Transitivity-based Removal of Correspondence Outliers for Motion Analysis

Antonio Robles-Kelly [1] and Yael Moses* [2]
[1] National ICT Australia (NICTA)†, RSISE Bldg. 115, ANU, Canberra, ACT 0200, Australia
[2] The School of Computer Science, The Interdisciplinary Center, Herzliya 46150, Israel
[1]Antonio.Robles-Kelly@nicta.com.au [2]yael@idc.ac.il

Abstract

In this paper we address the problem of removing correspondence outliers in a sequence of images. The input to the system is a set of putative matches which are based upon image-feature similarity. Classical methods for outlier removal, such as RANSAC-based approaches, assume consistency and rigidity in the scene motion between two or three frames in the sequence. Here we propose a novel method for removing correspondence outliers that does not depend on such assumptions. Our method is based on the observation that correspondence is an equivalence relation, and, hence, transitivity must hold between corresponding features in different frames. We impose consistency on the transitivity by representing the matching information as a weighted graph with positive and negative edge-weights. Consistency is then enforced by partitioning the nodes in the graph so as to remove edges corresponding to false-positive correspondences. The clustering algorithm is of spectral nature and can handle graphs whose edge-weights are non-positive. Our method is a general one that can be used for purposes of outlier removal from correspondences between any entities whose putative matches imply equivalence relations. We illustrate the utility of the method for purposes of outlier removal on a real-world image sequence and compare our results with those yield using an alternative.

1. Introduction

This paper proposes a new method of removing outliers for purposes of motion analysis and tracking. The task of matching point patterns between two or more images is of central importance for a number of problems in image analysis and pattern recognition. A matching algorithm should function well in the presence of image noise, geometric distortion and structural corruption. A large percentage of false positive matches results in unreliable data for computations such as shape analysis [5], motion analysis [18], motion segmentation, and stereo reconstruction [15].

As a result, a substantial effort in the literature has focused on removing outliers from a set of matching features. Furthermore, computational theories on the perception of motion [20, 1] have been proposed so as to capture global feature coherence on local measurements. In this paper, we present an outlier removal method for motion analysis that imposes consistency constraints in the temporal domain without assuming any motion or scene model. This contrasts with other methods elsewhere in the literature for removing matching errors, which are based on assumptions on the scene motion or shape e.g., rigid motion or a predetermined model [8, 9]. In these approaches, corresponding pairs of features are evaluated and outliers are removed based on the consistency of corresponding features across the image with the predefined model of the motion or the scene. Often these assumptions do not hold, and therefore the quality of the matching results is hindered.

To develop our outlier removal method, we depart from the work in [6] and [2] by viewing the set of putative matches between features in a pair of frames as an equivalence relation. Consistency is imposed on the transitivity for the feature correspondences between a number of consecutive frames on image sequence, i.e. a time-window over the frame-set under study. Following this treatment, the correspondence outlier removal problem can be viewed as a grouping of features in different frames so as to enforce consistency in the transitivity information. We cast the outlier removal problem in a graph-theoretic setting. Further, our algorithm removes outliers from relatively noisy correspondence data without making any simplifying assumptions regarding image motion or the geometry of the scene. In addition, it is not limited to a given number of views, i.e. 2 or 3, which allows us to impose consistency on the inliers making use of more correspondence information than competing approaches. Thus, the method presented here can be used for complex motion analysis and tracking.

It is worth noting that our method is quite general and can be used for purposes of outlier removal from correspon-

*Yael Moses was partially supported by National ICT Australia.

†National ICT Australia is funded by the Australian Governments Backing Australia's Ability initiative, in part through the Australian Research Council.

dences between any entities whose putative matches imply equivalence relations. Examples of higher level entities include correspondence between objects (identify the same objects in different frames), and correspondence between events (matching events in different sequences based on a basic event matching).

1.1. Previous work

As mentioned earlier, the correspondence problem has attracted wide attention in the computer vision and pattern recognition communities. Hence, the literature on robust correspondence matching is vast. Statistics have been used by several authors to overcome measurement noise and contamination [18, 16]. Rangarajan *et al.*[13] have used softassign and Procrustes alignment to perform matching. Chui and Rangarajan [4] use a thin plate model and softassing to perform non-rigid point matching. Adopting a two stage process, Wills and Belongie [19] have used a planar model and a spline fit to achieve results robust under large levels of disparity between feature points.

