Algorithm. *Computer Vision and Image Understanding*, Vol. 63, No. 3, pp. 542–567, May 1996.

[29] H. Ishikawa and D. Geiger. Occlusions, Discontinuities, and Epipolar Lines in Stereo. In *Proc. ECCV '98*, Freiburg, Germany, 1998.

A Monotonic Correspondence Algorithms

A simple dynamic programming approach to estimating monotonic correspondence along conjugate epipolar lines was described by Ohta and Kanade [17]. The nodes of the program correspond to edges detected in each epipolar line. Intervals of nearly constant-intensity pixels are matched between conjugate epipolar lines, and points in a pair of matched intervals are put into correspondence by linearly interpolating between the endpoints. The function used to measure the cost of matching an interval $i_0 \in \ell_0$ with pixel intensities $\{a_1, \ldots, a_k\}$ to the interval $i_1 \in \ell_1$ with pixel intensities $\{b_1, \ldots, b_l\}$ is based on the variance of the intensities in the two intervals from a sample mean m, calculated as $m = \frac{1}{2}\left(\frac{1}{k}\sum_{i=1}^{k} a_i + \frac{1}{l}\sum_{j=1}^{l} b_j\right)$. The variance was computed as $\sigma^2 = \frac{1}{2}\left(\frac{1}{k}\sum_{i=1}^{k}(a_i - m)^2 + \frac{1}{l}\sum_{j=1}^{l}(b_j - m)^2\right)$. The cost of matching the two intervals was then defined as $C = \sigma^2\sqrt{k^2 + l^2}$. A slightly different cost was defined for intervals of pixels in one line that are occluded in the other line. Dynamic programming was used to find the lowest-cost path through each epipolar line matching graph. The authors also described a higher-dimensional matching problem over the entire image pair in which the nodes in the dynamic program are edges that cross many epipolar lines. This formulation explicitly enforces consistency between nearby epipolar lines.

Other, more sophisticated approaches to the epipolar-line-based correspondence problem exist, based on maximum *a posteriori* estimates [27], maximum likelihood estimates [28], and maximum-flow problems [29]. Each of these approaches also invokes the monotonicity assumption.

B The Correspondence Graph

Fix a pair of cameras $(\mathcal{C}_0, \mathcal{C}_1)$ whose centers of projection are O_0 and O_1, respectively. These cameras have associated image planes $\mathcal{P}_0$ and $\mathcal{P}_1$, that lie between the cameras' respective centers of projection and the scene $\mathcal{S}$, a collection of points in $\mathbb{R}^3$. Select a plane Φ containing the baseline, and view the intersection of Φ with the camera centers, the image planes, and the scene points as an imaging system with a 2-D scene $\mathbf{S} = \mathcal{S} \cap \Phi$ and 1-D image planes (the pair of conjugate epipolar lines (ℓ_0, ℓ_1)). We fix a coordinate system (x, y) on Φ by letting $O_0 = (0, 0)$ and $O_1 = (1, 0)$. Scene points are assumed to have positive y coordinates. The epipolar lines ℓ_0 and ℓ_1 inherit natural one-dimensional coordinate systems (denoted i and j respectively), oriented so that increasing i and j correspond to increasing x. In this setting, a correspondence is the realization of a point (x, y) in the scene as a pair $(i, j) \in \ell_0 \times \ell_1$. We will denote as $\mathbf{S}'$ the representation of the scene $\mathbf{S}$ in (i, j)-space.

Definition. The correspondence graph $C \subset \ell_0 \times \ell_1$ of a scene $\mathbf{S}$ with respect to the camera pair $(\mathcal{C}_0, \mathcal{C}_1)$ is the set of all points in $\mathbf{S}$ that are visible (i.e. unoccluded) in both ℓ_0 and ℓ_1, transformed into (i, j)-space.

The correspondence graph $C \subset \mathbf{S}'$. Generally $C \neq \mathbf{S}'$, since the correspondence graph takes occlusions into account and the transformed scene $\mathbf{S}'$ does not. However, the correspondence graph can be easily obtained from the set $\mathbf{S}'$. The construction is related to a certain morphological operation on points in (i, j)-space, described below.

Definition. A set A of points in (i, j)-space is a Southeast set if the subsets $\{(a, b) \in A \mid a = i\}$ and $\{(a, b) \in A \mid b = j\}$ have at most one element for all i, j.

Definition. The Southeasting operation $Se(\cdot)$ produces a Southeast set A' from a set A as follows:

$$A' = Se(A) = \{(i,j) \in A \mid \{(a,j) \in A \mid a > i\} \text{ and } \{(i,b) \in A \mid b < j\} \text{ are empty}\}$$

Proposition 1 *The correspondence graph C for a scene $\mathbf{S}$ with respect to $(\mathcal{C}_0, \mathcal{C}_1)$ can be generated by Southeasting the transformed scene $\mathbf{S}'$.*

Proof. We know that the correspondence graph C is a subset of the transformed scene $\mathbf{S}'$. It remains to determine which points in $\mathbf{S}'$ actually appear in both images. Fix i and consider the set of points $\mathbf{S}'_\mathbf{i} = \{(a,b) \in \mathbf{S}' \mid a = i\}$. These points lie on the same ray from $\mathcal{C}_0$ in (x,y)-space. The point p' with the smallest j coordinate is closest to $\mathcal{C}_0$ and is hence the only point along the ray that is imaged by $\mathcal{C}_0$. Therefore, the points in $\mathbf{S}'_\mathbf{i}$ with larger j coordinates than p' are not retained in the correspondence graph. Similarly, for fixed j, consider the set $\mathbf{S}'_\mathbf{j} = \{(a,b) \in \mathbf{S}' \mid b = j\}$. These points lie on the same ray from $\mathcal{C}_1$ in (x,y)-space, and the only point that is retained in the correspondence graph is that point q' with the largest i coordinate.

