Segmenting, modeling and matching video clips containing multiple moving objects

Fred Rothganger, Svetlana Lazebnik, Cordelia Schmid, Jean Ponce

Segmenting, Modeling, and Matching Video Clips Containing Multiple Moving Objects

Fred Rothganger[1], Svetlana Lazebnik[1], Cordelia Schmid[2], and Jean Ponce[1]
[1] Department of Computer Science and Beckman Institute
University of Illinois at Urbana-Champaign; Urbana, IL 61801, USA
[2] INRIA Rhône-Alpes; 665, Avenue de l'Europe; 38330 Montbonnot, France

Abstract. *This paper presents a novel representation for dynamic scenes composed of multiple rigid objects that may undergo different motions and be observed by a moving camera. Multi-view constraints associated with groups of affine-invariant scene patches and a normalized description of their appearance are used to segment a scene into its rigid parts, construct three-dimensional projective, affine, and Euclidean models of these parts, and match instances of models recovered from different image sequences. The proposed approach has been implemented, and it is applied to the detection and recognition of moving objects in video sequences and the identification of shots that depict the same scene in a video clip (shot matching).*

1. Introduction

This paper addresses the problem of recognizing three-dimensional (3D) objects in video clips. Viewpoint invariants (or *invariants* for short) provide a natural indexing mechanism for object recognition tasks. Unfortunately, although planar objects and certain simple shapes (e.g., bilaterally symmetric ones) admit invariants, general 3D shapes do not [8], which is the main reason why invariants have fallen out of favor after an intense flurry of activity in the early 1990s [24, 25]. We have shown in [29] that invariants provide a valuable *local* description for 3D objects: Indeed, although smooth surfaces are almost never planar in the large, they are *always* planar in the small—that is, sufficiently small surface patches can always be thought of as being comprised of coplanar points. The surface of a solid can thus be represented by a collection of small patches, their invariants, *and* a description of their 3D spatial relationship. Specifically, we proposed in [29] to represent the surface of a solid by a collection of (small, planar) *affine-invariant patches* as proposed by Lindeberg and Gårding [20] and Mikolajczyk and Schmid [23], *and* a description of their 3D spatial relationship in terms of the multi-view geometric consistency constraints studied in the structure-from-motion literature [10, 17, 32].

The approach proposed in [29] was intended for tasks such as modeling rigid 3D objects from a few unregistered still pictures and identifying these models in photographs of cluttered scenes [21, 33]. It is combined in this paper with the proven technology now available for tracking rigid and articulated objects [6, 7, 32, 35] and a new (as far as we know) *locally* affine model of image projection, resulting in a novel algorithm for segmenting a dynamic scene into its rigid parts, constructing projective, affine and Euclidean models of these parts, and matching instances of the models recovered from different image sequences.

The proposed approach has been implemented, and it is applied to the identification of shots that depict the same scene in a film (shot matching) [1, 2, 5, 30, 31, 34], a fundamental task in the annotation/indexing context where videos are commonly segmented into shots [14, 19]. Preliminary results are presented using both laboratory videos and shots from the film "Run Lola Run".

2. Local Invariants and Global 3D Constraints

The approach proposed in [29] combines a normalized representation of local surface appearance in terms of local affine-invariant patches [20, 23] with the global 3D affine multi-view constraints studied in the structure-from-motion literature [10, 17, 32] to effectively represent the surfaces of solids in modeling and recognition tasks. We briefly recall the main ideas of this approach in Sections 2.1 and 2.2 before introducing in Section 2.3 a new, *locally* affine model of the image formation process capable of handling the large *global* perspective distortions common in urban scenes for example.

2.1. Affine-Invariant Patches

Operators capable of finding affine-invariant [3, 23, 28, 33] image descriptors in the neighborhood of salient image features ("interest points" [16]) have recently been proposed in the context of wide-baseline stereo matching and image retrieval. We use an implementation of the affine-invariant region detector developed by Mikolajczyk and Schmid [23] for low-level image description. Its output consists of a set of image patches in the shape of parallelograms, together with the corresponding affine *rectifying transformations*. The transformation $\mathcal{R}$ associated with

each patch maps the corresponding parallelogram onto a square with unit edge half-length centered at the origin (Fig. 1). The rectified patch is a *normalized* representation of the local surface appearance that is invariant under planar affine transformations. Such transformations are induced by arbitrary changes in viewpoint under the affine (orthographic, weak-perspective, or para-perspective) projection model as well as the *locally* affine model introduced in Section 2.3.

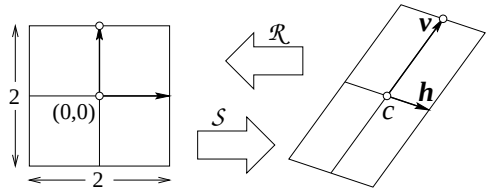

Figure 1: Geometric interpretation of the rectifi cation matrix $\mathcal{R}$ and its inverse $\mathcal{S}$.

