RECURSIVE PROPAGATION OF CORRESPONDENCES WITH APPLICATIONS TO THE CREATION OF VIRTUAL VIDEO

Richard Radke, Peter Ramadge,
Sanjeev Kulkarni

Department of Electrical Engineering
Princeton University
Princeton, NJ 08544

Tomio Echigo

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

Shun-ichi Iisaku

Communication Research Laboratory
Ministry of Posts and Telecommunications
4-2-1 Nukuikita-machi
Koganei-shi, Tokyo, Japan

1. INTRODUCTION

This paper is concerned with the efficient temporal propagation of correspondences between frames of two video sequences, an integral component of many video processing tasks. The main contribution of the paper is a framework for the recursive propagation of these correspondences.

The propagation consists of a time update step and a measurement update step. The time update depends only on the dynamics of the rotating source cameras, while the measurement update can be tailored to any member of a general class of image correspondence algorithms. Using these results, the correspondence between points of each frame pair can be propagated and updated in a fraction of the time required to estimate correspondences anew at every frame.

We discuss an application of the recursive correspondence propagation framework to the creation of virtual video. Previous virtual view algorithms have been used to generate synthetic video of a static scene, in which objects seem frozen in time. In contrast, the algorithms described here allow the creation of "true" virtual video, in the sense that the synthetic video evolves dynamically along with the scene.

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.

2. IMAGE DYNAMICS

We consider a pair of rotating cameras, $\mathcal{C}_0$ and $\mathcal{C}_1$, taking images of a dynamic scene. The image taken by $\mathcal{C}_j$ at time i is defined by $\mathcal{I}_j(i)$. This image lies on a coordinatized image plane $\mathcal{P}_j(i)$ (Figure 1).

We assume the cameras' centers of projection are not coincident, so the image planes $\mathcal{P}_0(i)$ and $\mathcal{P}_1(i)$ are related by a fundamental matrix $F(i)$ [1]. We also assume each camera's center of projection to be constant. Hence, the plane coordinates of $\mathcal{P}_j(i-1)$ and $\mathcal{P}_j(i)$ are related by a projective transformation [2], denoted by $P(i)$ and $Q(i)$ for $j = 0, 1$ respectively.

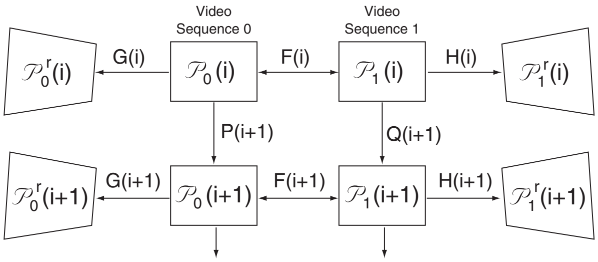


Fig. 1. Relationships between image planes.

To facilitate the detection and matching of corresponding regions between the image planes $\mathcal{P}_0(i)$ and $\mathcal{P}_1(i)$, it is common to rectify the planes by applying projective transformations $G(i)$ and $H(i)$ to produce image planes $\mathcal{P}_0^r(i)$ and $\mathcal{P}_1^r(i)$ in which conjugate epipolar lines are horizontal and have the same image plane y coordinate. As discussed in Hartley [3], the choice of a rectifying pair of projective transformations is not unique.

3. RECURSIVE PROPAGATION

Fix N scene points $S_k, k = 1, \ldots, N$, which are visible from the perspectives of both cameras. Let $x^*(i)$ denote the associated vector in $(\mathbb{R}^2 \times \mathbb{R}^2)^N$ defined by corresponding pairs of image points $\{(p_0^k(i), p_1^k(i)) \in \mathcal{P}_0(i) \times \mathcal{P}_1(i), k = 1, \ldots, N\}$, i.e., the points in the k^{th} pair are projections of S_k at time i. Let $\tilde{x}(i)$ be a vector of estimates of $x^*(i)$ 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.

We wish to more efficiently estimate $x^*(i)$ at each time. We do so by exploiting the temporal regularity of the video,

This research was partially supported by grants from the IBM Tokyo Research Laboratory and the New Jersey Center for Multimedia Research.

estimating the effect of camera motion and using a computationally simpler approximation of C^i. Namely, let $\hat{x}(i \mid j)$ be an approximation of $\tilde{x}(i)$ based on information from time j. $\hat{x}(i \mid j)$ satisfies:

$$\begin{aligned} \hat{x}(0 \mid 0) &= \tilde{x}(0) && (1) \\ \hat{x}(i+1 \mid i) &= T^{i+1}(\hat{x}(i \mid i)) && (2) \\ \hat{x}(i+1 \mid i+1) &= M^{i+1}(\hat{x}(i+1 \mid i)) && (3) \end{aligned}$$

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

To make this algorithm more concrete, we now discuss the operators T^i and M^i in more detail.

