Video Alignment

Final Report

Spring 2005
Prof. Brian Evans
Multidimensional Digital Signal Processing Project
The University of Texas at Austin

Omer Shakil

Abstract

This report describes a method to align two videos. The idea can be extended for alignment of multiple video sequences. The sequences are recorded by un-calibrated video cameras with fixed (but unknown) internal parameters. It will be shown that by using a combination of spatial information, temporal changes and the inbuilt frame-to-frame transformation of the video sequences, efficient video alignment can be achieved. The sequences used in the algorithm described may have no or minimal spatial overlap and the camera sources are also allowed free movement in space. Video alignment gives rise to a wide variety of new applications that are not possible when only image-to-image alignment is used.

1. Introduction

The problem of image alignment has been extensively studied, and successful solutions have been suggested [4, 6]. The issue here is to estimate point correspondences between two images, i.e. given any pixel *(x, y)* in one image, find its corresponding pixel *(x', y')* in the other image, where $x' = x + u$, $y' = y + v$, and *(u, v)* is the calculated spatial displacement vector. The image alignment techniques can be divided into two broad categories: the first category includes feature-based approaches [7], in which common features are detected and matched across the two images; the second category comprises of methods that directly match image intensities [4]. As both of these categories rely on the spatial coherence of the given images, there must be a significant overlap between the two images for these techniques to work.

This report addresses the issue of video alignment. Here, the problem is that the sequences may not be synchronized; hence alignment in time is required as well. Given points *(x, y, t)* from video sequence *V* and *(x', y', t')* from video sequences *V',* the task is to recover the transformation between them i.e. find *(u, v, w)* such that $(x', y', t') = (x+u, y+v, t+w)$. The temporal "redundancy" in successive frames of a video can be used to move a step beyond from the traditional image alignment techniques that exploit spatial coherence between the given images. Therefore, by using the temporal behavior of the videos (frame-to-frame transformations), alignment can be achieved even when the corresponding frames from each sequence have no spatial overlap between them, given the cameras are jointly moving [3, 5]. A novel method to achieve alignment with freely moving camera sources has been proposed in this report.

A variety of applications, esp. in surveillance and security systems, can benefit heavily from alignment of multiple video sources. A primary application is monitoring multiple video inputs (high security areas) and the output being displayed in one composite video sequence,

Technique	Spatial	Temporal	Ambiguities	Overlap	Source	Extra Processing	Complexity
Image-Based [1, 5]	Yes	No	Many	Yes	Stationary	Coarse-to-fine	Low
Feature-Based [1, 7, 9]	No	Yes	Few	Yes	Jointly Moving	Feature Extraction	Medium
Direct-Based [1,4, 5]	Yes	Yes	No	Yes	Jointly Moving	Coarse-to-fine	Medium
Video Matching [2]	Yes	Yes	No	Yes	Similarly Moving	Statistical Methods	High
Non-Overlapping [3,8,10]	Yes	Yes	No	No	Jointly Moving	Transformations	Medium
Proposed Method	Yes	Yes	No	No	Freely Moving	Transformations	Medium

Table 1: Summary of the advantages and disadvantages of various approaches for video alignment

instead of various monitors dedicated for each individual input source. Other applications include generation of wide screen movies and super-resolution in time and space.

2. Overview

A summary of the advantages and disadvantages of various approaches with different constraints that attempt to solve the problem of video alignment is shown in Table 1. All of the existing methods impose some restrictions on the movement of camera sources, which is overly constraining and often undesirable. This report describes a simple and innovative method to achieve alignment with freely moving video cameras. The approach can be divided into two parts: initially determining the inter-camera transformation with jointly moving sources [3, 8] followed by utilizing the inter-frame relations [4, 6] to recover the later transformations. A reference frame is chosen in the initial part of the video which is then used to find the overall transformation between any two points from the image sequences.