The studies concerning the removal of outliers closest to our approach are given in [6] and [2]. These methods use transitivity as a relatively simple local heuristic manner for correcting errors in a Wide Baseline, multiple view correspondence. [2] also gave a theoretical analysis on the detectability of matching errors, both false positive and false negative, based solely on transitivity. The current work generalises, from a theoretical viewpoint, the developments in [6] and [2] by using a weighted graph rather than a binary one, and using the complete information from the matching graph rather than local binary relations.

Here, we enforce consistency making use of a graph cut which aims at removing those inconsistencies detected through the transitive closure without introducing false positives. Once the conflicting matches are detected making use of transitivity rules, we cut the graph based upon confidence values governed by the matches at hand. Thus, we model the problem of enforcing consistency as a pairwise clustering one.

Indeed, many problems in computer vision can be posed as ones of pairwise clustering. One of the most elegant solutions to the pairwise clustering problem comes from spectral graph theory, i.e. the characterisation of the eigenpairs of the graph Laplacian and the adjacency matrix [17, 12, 14]. We use a graph-spectral method to enforce consistency over the matches. Our clustering algorithm is such that it employs an eigenvalue problem to maximise a cost function defined in terms of the correspondence information between image features. What distinguishes it from other approaches in the literature is the fact that we use positive and negative edge weights to model the correspondences between features and, hence, our method is capable of partitioning weighted graphs whose weights are not necessarily non-negative.

2. The method

The goal of our method is to remove outliers from the set of feature correspondences between images in a video sequence. The input is the result of a matching algorithm between image features in a set of N successive frames. We assume that a similarity measure can be assigned to each pair of image features by the matching algorithm. The proposed method detects and removes outliers based only on the consistency of correspondence information among the set of N frames under study. A consistency constraint follows from the observation that correspondence is an equivalence relation, and is therefore transitive. We present the matching information in the form of a weighted graph with positive and negative weights. Following this treatment, the consistency constraint can be enforced making use of a score function, maximised by an algorithm for clustering nodes in the graph.

2.1. Graph-theoretic Setting

We represent the correspondence information computed by the matching algorithm between the N frames under study as an undirected graph $\mathcal{G} = (V, E)$. The set of nodes V stands for the set of detected image features in the N frames. The set of edges is defined by the putative matches. Formally, let the i^{th} frame of the image sequence be I_i. For each frame I_i, we denote the set of n_i image feature vectors as $\Omega_i = \{\omega_i(1), \omega_i(2), \ldots, \omega_i(n_i)\}$, where $\omega_i(k)$ is the k^{th} image feature in the frame I_i. An edge connects the vertices $\omega_i(\ell)$ and $\omega_j(k)$ if a putative match was found between the feature points $\omega_i(\ell)$ and $\omega_j(k)$ when images I_i and I_j were compared. In this case, $(\omega_i(\ell), \omega_j(k)) \in E$. Also let $f : V \times V \rightarrow [0, 1]$ be the similarity function between image features.

Consider the connected component $\mathcal{R}$ of a given image feature $\omega_i(k)$, corresponding to the node $v \in V$ in $\mathcal{G}$. Clearly, $\mathcal{R} = (V_R, E_R)$ is a subgraph of $\mathcal{G}$. The graph $\mathcal{R}$ consists of all feature points that are detected either directly or transitively as corresponding to v. Ideally, the fact that correspondence is an equivalence relation should imply that each connected component of $\mathcal{G}$ is a clique. Indeed, this is the case so long as the matching process does not produce false-positive and false-negative errors. False-negative errors result in missing clique edges in $\mathcal{R}$. False-positive errors may connect subgraphs of $\mathcal{G}$ that represent different features. The graph $\mathcal{R}$ can therefore contain non-corresponding features that are mistakenly considered to be corresponding due to a false positive error. In this case, missing edges among nodes of $\mathcal{R}$ may be correct and represent the fact that the two points do not correspond, despite being transitively matched. In general, when $\mathcal{R}$ is not a clique it is not always possible to determine whether this is due to false-positive or false-negative errors of the matching process. However, unequivocal evidence for the existence of a false-positive error can be obtained if the graph contains a path between two vertices that represents two points in the same image. We name this case a *local conflict*.