3.1. Time Update

Given complete knowledge of the camera motion of Figure 1, the time update for a correspondence $(p_0, p_1) \in \mathcal{P}_0(i) \times \mathcal{P}_1(i)$ is

$$T^{i+1}(p_0, p_1) = (P(i+1)p_0, Q(i+1)p_1) \qquad (4)$$

However, the projective transformations are generally estimated using a regression algorithm [4, 5], and in practice we use an approximation $\hat{T}^{i+1}$ of T^{i+1} given by

$$\hat{T}^{i+1}(p_0, p_1) = (\hat{P}(i+1)p_0, \hat{Q}(i+1)p_1) \qquad (5)$$

where $\hat{P}(i+1)$ and $\hat{Q}(i+1)$ are estimates of $P(i+1)$ and $Q(i+1)$ respectively. In Section 3.3 we will analyze the implications of this approximation.

In the case of rectified images, the image plane pairs $(\mathcal{P}_0^r(i), \mathcal{P}_0^r(i+1))$ and $(\mathcal{P}_1^r(i), \mathcal{P}_1^r(i+1))$ are related by projective transformations $R(i+1) = G(i+1)P(i+1)G(i)^{-1}$ and $S(i+1) = H(i+1)Q(i+1)H(i)^{-1}$, respectively. In this case, for $(p_0, p_1) \in \mathcal{P}_0^r(i) \times \mathcal{P}_1^r(i)$,

$$T^{i+1}(p_0, p_1) \;=\; (R(i+1)p_0, S(i+1)p_1) \qquad (6)$$

As above, the rectifying projective transformations are generally estimated from noisy data [3, 6, 7] and $R(i+1)$ and $S(i+1)$ are replaced in (6) by noisy estimates $\hat{R}(i+1) = \hat{G}(i+1)\hat{P}(i+1)\hat{G}(i)^{-1}$ and $\hat{S}(i+1) = \hat{H}(i+1)\hat{Q}(i+1)\hat{H}(i)^{-1}$ to produce an approximation $\hat{T}^{i+1}$.

In practice, we propagate the estimates $\hat{G}(i)$ and $\hat{H}(i)$ by the relations $\hat{G}(i+1) = \hat{G}(i)\hat{P}(i+1)^{-1}$ and $\hat{H}(i+1) = \hat{H}(i)\hat{Q}(i+1)^{-1}$. In this case the time update takes the particularly simple form

$$\hat{T}^{i+1}(p_0, p_1) = (p_0, p_1) \qquad (7)$$

3.2. Measurement Update

For clarity, we briefly present an example pair of families $\{C^i\}$ and $\{M^i\}$ when the desired correspondence lies along a pair of conjugate epipolar lines, a commonly made assumption in correspondence algorithms.

A classical approach to correspondence estimation is Ohta and Kanade [8], in which intervals of nearly constant-intensity pixels (obtained using an edge detector) are matched between a pair of conjugate epipolar lines (ℓ_0, ℓ_1), using the commonly made assumption that correspondences appear along the lines in the same order. Points in a pair of matched intervals are put into correspondence by linearly interpolating between the endpoints.

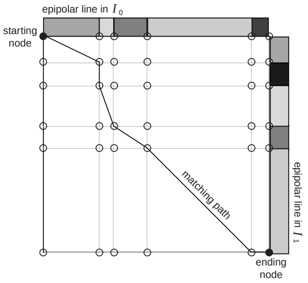


Fig. 2. Epipolar pair matching graph.