The first step is to determine the inter-camera homography, *H*, with jointly moving sources (see Figure 1) based on the algorithm by Caspi and Irani [3]. Therefore, given two video sequences $S = I_1, \ldots, I_{n+1}$ and $S' = I'_1, \ldots, I'_{m+1}$, we can recover *H* using frame-to-frame

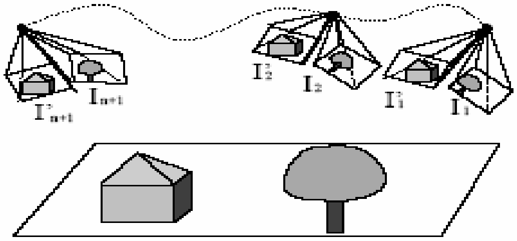

Figure 1: Two video cameras are attached to each other, so that they have the same center of projection, but non-overlapping fields-of-view. The two cameras are moved jointly in space, producing two separate video sequences S and S' [1]

transformations, $T_1, \ldots, T_n$ and $T'_1, \ldots, T'_m$, where T_i represents the transformation between frame I_i and I_{i+1}. This idea can be extended to achieve synchronization in time as well. In that case an offset needs to be handled as the frames I_i and I_i' may not be the corresponding frames in time. This report assumes that the given video sequences have planar or distant scenes (can be extended to 3D scenes as described in [3, 8]) and that the offset is already known (can be found using the method described in [3, 8] or from the timestamps encoded in each of the two sequences). Once H is computed, we can now allow free camera movement. The new frames can be registered on the previous coordinate system by keeping track of all successive frame-to-frame transformations.

3. Recovering Inter-Camera Homography, *H*

The algorithm described is for synchronized planar (or distant) scenes. This can be extended for non-synchronized or 3D scenes [3, 8]. Let P be any 3D point in the given planar scene. Use p_i and p_i' to denote its image coordinates in frames I_i and I_i' respectively (the point P. is not required to be visible in both of the frames). Similarly, use p_{i+1} and p_{i+1}' to denote its image coordinates in frames I_{i+1} and I_{i+1}' respectively. Using transformations, T_i and T_i', we have: $p_{i+1} \cong T_i p_i$ and $p'_{i+1} \cong T_i p'_i$. Moreover, because there is a fixed homography H between the two sequences, then: $p'_i \cong Hp_i$ and $p'_{i+1} \cong Hp_{i+1}$. Therefore, in case of planar or distant scene, the temporally corresponding transformations T_i and T_i' are related by the same fixed inter-camera homography H:

$$HT_i p_i \cong Hp_{i+1} \cong p'_{i+1} \cong T_i' p'_i \cong T_i' Hp_i \qquad (1)$$

As Equation (1) holds for all points p_i, all scalars will be equal, hence we may write (given H is non-singular): $T'_i \cong HT_iH^{-1}$ or

$$T'_i \cong s_i HT_iH^{-1} \qquad (2)$$

where s_i is a frame-dependent scale factor. Therefore, there is a similarity relation (conjugacy relation) between two matrices T_i and T_i' (upto a scale factor). Hence, s_i can be estimated using least squares minimization (three equations, one unknown) as we have $eig(T'_i) \cong s_i eig(T_i)$, where $eig(A) = [\lambda_1, \lambda_2, \lambda_3]^T$ is a 3x1 vector containing the eigenvalues of a 3x3 matrix A (in descending order). Alternatively to avoid the need to compute these scale factors, the input transformations can be normalized to have determinant equal to 1. After computing all s_i, Equation (2) can be rewritten as:

$$s_i H T_i - T'_i H = 0 \tag{3}$$

This is linear in the unknown components of homography matrix, *H*. Rearranging these unknowns in a column vector $\vec{h} = [H_{11}H_{12}H_{13}H_{21}H_{22}H_{23}H_{31}H_{32}H_{33}]^T$, Equation (3) can be rewritten as a set of homogenous equations: $M_i\vec{h} = \vec{0}$, where M_i is a 9x9 matrix defined by T_i, T'_i and s_i:

$$M_i = \begin{pmatrix} s_iT_i^t - T_{i11}^{'}I & -T_{i12}^{'}I & -T_{i13}^{'}I \\ -T_{i21}^{'}I & s_iT_i^t - T_{i22}^{'}I & -T_{i23}^{'}I \\ -T_{i31}^{'}I & -T_{i32}^{'}I & s_iT_i^t - T_{i33}^{'}I \end{pmatrix} \tag{4}$$