The rectifying transformation associated with a planar patch and its inverse can be represented by two 2×3 matrices $\mathcal{R}$ and $\mathcal{S}$ that map homogeneous (affine) plane coordinates onto non-homogeneous ones. The column vectors of the matrix $\mathcal{S}$ admit a simple geometric interpretation [29]: The third column $\boldsymbol{c}$ of $\mathcal{S}$ is the coordinate vector of the patch center c, and its first two columns $\boldsymbol{h}$ and $\boldsymbol{v}$ are respectively the coordinate vectors of the "horizontal" and "vertical" vectors joining c to the sides of the patch (Fig. 1).

2.2. Affine Projection Constraints

Let us consider n patches observed in m images (we will assume *for the time being* that all patches are visible in all images), and denote by $\mathcal{S}_{ij}$ the corresponding 2×3 matrices as defined in Section 2.1. Here i and j respectively serve as image and patch indices, with $i = 1, \ldots, m$ and $j = 1, \ldots, n$. Under affine projection, the matrix $\mathcal{S}_{ij}$ records the projection of a parallelogram drawn on the surface into the corresponding image. Thus it can be written as $\mathcal{S}_{ij} = \mathcal{M}_i \mathcal{N}_j$, where $\mathcal{M}_i$ is the projection matrix associated with image number i and

$$\mathcal{N}_j = \begin{bmatrix} \boldsymbol{H}_j & \boldsymbol{V}_j & \boldsymbol{C}_j \\ 0 & 0 & 1 \end{bmatrix}$$

gives the position and shape of patch j on the surface of the object. The vectors $\boldsymbol{H}_j$, $\boldsymbol{V}_j$, and $\boldsymbol{C}_j$ are the 3D analogs of $\boldsymbol{h}_j$, $\boldsymbol{v}_j$, and $\boldsymbol{c}_j$ and have a similar interpretation. We follow Tomasi and Kanade [32] and pick the center of mass of the observed patches' centers as the origin of the world coordinate system, and the center of mass of these points' projections as the origin of every image coordinate system: In this case, the projection matrices reduce to $\mathcal{M}_i = [\mathcal{A}_i \quad \boldsymbol{0}]$, where $\mathcal{A}_i$ is a 2×3 matrix, and

$$\mathcal{S}_{ij} = \mathcal{A}_i \mathcal{B}_j, \text{ where } \mathcal{B}_j = [\boldsymbol{H}_j, \boldsymbol{V}_j, \boldsymbol{C}_j]. \tag{1}$$

It follows that the reduced $2m \times 3n$ matrix

$$\hat{\mathcal{S}} = \hat{\mathcal{A}}\hat{\mathcal{B}}, \text{ where } \hat{\mathcal{A}} \stackrel{\text{def}}{=} \begin{bmatrix} \mathcal{A}_1 \\ \ldots \\ \mathcal{A}_m \end{bmatrix}, \; \hat{\mathcal{B}} \stackrel{\text{def}}{=} [\, \mathcal{B}_1 \quad \ldots \quad \mathcal{B}_n \,],$$

has at most rank 3. Singular value decomposition can be used as in Tomasi and Kanade [32] to factorize $\hat{\mathcal{S}}$ and compute estimates of the matrices $\hat{\mathcal{A}}$ and $\hat{\mathcal{B}}$ that minimize the squared Frobenius norm of the matrix $\hat{\mathcal{S}} - \hat{\mathcal{A}}\hat{\mathcal{B}}$. The residual (normalized) Frobenius form $|\hat{\mathcal{S}} - \hat{\mathcal{A}}\hat{\mathcal{B}}|/\sqrt{3mn}$ of this matrix can be interpreted geometrically as the root-mean-squared distance (in pixels) between the predicted and observed values of of $\boldsymbol{c}_{ij}$, $\boldsymbol{h}_{ij}$, and $\boldsymbol{v}_{ij}$.

2.3. Locally Affine Projection Constraints

We assume in this section that the relief of each patch is small compared to the overall depth of the scene, so that an affine projection model is appropriate for *each* patch, yet a *global* affine projection model is inappropriate (this is the case for scenes with important perspective distortions such as the street scenes used in some of our experiments). A *local* affine model is obtained by linearizing the perspective projection equations in the neighborhood of the patch center. Consider the homogeneous projection equation

$$(\boldsymbol{a}_3 \cdot \boldsymbol{P} + 1) \begin{bmatrix} \boldsymbol{p} \\ 1 \end{bmatrix} = \mathcal{M}\boldsymbol{P} = \begin{bmatrix} \mathcal{A} & \boldsymbol{b} \\ \boldsymbol{a}_3^T & 1 \end{bmatrix} \boldsymbol{P},$$

where $\mathcal{M}$ is the perspective projection matrix, $\mathcal{A}$ is a 2×3 sub-matrix of $\mathcal{M}$, $\boldsymbol{p}$ is the non-homogeneous coordinate vector for the point in the image, and $\boldsymbol{P}$ is the homogeneous coordinate vector of the point in 3D. We can write the perspective projection mapping as

$$\boldsymbol{p} = f(\boldsymbol{P}) = \frac{1}{\boldsymbol{a}_3 \cdot \boldsymbol{P} + 1}(\mathcal{A}\boldsymbol{P} + \boldsymbol{b}), \tag{2}$$