Fix a pair of conjugate epipolar lines $\ell_0 \in \mathcal{P}_0(i)$ and $\ell_1 \in \mathcal{P}_1(i)$ of length N, and define $\mathcal{A}$ as the set of monotonic sequences of straight lines which connect the left endpoints of (ℓ_0, ℓ_1) to the right endpoints in the graph $\ell_0 \times \ell_1$. For a cost function $J : \mathcal{A} \to \mathbb{R}^+$, let

$$C^i(\ell_0, \ell_1) = \min_{x \in \mathcal{A}} J(x) \qquad (8)$$

Finding C^i by an exhaustive search over $\mathcal{A}$ without any a priori notion of the correct matching path is computationally expensive. However, let $\hat{x}$ be a prior estimate of the matching path, and define $\mathcal{B} = \{x \in \mathcal{A} \mid d(x, \hat{x}) \le \epsilon\}$ for some distance function $d : \mathcal{A}^2 \to \mathbb{R}^+$. Then define

$$M^i(\ell_0, \ell_1 \mid \hat{x}) = \min_{x \in \mathcal{B}} J(x) \qquad (9)$$

For a distance function d based on the L_1 norm, $\mathcal{B}$ is approximately $\frac{2\epsilon N - \epsilon^2}{N^2}$ the size of $\mathcal{A}$, a substantial difference when ϵ is small relative to N.

3.3. Error Analysis

We use the recurrence $\hat{x}(i \mid i) = M^i\hat{T}^i(\hat{x}(i-1 \mid i-1))$, where $\hat{T}^i$ is an estimate of the true T^i induced by camera dynamics as in Section 3.1. We are interested in bounding the difference between the output of the $(\hat{T}, M)$ algorithm and the true correspondence $x^*(i)$. To this end, we fix a norm $\|\cdot\|$ on $(\mathbb{R}^2 \times \mathbb{R}^2)^N$ and define two estimation errors at each time i:

$$\begin{aligned} \epsilon_{TM}(i) &= \|x^*(i) - \hat{x}(i \mid i)\| && (10) \\ \epsilon_D(i) &= \|\tilde{x}(i) - \hat{x}(i \mid i)\| && (11) \end{aligned}$$

We assume that there exist constants α, $\hat{\alpha}$, β, γ, δ, and ρ such that for all i,

$$\|T^i(x) - \hat{T}^i(x)\| \;\le\; \gamma \qquad (12)$$

$$\begin{aligned}
\|T^i(x) - T^i(y)\| &\leq \alpha\, \|x - y\| && (13)\\
\|\hat{T}^i(x) - \hat{T}^i(y)\| &\leq \hat{\alpha}\, \|x - y\| && (14)\\
\|x^*(i+1) - T^i(x^*(i))\| &\leq \delta && (15)\\
\|\tilde{x}(i) - x^*(i)\| &\leq \rho && (16)\\
M^i(\tilde{x}(i)) &= \tilde{x}(i) && (17)\\
\|M^i(x) - M^i(y)\| &\leq \beta\, \|x - y\| && (18)
\end{aligned}$$

The error in the approximation of T^i by $\hat{T}^i$ is bounded by γ, where γ is a function of the data and the projective transformation estimation algorithm. The parameters of the Lipschitz conditions (13) and (14) can be extracted from the rotation parameters of the cameras, the derivative of the projective transformation function, and the finite extent of the image planes. The δ parameter reflects scene dynamics that are not modeled by the rotation of the cameras. The error in the correspondence estimator C^i is bounded by ρ. We require that the output $\tilde{x}(i)$ of C^i is fixed by M^i, which is the case when M^i is a restriction of C^i over a smaller domain. In the example of Section 3.2, the parameter β of the Lipschitz condition (18) is a function of the length of the epipolar lines and the diameter ϵ of the search neighborhood.

Theorem 1 *If the operators T^i, $\hat{T}^i$, C^i and M^i satisfy (12)-(18) with $\hat{\alpha}\beta < 1$, the $(\hat{T}, M)$ algorithm is stable in the sense that*

$$\operatorname{limsup}\, \epsilon_{TM}(i) \leq \rho + \frac{\beta((\alpha+1)\rho + \delta + \gamma)}{1 - \hat{\alpha}\beta} \qquad (19)$$

Proof Sketch. It can be shown that the errors $\epsilon_D(i)$ satisfy

$$\epsilon_D(i) \leq \hat{\alpha}\beta\, \epsilon_D(i-1) + \beta\, ((\alpha+1)\rho + \delta + \gamma) \qquad (20)$$

for $i = 1, \ldots, \infty$. The assumption $\hat{\alpha}\beta < 1$ guarantees boundedness of ϵ_D, and hence the asymptotic estimation error $\operatorname{limsup}\, \epsilon_D(i)$ is bounded by $\frac{\beta((\alpha+1)\rho+\delta+\gamma)}{1-\hat{\alpha}\beta}$. The result then follows by taking the limsup of the inequality $\epsilon_{TM}(i) \leq \rho + \epsilon_D(i)$. □