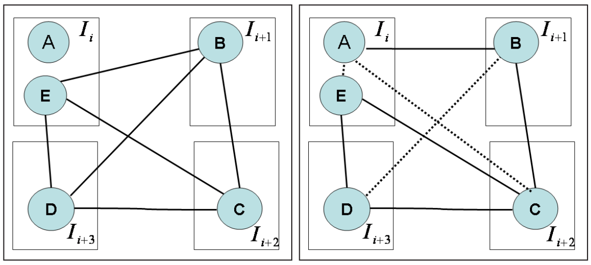
A local conflict indicates an inconsistency in the putative matches that cannot be tolerated and must be resolved. To see this more clearly, consider the case illustrated in Figure 1**a**, where we show four frames in I. The edges of $\mathcal{R}$, i.e. the putative correspondences between image features, are denoted by the solid lines between frames. The dash-dotted lines correspond to the complement of $\mathcal{R}$, that is, the matches recovered via the transitive closure. Note that, in Figure 1**b**, through the transitive closure, we can conclude that the two features A and E in frame I_i are transitively matched to one another. This cannot be true since an image feature cannot match to any other feature in the same frame. Thus, we can locate conflicting matching results making use of the transitive closure. Once these conflicts have been located, they can be removed by enforcing consistency making use of graph-cuts.

An edge between a pair of vertices in $\mathcal{R}$ supports the existence of correspondence between these two vertices, while a missing edge between the two vertices can be regarded as a negative indication for correspondence between them. Furthermore, a weight can be assigned to the negative and positive support of correspondence between two feature points based on the similarity between them (obtained via the similarity function f mentioned above and defined in Equation 8). We represent the positive and negative support for the correspondence between the nodes of $\mathcal{R}$ by a weighted graph $G = (V_G, E_G, W_G)$ which is the transitive closure of $\mathcal{R}$. Positive weights are assigned to the original edges of $\mathcal{R}$, whereas negative weights are assigned to the edges in $\mathcal{R}$'s complement. The edge weights indicate the confidence that the two features points correspond. A positive weight supports correspondence. A negative weight suggests that the points do not correspond. Formally, the weight of an edge is defined based on $\mathcal{R}$ and the similarity function f as follows:

$$w(k,\ell) = \begin{cases} f(k,\ell) & \text{if } (k,\ell) \in E_R \\ -1 + f(k,\ell) & \text{otherwise} \end{cases} \tag{1}$$

2.2. Detecting and correcting failures

The formulation above implies that, if the set of negative weights in G is nonempty, then the graph represents inconsistent correspondences which must be corrected. Our goal is to modify the initial putative set of matches so as to obtain a consistent set of pairwise correspondences. Adding edges to $\mathcal{R}$ corresponds to correcting false negative matches. Removing edges from $\mathcal{R}$ corresponds to removing false positive edges. Adding and removing edges from $\mathcal{R}$ corresponds to changing positive to negative, and negative to positive weights of G-edges, respectively.



(a) (b)

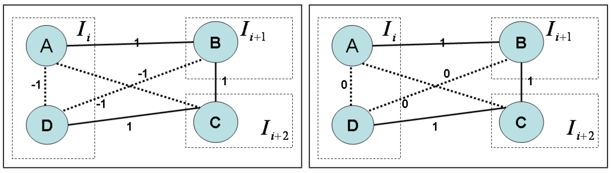
Figure 1. Matching graphs. (a): when only a false negative error occurs between features B and D. (b): with false positive between features A and B. A local conflict occurs in frame I_i. The solid lines are the putative matches. The matches yield by the transitive closure have been drawn using dash-dotted lines.

There are many possibilities to add or remove positive-weight matches in order to impose consistency. One possibility is to remove positive-weighted edges so that each connected component does not contain negative edges. This solution clearly gives a strong bias towards edges with negative weights. Another example is where edges with negative weights are removed by replacing them with positive-weight edges until no negative edges remain. This clearly gives a strong bias to negative-weighted edges. Furthermore, it cannot be applied when local conflict occurs.

When a local conflict occurs in G, the graph must be partitioned into at least two disjoint subgraphs such that the local conflict is resolved. Formally, let $\omega_i(k)$ and $\omega_i(\ell)$ be two vertices of G which correspond to two image-features in the same image. Then, G is partitioned into $G_1 = (V_1, E_1, W|_{E_1})$ and $G_2 = (V_2, E_2, W|_{E_2})$ such that if $\omega_i(k), \omega_i(\ell) \in V_G$ then the partition must satisfy that $\omega_i(k) \in V_1$ and $\omega_i(\ell) \in V_2$.