and a Taylor expansion of order 1 of the function f in $\boldsymbol{P}$ yields $f(\boldsymbol{P} + \boldsymbol{\delta P}) = \boldsymbol{p} + \boldsymbol{\delta p} = f(\boldsymbol{P}) + f'(\boldsymbol{P})\boldsymbol{\delta P}$, or

$$\begin{aligned} \boldsymbol{\delta p} &= f'(\boldsymbol{P})\boldsymbol{\delta P} \\ &= \frac{\mathcal{A}(\boldsymbol{a}_3 \cdot \boldsymbol{P} + 1) - (\mathcal{A}\boldsymbol{P} + \boldsymbol{b})\boldsymbol{a}_3^T}{(\boldsymbol{a}_3 \cdot \boldsymbol{P} + 1)^2}\boldsymbol{\delta P} \\ &= \frac{1}{\boldsymbol{a}_3 \cdot \boldsymbol{P} + 1}(\mathcal{A} - \boldsymbol{p}\boldsymbol{a}_3^T)\boldsymbol{\delta P}. \end{aligned}$$

In particular, consider a patch defined by its center $\boldsymbol{C}$ and the directions $\boldsymbol{H}$ and $\boldsymbol{V}$. Taking $\boldsymbol{P} = \boldsymbol{C}$ and $\boldsymbol{\delta P} = \boldsymbol{H}$ (resp. $\boldsymbol{V}$) and $\boldsymbol{\delta p} = \boldsymbol{h}$ (resp. $\boldsymbol{v}$) yields

$$(\boldsymbol{a}_3 \cdot \boldsymbol{C} + 1)\,[\,\boldsymbol{h} \quad \boldsymbol{v}\,] = (\mathcal{A} - \boldsymbol{c}\boldsymbol{a}_3^T)\,[\,\boldsymbol{H} \quad \boldsymbol{V}\,]. \tag{3}$$

Finally, taking $\boldsymbol{c} = \boldsymbol{p}$ in (2) yields

$$\begin{aligned} \boldsymbol{c}(\boldsymbol{a}_3 \cdot \boldsymbol{C} + 1) &= \mathcal{A}\boldsymbol{C} + \boldsymbol{b}, \\ \boldsymbol{c} &= (\mathcal{A} - \boldsymbol{c}\boldsymbol{a}_3^T)\boldsymbol{C} + \boldsymbol{b}. \end{aligned} \tag{4}$$

Given a fixed projection matrix $\mathcal{M}$, putting Eqs. (3) and (4) together now yields a system of 6 linear equations in the 9 unknown coordinates of $\boldsymbol{H}$, $\boldsymbol{V}$, and $\boldsymbol{C}$:

$$\left[\begin{array}{c|c|c} \mathcal{A}-\boldsymbol{c}\boldsymbol{a}_3^T & \boldsymbol{0}^T & -\boldsymbol{h}\boldsymbol{a}_3^T \\ \boldsymbol{0}^T & \mathcal{A}-\boldsymbol{c}\boldsymbol{a}_3^T & -\boldsymbol{v}\boldsymbol{a}_3^T \\ \boldsymbol{0}^T & \boldsymbol{0}^T & \mathcal{A}-\boldsymbol{c}\boldsymbol{a}_3^T \end{array}\right] \left[\begin{array}{c} \boldsymbol{H} \\ \boldsymbol{V} \\ \boldsymbol{C} \end{array}\right] = \left[\begin{array}{c} \boldsymbol{h} \\ \boldsymbol{v} \\ \boldsymbol{c} \end{array}\right] - \left[\begin{array}{c} \boldsymbol{0} \\ \boldsymbol{0} \\ \boldsymbol{b} \end{array}\right]. \tag{5}$$

Given fixed vectors $\boldsymbol{H}$, $\boldsymbol{V}$, and $\boldsymbol{C}$, Eqs. (3) and (4) also provide a system of 6 linear equations in the 11 unknown entries of $\mathcal{M}$:

$$\left[\begin{array}{c|c|c} \mathcal{H} & -\boldsymbol{h}\boldsymbol{C}^T - \boldsymbol{c}\boldsymbol{H}^T & 0_2 \\ \mathcal{V} & -\boldsymbol{v}\boldsymbol{C}^T - \boldsymbol{c}\boldsymbol{V}^T & 0_2 \\ \mathcal{C} & -\boldsymbol{c}\boldsymbol{C}^T & \mathrm{Id}_2 \end{array}\right] \left[\begin{array}{c} \boldsymbol{a}_1 \\ \boldsymbol{a}_2 \\ \boldsymbol{a}_3 \\ \boldsymbol{b} \end{array}\right] = \left[\begin{array}{c} \boldsymbol{h} \\ \boldsymbol{v} \\ \boldsymbol{c} \end{array}\right], \tag{6}$$

where 0_2 and Id_2 are respectively the 2×2 zero and identity matrices, $\boldsymbol{a}_1^T$ and $\boldsymbol{a}_2^T$ are the first two rows of $\mathcal{M}_1$, and

$$\mathcal{H} = \begin{bmatrix} \boldsymbol{H}^T & \boldsymbol{0}^T \\ \boldsymbol{0}^T & \boldsymbol{H}^T \end{bmatrix}, \ \mathcal{V} = \begin{bmatrix} \boldsymbol{V}^T & \boldsymbol{0}^T \\ \boldsymbol{0}^T & \boldsymbol{V}^T \end{bmatrix}, \ \mathcal{C} = \begin{bmatrix} \boldsymbol{C}^T & \boldsymbol{0}^T \\ \boldsymbol{0}^T & \boldsymbol{C}^T \end{bmatrix}.$$