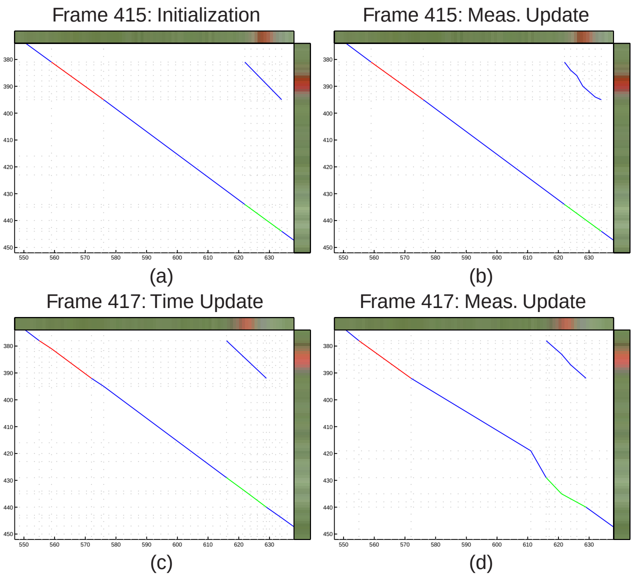
By (19), the recursive propagation algorithm can only approximate the true correspondence as well as the correspondence algorithm C^i. In particular, when $\rho = 0$, i.e. the operator C^i produces the true correspondence $x^*(i)$, the error in the $(\hat{T}, M)$ algorithm is bounded by a quantity which depends on the amount of object motion in the scene and the error in the approximation of T^i by $\hat{T}^i$. As these quantities decrease to zero, so does the asymptotic error of the $(\hat{T}, M)$ algorithm.

4. VIRTUAL VIDEO

Previous work (e.g. [7]) addressed the virtual view problem. Fix a scene and a triple of cameras $(\mathcal{C}_0, \mathcal{C}_1, \mathcal{C}_v)$. Given the pair of images $(\mathcal{I}_0, \mathcal{I}_1)$ of the scene produced by the uncalibrated cameras $(\mathcal{C}_0, \mathcal{C}_1)$, the problem is to synthesize the "virtual" image $\mathcal{I}_v$ of the scene from the perspective of $\mathcal{C}_v$. Under certain constraints on the position of the virtual camera with respect to the source cameras, this problem can be solved using a dense set of correspondences between the pair of input images $(\mathcal{I}_0, \mathcal{I}_1)$.

The virtual video problem is: for a pair of image sequences $(\{\mathcal{I}_0(i)\}, \{\mathcal{I}_1(i)\})$ of a dynamic scene produced by rotating, uncalibrated cameras $(\mathcal{C}_0, \mathcal{C}_1)$, synthesize the "virtual" image sequence $\{\mathcal{I}_v(i)\}$ of the scene from the perspective of a moving virtual camera $\mathcal{C}_v$.

One naïve solution to the virtual video problem is to treat it as a sequence of virtual view problems over a period of time, processing each image triple independently. However, since correspondences are expensive to obtain, this approach is prohibitively time-consuming. More importantly, it does not exploit the temporal regularity of the input video. Since the source video sequences fit into the framework of Figure 1, we apply the recursive algorithm discussed above to propagate the dense correspondences required to create each virtual image.



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

Fig. 3. (a) Initialization of the correspondence graph. (b) Measurement update of the initial estimate. (c) Time update to compensate for camera motion. (d) Measurement update to compensate for object motion.

To construct the correspondence and measurement update operators, we use a conjugate epipolar line matching approach with a modified version of the Ohta and Kanade [8] cost function. The monotonicity assumption is usually violated in our virtual video database, and our implementation handles these cases by searching the set of physically valid, piecewise monotonic matching paths through each