where *I* is a 3x3 identity matrix. We use all available constraints from all pairs of transformations to compute *H*. The constraints from all the transformation T_1, …, T_n and T'_1, …, T'_n can be combined into a single set of linear equations in $\vec{h}$, such that $A\vec{h} = \vec{0}$, where A in a 9nx9 matrix, $A = [M_1 M_2 ... M_n]^T$. This forms a homogenous set of linear equations that can be solved in a variety of ways. In particular, $\vec{h}$ may be recovered by computing the eigenvector which corresponds to the smallest eigenvalue of the matrix A^tA [3, 8].

The approach for a non-planar scene is similar; transformation matrices and epipolar constraints together define the complete 3D transformation. This approach works when the calculated transformations are extremely accurate. Spencer and Shah [10] have defined four new measures that are robust to noisy signals: Similarity, Roll motion, Translation Magnitude and Translation Direction. This algorithm works only with stationary or jointly moving camera sources.

4. Allowing Free Camera Movement

This section describes a simple and novel approach to allow free camera movement once the inter-camera homography has been found using the algorithm described in the last section. The idea here is to utilize the “redundancy” of the video sequences i.e. to use the successive inter-frame relations to find the overall transformations between the two sequences even when the cameras are moving freely in space. This can be achieved by selecting one reference frame from the initial part of the video sequence (learning phase). This is shown in light grey in Figure 2 (a). The found inter-camera homography, *H* relates the two points between these frames. $T_L=(T_n)(T_{n+1})(T_{n+2})$ and $T_R=(T'_n)(T'_{n+1})(T'_{n+2})$ denote the overall successive frame-to-frame transformations incurred to reach the current frames (shown in dark grey). The overall transformation between any two points in the current frames is given by $(T_L)^{-1}\, H\, T_R$.