Given n patches observed in m images, we can use the following iterative process[1] to solve for the corresponding matrices $\mathcal{M}_i$ ($i = 1, \ldots, m$) and vectors $\boldsymbol{H}_j$, $\boldsymbol{V}_j$, and $\boldsymbol{C}_j$ ($j = 1, \ldots, n$):

(1) Initialize the vectors $\boldsymbol{H}_j, \boldsymbol{V}_j, \boldsymbol{C}_j$ ($j = 1, \ldots, n$) using the affi ne method described in section 2.2.
(2) Repeat until convergence:
(a) For $i = 1, \ldots, m$, use linear least-squares to solve for $\mathcal{M}_i$ by stacking the n_i instances of Eq. (6) associated with the patches observed in image i.
(b) For $j = 1, \ldots, n$, use linear least-squares to solve for $\boldsymbol{H}_j, \boldsymbol{V}_j, \boldsymbol{C}_j$ by stacking the m_j instances of Eq. (5) associated with the images containing patch j.

Given the ambiguity of projective structure from motion, we have $6mn$ equations in $11m + 9n - 15$ unknowns. These equations are redundant whenever $n \geq 2$ image tracks share at least $m \geq 3$ frames, and it is possible to judge whether the corresponding patches rigidly move together by solving for the structure and motion parameters and measuring as before the mean-squared distance in pixels between the predicted and measured values of the vectors $\boldsymbol{c}_{ij}$, $\boldsymbol{h}_{ij}$, and $\boldsymbol{v}_{ij}$. Note that, unlike factorization, this methods is readily adapted to the case where patches are only available in some of the images.

3. Model Construction and Motion Segmentation

Although it is relatively challenging to match patches between two widely separated views (such as discussed in [29]), it is easy to match them in a continuous image sequence, thanks in part to the recent emergence of reliable techniques for tracking rigid and articulated objects [6, 7, 32, 35], but also to the normalized appearance model of the patches themselves, which is (in principle) invariant under viewpoint and illumination changes. The remaining difficulties in this setting are the identification of groups of image patches that move rigidly together, and the effective estimation of the corresponding structure and motion parameters.

We propose in this section a simple approach to these two problems, assuming for simplicity that points moving rigidly together do so over all the frames in which they are visible (which may of course not be true in all videos). Our algorithm maintains a list M of growing models and a list T of unassigned tracks, updating the two lists and the structure and motion parameters associated with the elements of M at time stamps $t_1, \ldots, t_k$ regularly sampled over the m frames of the video. Here, a *track* is the set of images where a patch is visible, along with the corresponding matrices $\mathcal{S}_{ij}$, and a *model* is a set of tracks rigidly moving together, along with the corresponding structure and motion parameters. The algorithm outputs M after it has gone through all time stamps. It proceeds as follows:

(1) Initialize M to the empty set. Initialize T to all tracks found in the image sequence.
(2) For $t = 1$ to k do:
—(a) Run the seeding procedure (described later in this section) on the subset of T visible at time t. For each new model found, add it to M, remove its associated tracks from T, and estimate its structure and motion parameters.
—(b) Find the elements of T most consistent with each element M_i of M. If the reprojection error for a given element T_j relative to M_i is below some threshold, then add T_j to M_i and remove it from T.
—(c) Update the structure and motion parameters associated with the elements of M.
(3) Output M.

The seeding procedure used to initialize new models follows the segmentation approach in [13]: Given a set of points (in our case tracks), select the largest consistent subset, remove it, and repeat until no more large consistent subsets may be found. To select a consistent subset, we use a modified form of the RANSAC algorithm [12]. It first *deterministically* grows small sets of tracks that are likely to be rigidly connected (analogous to the random sampling part of RANSAC), and then finds all other compatible tracks (exactly like the consensus part of RANSAC). It relies heavily on the fact that, as noted in the previous section, two overlapping tracks of patches rigidly moving together generally provide an over-constrained system of equations on the corresponding structure and motion parameters, so the consistency of a pair of tracks

[1] See [22] for related work in the purely projective structure-from-motion domain.

can be assessed by measuring the reprojection error. In addition, the motion parameters can be used to assess the consistency of other tracks. The algorithm proceeds as follows, starting with an empty M.

Grow seeds:
(1) $S \leftarrow \{\}$.
(2) For each track T_i in T, generate a new seed S_i.
(3) For each seed S_i in S, do:
—(a) Select the track in T most consistent with S_i.
—(b) If the reprojection error is below E_1, add the track to S_i, reestimate the model, and go to (a).
Form consensus:
(4) For each sufficiently large seed, find its consensus set, i.e., the tracks whose reprojection errors are below a second threshold E_2, and add it to the seed.
(5) If the largest seed found is larger than some threshold, reestimate the corresponding structure and motion parameters, add it to M (permanently removing its components from the set of unassigned tracks), and go to (1).
(6) Output M.