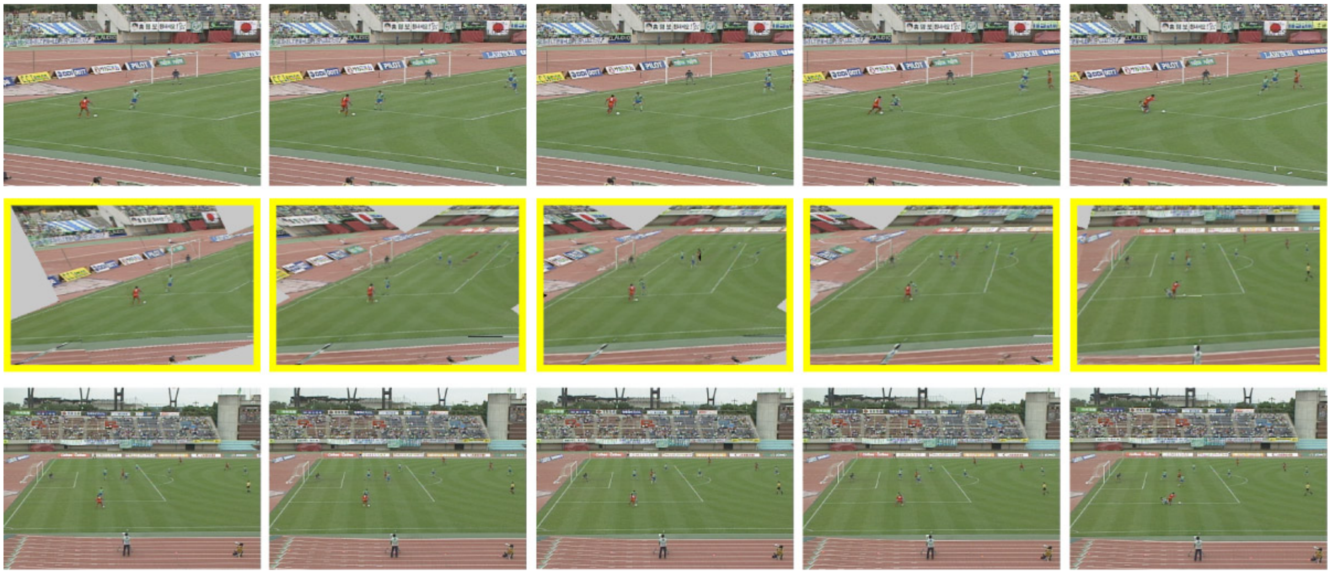


Fig. 4. Frames of virtual video. The top row contains original frames from camera 1, the bottom row contains original frames from camera 2 taken at the same time, and the middle row contains virtual images synthesized from the perspective of a camera moving along the baseline of the original camera pair.

graph $\ell_0 \times \ell_1$. We call these special piecewise monotonic paths *correspondence graphs*.

Figure 3 illustrates the results of our recursive propagation algorithm for a typical epipolar line pair from real video. The blue segments indicate points which are seen in both images, while the red and green segments indicate points which are seen in just $\mathcal{I}_0$ or $\mathcal{I}_1$, respectively. Moving objects in the video have been segmented and tracked, and this information is used to determine the structure of the correspondence graph for each epipolar line pair. The initial correspondence $x^*(0)$ for the background pixels in this example is approximated by a straight line through the matching path. This line is induced by an estimate of the projective transformation which relates a dominant plane in the image pair.

Figure 4 illustrates the final virtual video result, using view morphing [7] to render the virtual frames. The top and bottom rows are synchronized frames of video from two real cameras. The first camera undergoes a slow pan to the right over the course of the clip, while the second camera slowly zooms in. The virtual camera moves smoothly from the perspective of the first camera to the perspective of the second, while the virtual images evolve at the same rate as the input video. The virtual video contains arrangements of objects (e.g. the goalie and the goalpost) that did not occur in either of the original sequences. Operating with correspondence graphs instead of monotonic matching paths makes this realism possible.

Our current implementation produces virtual video at about 20 frames per minute. The only user intervention required is a sparse set of point correspondences in the initial frame pair (used to estimate the fundamental matrix and the projective transformation relating the dominant plane in the image pair), and segmentation and tracking information for moving objects in each frame (used to construct correct correspondence graphs). In future implementations it would be convenient to incorporate the segmentation and tracking algorithm inline, and to automatically detect when the propagation process destabilizes.

5. REFERENCES

[1] Q.-T. Luong and O.D. Faugeras. The Fundamental Matrix: Theory, Algorithms, and Stability Analysis. *IJCV*, Vol. 17, No. 1, pp. 43-76, 1996.

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

[3] R.I. Hartley. Theory and Practice of Projective Rectification. *IJCV*, Vol. 35, No. 2, pp. 115-127, November 1999.

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

[5] H.S. Sawhney, S. Ayer, and M. Gorkani. Model-based 2D and 3D Dominant Motion Estimation for Mosaicing and Video Representation. *Proc. ICCV '95*, pp. 583-590, 1995.

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

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

[8] 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.