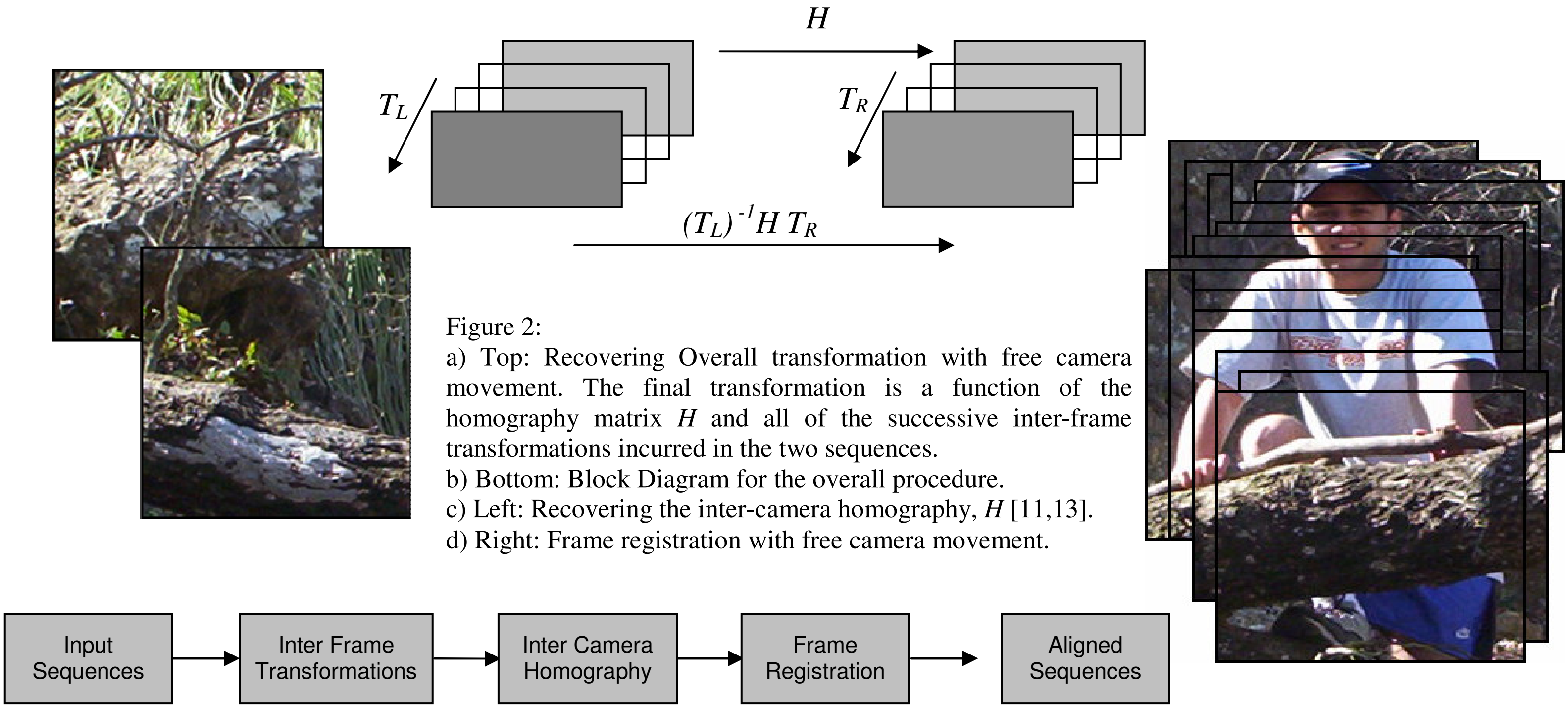

Figure 2:
a) Top: Recovering Overall transformation with free camera movement. The final transformation is a function of the homography matrix *H* and all of the successive inter-frame transformations incurred in the two sequences.
b) Bottom: Block Diagram for the overall procedure.
c) Left: Recovering the inter-camera homography, *H* [11,13].
d) Right: Frame registration with free camera movement.

5. Results and Implementation Details

The block diagram for the overall procedure is shown in Figure 2(b). Each of these steps will be addressed individually in detail.

Inter-Frame Transformations

The first step is to find the inter-frame transformations for both sequences. Every n^{th} frame is used for this procedure. The value for *n* specifies the trade-off between accuracy and efficiency. The larger the value of *n*, the faster the algorithm will finish compromising on the accuracy of the results. On the other hand, $n = 1$ implies that all frames should be used, which will result in the most accurate values but may take a considerably long time to terminate.

A couple of approaches [4, 6] were tested to find the transformation between two frames. The hierarchical iterative approach by Bergen et al. [4] gave better results than the recursive approach by Szeliski [6, 12]. Therefore, the former method was used to find the inter-frame transformations. The implementation complexity was also less compared to Szeliski's approach [6, 12]. Table 2 shows the recovered transformations using Bergen's approach.

To further confirm the accuracy of found transformations, a reliability threshold was used. The method employed was to find both the forward as well as the backward transformation between consecutive frames. Ideally, the found transformations should be the inverse of each other. A threshold was then used to get rid of the erroneous transformations [3,8].

Actual	$\begin{pmatrix} 1.2 & 0 & -10 \\ 0 & 1.2 & -10 \\ 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 10 \\ 0 & 1 & 20 \\ 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 0.9397 & -0.3420 & 20 \\ 0.3420 & 0.9397 & -30 \\ 0 & 0 & 1 \end{pmatrix}$
Recovered	$\begin{pmatrix} 1.2000 & 0.0000 & -9.9992 \\ 0.0000 & 1.1999 & -9.9949 \\ 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 1.0000 & 0.0000 & 10.0000 \\ 0.0000 & 1.0000 & 20.0000 \\ 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 0.9397 & -0.3420 & 20.0018 \\ 0.3420 & 0.9396 & -29.9931 \\ 0 & 0 & 1 \end{pmatrix}$

Table 2: Actual vs. Recovered Transformations using [4] for simple scaling, translation

Inter-Camera Homography

The next step is to compute the inter-camera homography using the algorithm previously described. This requires all the inter-frame transformations as an input, which was found in the previous step.