In practice, we use $E_1 = 0.08$pixel, and $E_2 = 1$pixel. The reason for picking such a small value for E_1 is to ensure that the corresponding structure and motion parameters are estimated precisely enough to bootstrap the modeling process. The sub-pixel tracking method used in our experiments (see Section 5) allows us to find numerous seeds with errors below this value. Note that the structure and motion parameters are kept fixed during the consensus formation phase (step 4) of our algorithm, as is customary in RANSAC. On the other hand, they are re-estimated at each stage of seed formation (step 3 of the algorithm) to increase the reliability of that stage at relatively low cost.

Given a set of tracks and an initial estimate of the structure and motion parameters, it is easy to update the parameters as new tracks are added. The iterative process described in Section 2.3 directly supports this, since it treats the patch and projection matrices independently. However, more work is required to form the estimate of structure and motion for the first time, because it must handle missing data (i.e., places where some images lack measurements for some patches). One solution is find full blocks (subsets of the images and patches such that all patches appear in all images), factorize each block, and then register the resulting sub-models into a single global model. Full blocks can be found efficiently using an interval graph algorithm similar to those described in [15].

Once an affine or locally affine model is available, it is a simple matter to compute (if necessary) the corresponding Euclidean model when some of the camera intrinsic parameters are known using one of the self-calibration techniques available in that case for affine or projective cameras [26, 27, 32].

4. Model Matching

We now assume that the technique described in the previous section has been used to create a number of 3D models for the rigid components of a scene observed in a video, and address the problem of matching the models associated with different video clips, or with different shots within a single video. The approach we have chosen is essentially a 3D affine or projective alignment method (see [11, 18, 21] for related work), aided by the facts that (1) the normalized appearance of affine-invariant patches can be used to select promising matches between tracks, and (2) a pair of matches is sufficient to estimate the affine or projective alignment transformation. A match is a pair of 3D patches, one in each model. Each 3D patch contains three points (its center and two corners), so each match provides three point matches, and a pair of matches provides six point matches. The algorithm is an application of the same modified RANSAC algorithm used in the seeding procedure described earlier, because the goal is similar: find the largest subset of consistent matches.

(1) Build a set T of potential matches by pairing each patch in the first model to the k closest patches in the second model, based on similarity of appearance.
RANSAC-like selection of consistent matches:
(2) For each match T_i in T, generate a new seed S_i.
(3) For each seed S_i, do:
—(a) Find the match T_j not in S_i, that, when used jointly with the matches in S_i, minimizes the alignment error.
—(b) If the error is below some threshold E_1, add the match to S_i, and goto (a).
(4) For each sufficiently large seed, find its consensus set. Specifically, estimate the corresponding alignment transformation and select all matches whose error is below a second threshold E_2.
(5) Retain the seed with the largest consensus set and recompute the aligning transformation.
Expand the set of matches:
(6) Add more matches to the final result using the aligning transformation as a guide. New matches must have error below E_2.

A notable difference between this algorithm and the seeding procedure used in segmentation and modeling is that candidate matches to be added to the support set are used to compute the alignment parameters *before* computing the corresponding error. This is because the projective alignment estimates obtained from only two matches have proved unreliable in our experiments. The threshold E_1 is chosen as the average of the alignment errors found by exhaustively checking all pairs of matches in T. The threshold E_2 is chosen as half the average distance between patch centers in the first model.

The job of the "expansion" phase is to find all possible

true matches between the two models, including those that may have been missed by comparing patch appearance. The number of true matches provides a similarity measure between the models. Specifically, we use the repeat rate $M/\min(A, B)$, where M is the number of matches, A is the number of patches in one model, and B is the number of patches in the other.

5. Implementation and Results

We have implemented the proposed approach, and this section presents preliminary modeling and matching experiments. Affine-invariant patches are found and tracked using a variant of the Kanade-Lucas-Tomasi (KLT) feature tracker [32], tuned to track affine-invariant patches with sub-pixel localization. Concretely, we have augmented Birchfield's implementation [6] of the KLT tracker as follows: For each new frame i, we find points in the image that aren't currently being tracked and determine their patch parameters using the affine-adaptation process described in [23], providing an initial value for the matrix $\mathcal{S}_{ij}$ associated with each patch j. For all patches that are currently being tracked (i.e., that exist in frame $i-1$), we use the KLT tracker to update the location of the patch center in frame i, and then use non-linear least squares to refine the parameters of the patch, maximizing the normalized correlation between the patch in frame i and the same patch in the frame where it first appeared. In addition to the criteria that KLT itself uses to stop tracking a point, we also check whether the ratio of the dimensions of the patch exceed some threshold, and whether the correlation with the initial patch falls below some threshold (currently 0.95). It takes an average of 30 seconds to process one frame of video. In practice, this technique gives excellent results, yielding very robust tracking results as well as sub-pixel localization, which is crucial for the reliability of the multi-view constraints used in segmentation and modeling.

We have conducted preliminary experiments using both laboratory videos and shots from the film "Run Lola Run", and Figures 2 and 3 below show some results. Representative videos are also available on our group web site.