To choose the partition, we consider the global information about positive and negative weights in the graph. We next define a score function for partitioning the graph G into two subgraphs G_1, and G_2. The score function should reflect the consistency of each of these graphs and the cost of separating the vertices in G_1 from those in G_2. That is, within each cluster we would like to have as many positive weighted edges as possible and the least number of negative weighted edges. Similarly, between clusters we would like to have as few positive weighted edges as possible and as many negative weighted edges as possible. This is equivalent to maximising the positive and negative edge weights within each cluster, and minimising negative and positive edge-weight mass between clusters. Different scores may correlate with different assumptions on the type of errors expected.

Let $\Psi = \{\psi_1, \psi_2\}$ be a partition of V_G. Given a cost function, we can compute the clustering that maximises



(a) (b)

Figure 2. (a): Example of a connected component with negative edge-weights; (b): Example with non-negative edge-weights.

the score and obtain the set of correspondences that are both, free of conflicts and optimise over the conditions stated above. Let the sum of positive edge weights between the two clusters ψ_1 and ψ_2 in the original graph G be $s_P(\psi_1, \psi_2)$. We denote the sum of negative edge weights between the clusters $s_N(\psi_1, \psi_2)$. Similarly, the sum of inter-cluster positive and negative edge weights for the h^{th} cluster, i.e. $h = \{1, 2\}$, is given by $s_P(\psi_h)$ and $s_N(\psi_h)$, respectively. With these ingredients, the cost function we aim to maximise becomes

$$\begin{aligned} s(\Psi) &= \sum_{h=\{1,2\}} (s_P(\psi_h) + s_N(\psi_h)) \\ &\quad -(s_P(\psi_1, \psi_2) + s_N(\psi_1, \psi_2)) \\ &= \sum_{h=\{1,2\}} \sum_{v_i, v_j \in \psi_h} w(v_i, v_j) \\ &\quad - \sum_{v_i \in \psi_1, v_j \in \psi_2} w(v_i, v_j) \end{aligned} \tag{2}$$

Before we describe the clustering algorithm that maximises the cost function $s(\Psi)$, let us first illustrate the role of the negative edge-weights. Consider two connecetd component graphs which result from the same putative matches. The graph in Figure 2**a** contains negative edge weights and that of Figure 2**b** does not. The only difference between the two graphs is in the edge-weights of the edges connecting unmatched points: on the right-hand graph such edge-weights are set to zero while on the the left-hand graph it is set to -1.

Consider two different partitions that seperate the two vertices that correspond to features in the same image, A and D. The first consists of two vertices each, $\phi_1 = \{A, B\}$ and $\phi_2 = \{C, D\}$; The second consists of a single vertex in one cluster and three vertices in the other, e.g., $\phi_1 = \{A\}$ and $\phi_2 = \{B, C, D\}$; Note that, for the case in which the negative weights are present, the score function $s(\Psi)$ clearly favours the partition of the clique such that $\phi_1 = \{A, B\}$ and $\phi_2 = \{C, D\}$ (the score is 4 compared to 1). When no negative weights are present, the score function yields the same value, 1, for the two partitions. It is, hence, clear that the inclusion of negative edge-weights favours those partitions whose end result is to “cut” through the edges whose weights are negative.

2.3. Clustering

In this section, we describe an algorithm for partitioning the graph so as to maximise the cost function $s(\Psi)$ in Eq. 2. If $|V_G|$ is small, an exhaustive search method can be used to recover the optimal partition.

In practice, we would like to use a large number of image frames to impose consistency constraints upon the correspondences recovered by the relational matching algorithm. We solve this problem by adopting a graph-spectral partitioning approach. To do so, the cost function $s(\Psi)$ is first expressed in a matrix form. Let $\mathcal{R}$ be the binary putative matching matrix, that is the adjacency matrix of the graph $\mathcal{R}$, let F be a similarity matrix that consists of the similarity measure defined by the function f between all vertices of G, and let W be the weight matrix which consists of the weight, defined in Equation 1 between a pair of nodes in G. Consider the cluster membership variables $x = [x(1), x(2), \ldots, x(|V|)]^T$ such that $x(i) \in \{-1, 1\}$. The nodes of G are assigned to either ψ_1 or ψ_2 based upon the sign of the cluster membership variables so as to satisfy $\psi_1 = \{v(k) \mid x(k) = 1\}$ and $\psi_2 = \{v(k) \mid x(k) = -1\}$.