As described in [3], all of the reliable transformation matrices were normalized to have determinant equal to one, to avoid the need to compute scale factor s_i for each transformation pair. By removing this step of approximating scale factors, the recovered inter-camera homography was improved. However, the results were still not perfect (see Figure 2(c)). A synthetically generated video sequence was tested with a translation of 150 pixels in x-direction and 50 pixels in y-direction. The recovered translation was 131.5 pixels in x-direction and 65.0 pixels in y-direction [11, 13].

Frame Registration

Once the inter-camera homography is recovered, the cameras are allowed to move freely in space. As described previously, by choosing a reference frame from the initial part of the video and by computation of all successive inter-frame transformations, the new frames can be easily registered in the old coordinate axes. The frames are warped back using a bilinear transformation to the previous frames. The overlapping values can be averaged together or simply one of the two can be chosen. The results of this phase are very accurate, a mosaic generated (instead of panning so that the image could be shown) using inter-frame transformations is shown in Figure 2(d). The black outlines are deliberately marked for each frame to aid viewing. The sequence was generated manually using simple translations. The result would be accurate for any other transformations as well (see Table 2).

6. Conclusion

This paper introduces a novel approach for aligning two sequences with freely moving camera sources. The method works even when there is no common spatial information between the sequences [3, 8]. This was made possible by replacing the need for "coherent appearance" (which is an essential requirement for standard image alignment techniques), with the requirement of "coherent temporal behavior", which is often easier to satisfy. The "redundancy" in the video sequences i.e. the inter-frame transformations is then exploited to allow free camera movement.

The idea has been proposed for planar synchronized scenes, but as suggested in the report, it can be easily extended to 3D scenes and to recover alignment in time as well [3, 8]. The results of the implementation are shown in Figure 2(c, d).

7. References

[1] Y. Caspi and M. Irani, "Spatio-Temporal Alignment," *Proc. IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1409-1424, November 2002.

[2] P. Sand and S. Teller, "Video Matching," *Proc. ACM Transactions on Graphics*, pp. 592-599, August 2004.

[3] Y. Caspi and M. Irani, "Aligning non-overlapping sequences," *Proc. International Journal of Computer Vision*, pp. 39-51, June 2002.

[4] J. Bergen, P. Anandan, K. Hanna, and R. Hingorani, "Hierarchical Model-Based Motion Estimation", *Proc. European Conference on Computer Vision (ECCV),* pp. 237–252, May 1992.

[5] Y. Caspi and M. Irani, "A step towards sequence-to-sequence alignment," *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 682–689, June 2000.

[6] R. Szeliski. "Video mosaics for virtual environments", *IEEE Computer Graphics and Applications*, pp. 22-30, March 1996.

[7] G. P. Stein, "Tracking from multiple view points: Self-calibration of space and time," *Proc. DARPA IU Workshop*, pp. 1037–1042, 1998.

[8] Y. Caspi and M. Irani, "Alignment of non-overlapping sequences," *Proc. International Conference on Computer Vision*, pp. 76-83, July 2001.

[9] L. Lee, R. Romano and G. Stein, "Monitoring Activities from Multiple Video Streams: Establishing a Common Coordinate Frame," *Proc. IEEE Tranactions on Pattern Analysis and Machine Intelligence*, pp. 758-767, August 2000.

[10] L. Spencer and M. Shah, "Temporal Synchronization from Camera Motion," *Proc. Asian Conference on Computer Vision (ACCV)*, Jan 2004. http://www.cs.ucf.edu/~vision/projects/time_shift/time_shift.htm

[11] Matlab Code from Computer Vision Laboratory Website, University of Central Florida. http://www.cs.ucf.edu/~vision/source.html

[12] M. Sheikh, "Recovering Inter-Frame Transformations using Szeliski's Approach Matlab Code, " *Personal Correspondence (Email)*, Apr 2005.

[13] X. Cao, "Recovering Inter-Camera Homography Matlab Code," *Personal Correspondence (Email)*, Apr 2005.