Figure 2 show the results of our laboratory experiments using videos of stuffed animals. The top two rows show the result of a segmentation experiment using a bear and a dog rotating independently, but with similar speeds and axes of rotation. The segmentation program returns three models, one for the bear, and two for the dog. Essentially, it is unable to continue growing the first dog model when it presents its narrow side to the camera, and we obtain one model for each side of the dog. Slight over-segmentation due to rapid image changes, occlusion, or other factors is a recurring problem in our experiments, but it does not necessarily hamper applications such as shot matching since multiple models can in principle be matched independently. Representative frames of the video are shown in the figure, along with the corresponding patches and reprojections of the estimated models, surrounded by a black frame. The third row of Figure 2 shows a second segmentation experiment, where the head of the bear is moved by hand independently from its body. The head is found as one segment, and the body is (over) segmented into three components. The fourth row of the figure shows the bear models constructed from the bear-and-dog video and from a different video of the bear alone, along with the recovered cameras. The last row of the figure shows the result of matching. The model on the right has been aligned with the model on the left, and a red "+" marks the origin of the left model's coordinate system. It took about 210 seconds to match these two models.

Figure 3 shows similar results for several shots from the "Lola" video. The first two rows show two frames of a shot, along with the corresponding patches and reprojected models. Here, Lola runs around a street corner while a car drives down the street and toward the camera. The car is segmented correctly, while the rest of the street, which is static, is, as usual, slightly over-segmented. Note that Lola is not found as a part of any model. This is typical for non-rigid objects, since the motion segmentation algorithm is based on rigidity constraints. (We conjecture that such objects can be found by first subtracting all rigid components from the scene and then using a 2D motion coherence constraint, but we have not had a chance to experiment with this idea yet.) Also note that although the camera remains still in that particular part of the shot, it moves in earlier parts of the shot used to construct the street model shown in the figure. The third row shows the results of a matching experiment between the street model and another model computed from a different shot. The repeat rate is 17% in this case. The fourth row of the figure shows another segmentation experiment. In this case, the train is found as two distinct models by our program (only one of them is shown in the figure), and the static scene is found as 12 models. The camera is in constant and moderately complex motion. The last row shows the results of a matching experiment between models from the train scene taken from two different shots, with a repeat rate of 46%.

Although the segmentation, modeling, and matching results presented in this section are preliminary, we believe that they demonstrate the promise of the proposed approach with challenging datasets that include complex camera motions, multiple moving objects, large perspective distortions, and patches only visible in small parts of each shot. More experimental work is of course needed.

Acknowledgments. This research was supported in part by the National Science Foundation under grants IIS-0308087 and IIS-0312438, by the UIUC Campus Research Board and by the CNRS-UIUC Research Collaboration Agreements.