The operation described above is simply the Southeasting of the set $\mathbf{S}'$. By construction, the remaining elements in the Southeast set are precisely those points that appear in both cameras and hence this Southeast set is by definition the correspondence graph of $\mathbf{S}'$. ∎

Additionally, a partial converse to the above proposition is also true. That is, any Southeast set of points in (i,j)-space is the correspondence graph of some physical scene, provided that the graph lies within certain boundaries. Space precludes the inclusion of the converse here; see [14] for more details.

A scene $\mathbf{S}$ with a simple obstruction relative to two cameras is illustrated in Figure 12a. Figure 12b shows the scene transformed into (i,j)-space. The Southeasting process is applied in Figure 12c to obtain the correspondence graph in Figure 12d. The labeled line segments are projected to image plane $\mathcal{P}_0$ in the order 1-2-3-6-5, and to the image plane $\mathcal{P}_1$ in the order 1-6-3-4-5. Segments 3 and 6 appear in different orders in the projections; this reversal produces the phenomenon seen in the correspondence graph.

Belheumer [27] mentioned a "morphologically filtered version" of the disparity function between an epipolar line pair that is related to the correspondence graph. The filtering operation creates a continuous, monotonic path through the epipolar matching graph that includes regions that are "half-occluded", i.e. visible in one image only. However, this formalism only captures simple scenes that are constrained by monotonicity.

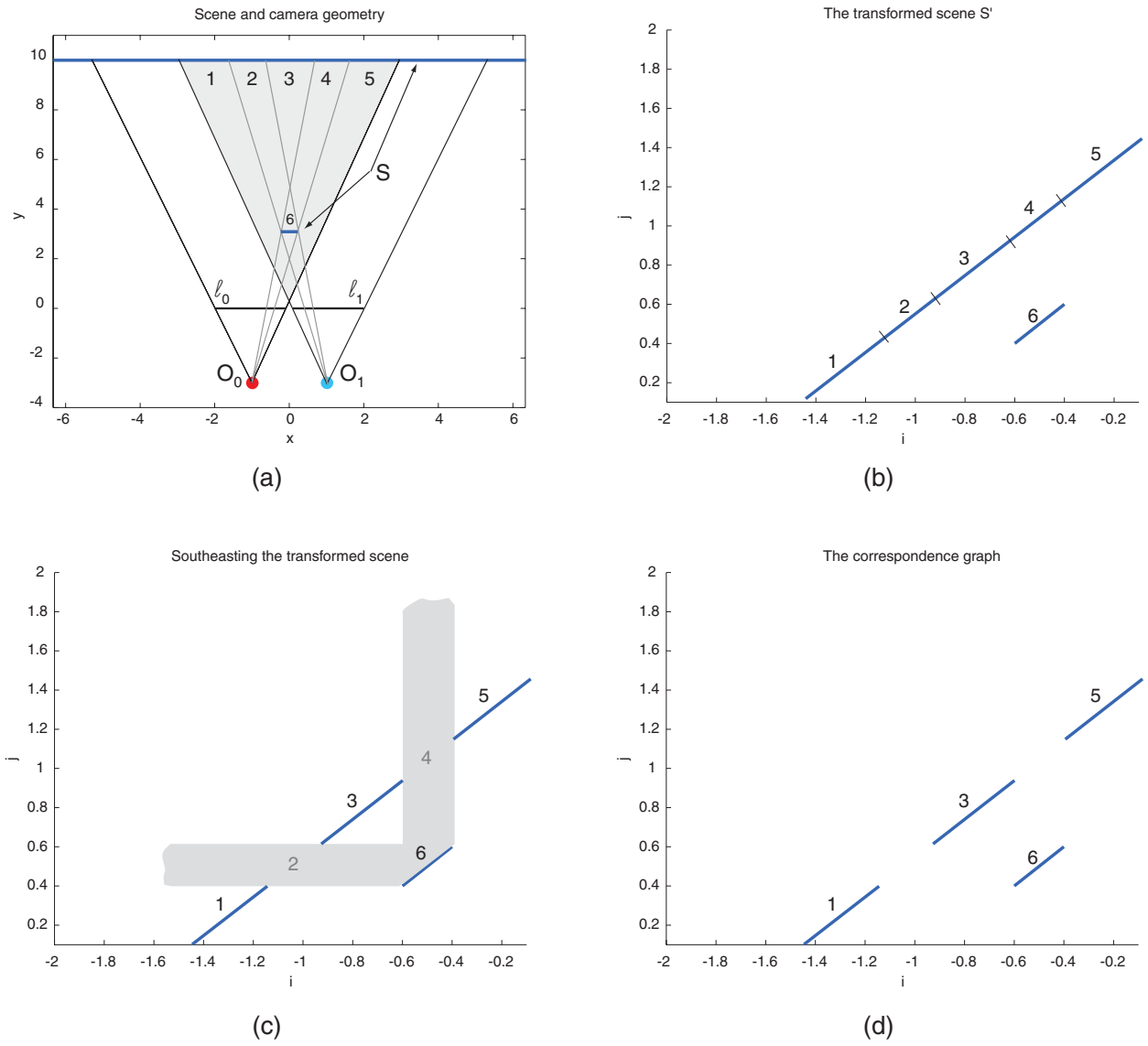

Figure 12: Example correspondence graph. (a) Scene $\mathbf{S}$ in (x, y)-space. (b) Transformed scene $\mathbf{S}'$ in (i, j)-space. (c) Southeasting the transformed scene. (d) Correspondence graph.