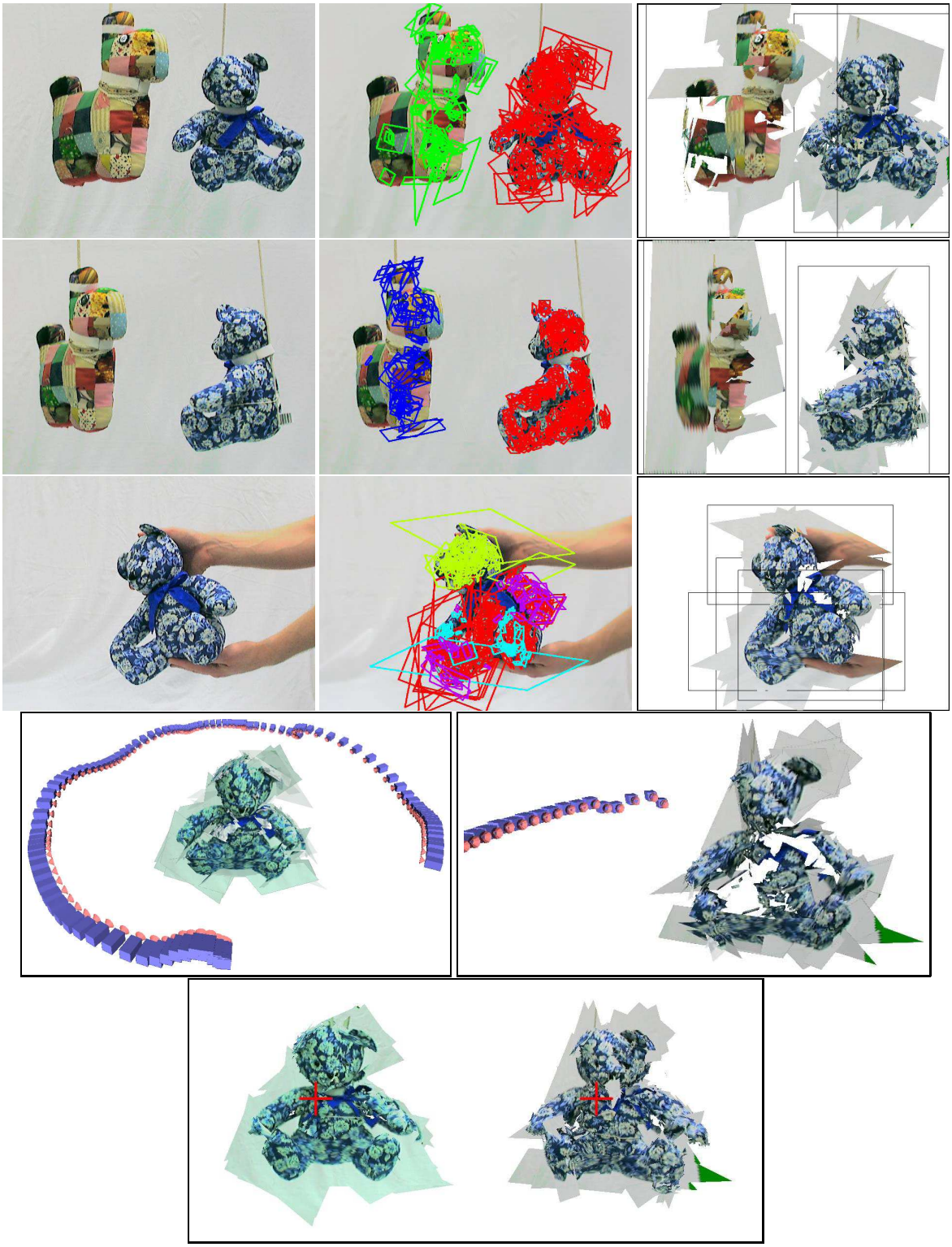

Figure 2: Laboratory experiments using an affi ne projection model. See text for details.

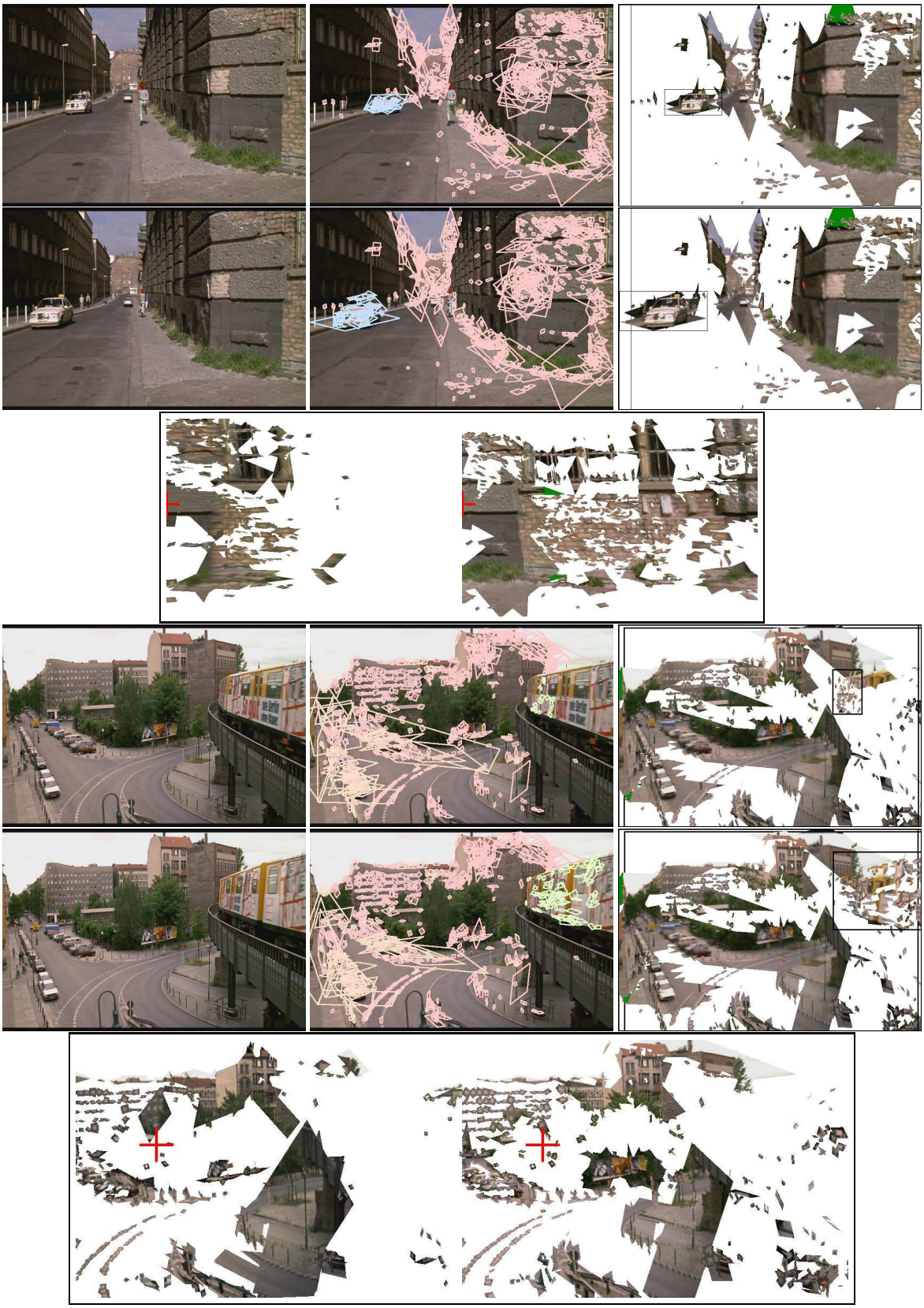

Figure 3: Experiments with shots from "Run Lola Run" using the locally affine projection model. See text for details.

References

[1] A. Aner and J. R. Kender. Video summaries through mosaic-based shot and scene clustering. In *Proc. ECCV*, 2002.

[2] M. Ardebilian, X. W. TU, L. Chen, and P. Faudemay. Video segmentation using 3D hints contained in 2D images. *SPIE* 2916, 1996.

[3] A. Baumberg. Reliable feature matching across widely separated views. In *Proc. CVPR*, 2000.

[4] S. Belongie, J. Malik, and J. Puzicha. Matching shapes. In *Proc. ICCV*, 2001.

[5] S. Benayoun, H. Bernard, P. Bertolino, M. Gelgon, C. Schmid, and F. Spindler. Structuration de vidéos pour des interfaces de consultation avancées. In *Proc. CORESA*, 1998.

[6] S. Birchfield. KLT: An implementation of the Kanade-Lucas-Tomasi feature tracker, 1998.

[7] C. Bregler and J. Malik. Tracking people with twists and exponential maps. In *Proc. CVPR*, 1998.

[8] J. B. Burns, R. S. Weiss, and E. M. Riseman. View variation of point-set and line-segment features. *PAMI*, 15(1), 1993.

[9] D.E. DiFranco, T.-J. Cham, and J.M. Rehg. Recovery of 3D articulated motion from 2D correspondences. Tech. Rep. CRL 99/7, Compaq Cambridge Res. Lab., 1999.

[10] O. Faugeras, Q.-T. Luong, and T. Papadopoulo. *The Geometry of Multiple Images*. MIT Press, 2001.

[11] O.D. Faugeras and M. Hebert. The representation, recognition, and locating of 3-D objects. *IJRR*, 5(3), 1986.

[12] M. Fischler and R. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *CACM*, 24(6), 1981.

[13] A. Fitzgibbon and A. Zisserman. Multibody Structure and Motion: 3-D Reconstruction of Independently Moving objects. In *Proc. European Conf. Comp. Vision*, pages 891–906, June 2000.

[14] U. Gargi, R. Kasturi, and S.H. Strayer. Performance characterization of video-shot-change detection methods. *IEEE Trans. Circuits and Systems for Video Technology*, 10(1), 2000.

[15] U.I. Gupta, D.T. Lee, and Y.Y.-T. Leung. Efficient algorithms for interval graphs and circular-arc graphs. *Networks*, 12, 1982.

[16] C. Harris and M. Stephens. A combined edge and corner detector. In *Alvey Vis. Conf.*, 1988.

[17] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge Univ. Press, 2000.

[18] A.E. Johnson and M. Hebert. Surface matching for object recognition in complex three-dimensional scenes. *IVC*, 16, 1998.

[19] R. Lienhart. Reliable transition detection in videos: A survey and practitioner's guide. Tech. Rep., Intel MRL, 2002.

[20] T. Lindeberg and J. Gårding. Shape-adapted smoothing in estimation of 3-D shape cues from affine deformations of local 2-D brightness structure. *IVC*, 15(6), 1997.

[21] D. Lowe, Distinctive image features from scale-invariant keypoints. *IJCV*, 2003. In press.

[22] S. Mahamud, M. Hebert, Y. Omori, and J. Ponce. Provably-Convergent Iterative Methods for Projective Structure from Motion. In *Proc. CVPR*, pages 1018–1025, 2001

[23] K. Mikolajczyk and C. Schmid. An affine invariant interest point detector. In *Proc. ECCV*, 2002.

[24] J.L. Mundy and A. Zisserman. *Geometric Invariance in Computer Vision*. MIT Press, 1992.

[25] J.L. Mundy, A. Zisserman, and D. Forsyth. *Applications of Invariance in Computer Vision*, LNCS 825, Springer-Verlag, 1994.

[26] C.J. Poelman and T. Kanade. A paraperspective factorization method for shape and motion recovery. *PAMI*, 19(3), 1997.

[27] M. Pollefeys, R. Koch, and L. Van Gool. Self-calibration and metric reconstruction in spite of varying and unknown internal camera parameters. In *Proc. ICCV*, 1998.

[28] P. Pritchett and A. Zisserman. Wide baseline stereo matching. In *Proc. ICCV*, 1998.

[29] F. Rothganger, S. Lazebnik, C. Schmid, and J. Ponce. 3D Object Modeling and Recognition Using Affine-Invariant Patches and Multi-View Spatial Constraints. In *Proc. CVPR*, 2003.

[30] F. Schaffalitzky and A. Zisserman. Automated scene matching in movies. In *Challenges of Image and Video Retrieval*, 2002.

[31] J. Sivic and A. Zisserman. Video Google: a text retrieval approach to object matching in videos. In *Proc. ICCV*, 2003.

[32] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. *IJCV*, 9(2), 1992.

[33] T. Tuytelaars and L. Van Gool. Matching widely separated views based on affinely invariant neighborhoods. *IJCV*, 2003. In press.

[34] M.M. Yeung and B. Liu. Efficient matching and clustering of video shots. In *Proc. ICIP*, 1995.

[35] Z. Zhang. Token tracking in a cluttered scene. *IJCV*, 12(2), 1994.