We commence by rewriting the cost function $s(\Psi)$ in matrix form making use of the fact that $x(k)x(\ell) = 1$ if the two corresponding nodes are within the same cluster and $x(k)x(\ell) = -1$ when they are at different clusters. Moreover, relaxing the constrains on the elements of x so as to be in the range $[-1, 1]$ results in the weighted version of $s(\Psi)$, which is given by

$$\hat{s}(\Psi) = x^T W x \tag{3}$$

Since W contains negative values, methods elsewhere in the literature, such as that in [17], cannot be employed to solve the grouping problem at hand. As a result, we must take our analysis further. We note that the relaxation of the values for the components $x(k) \in x$ turns the optimisation problem under study into an under-constrained one. Hence, we subject the maximisation problem to the constraint

$$\sum_{k=1}^{|V|} x(k) = 0 \tag{4}$$

on the components of the vector x.

Note that from the definition of the weight of the matrix G in Eq. 1, it follows that $W = F + \mathcal{R} + \mathbf{1} - I$ where $\mathbf{1}$ is all ones matrix. Therefore, Eq. 3 can be rewritten, as follows

$$\hat{s}(\Psi) = x^T(\mathcal{R} + F + \mathbf{I})x - x^T \mathbf{1} x \tag{5}$$

The algebraic manipulation of the score function $\hat{s}(\Psi)$ above is important due to the fact that it opens up the possibility of casting the maximisation of the score function $\hat{s}(\Psi)$ as an eigenvalue problem. To do this, we note that, as a consequence of Constraint 4, we have that $x^T \mathbf{1} x = 0$. Thus, Equation 5 becomes

$$\begin{aligned} \hat{s}(\Psi) &= x^T H x + x^T x \\ &= (D^{-\frac{1}{2}}\phi)^T H D^{-\frac{1}{2}}\phi + (D^{-\frac{1}{2}}\phi)^T D^{-\frac{1}{2}}\phi \end{aligned} \quad (6)$$

where we have used the shorthands $x = D^{-\frac{1}{2}}\phi$ and $H = \mathcal{R} + F$ and D is a diagonal matrix such that $D = \text{diag}(\text{deg}(1), \text{deg}(2), \ldots, \text{deg}(|V|))$ and $\text{deg}(k) = \sum_{\ell=1}^{\ell=|V|} H(k,\ell)$ is the k^{th} row-degree of H. It is worth noting that H is a positive definite, symmetric matrix and, therefore, the degree matrix D is always invertible.

From inspection, it is straightforward to note that we can express the cost function $\hat{s}(\Psi)$ in terms of the normalised Laplacian $\mathcal{L}$ of H. Hence, we write

$$\begin{aligned} \hat{s}(\Psi) &= \phi D^{-1}\phi - \phi^T\phi - \phi^T D^{-\frac{1}{2}}(D - H)D^{-\frac{1}{2}}\phi \\ &= \phi^T(D^{-1} - \mathbf{I} - \mathcal{L})\phi = \phi^T Q \phi \end{aligned}$$

where $Q = D^{-1} - \mathbf{I} - \mathcal{L}$. By imposing a unit-norm constraint on the vectors ϕ, we can write

$$\hat{s}(\Psi) = \frac{\phi^T Q \phi}{\phi^T \phi} \quad (7)$$

which is a Rayleigh quotient. Thus, the weighted cost function $\hat{s}(\Psi)$ is maximised by the eigenvector ϕ corresponding to the largest eigenvalue of the matrix Q.

At this point, it is worth noting that, since the degree matrix D and the identity matrix $\mathbf{I}$ are both diagonal, ϕ is an eigenvector of both, the normalised Laplacian $\mathcal{L}$ and the matrix Q. Further, let the eigenvalues of the normalised Laplacian $\mathcal{L}$ be $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_{|V|} = 0$. Similarly, the eigenvalues of the matrix Q are given by $\zeta_1 \geq \zeta_2 \geq \ldots \geq \zeta_{|V|}$. The i^{th} eigenvalue ζ_i of the matrix Q can be computed, making use of ζ_1 and the eigenvalue of $\mathcal{L}$ indexed i, using the relation $\zeta_i = \zeta_1 - \lambda_{|V|+1-i}$. As a result, $\hat{s}(\Psi)$ is maximised by taking the eigenvector of $\mathcal{L}$ corresponding to the eigenvalue λ_i whose rank is smallest, i.e. $\lambda_{|V|}$. Unfortunately, the eigenvector corresponding to $\lambda_{|V|}$ does not satisfy Constraint 4 and, hence, for purposes of computing x and separating the vertices in G into two disjoint subgraphs, we use the eigenvector of the normalised Laplacian $\mathcal{L}$ corresponding to its second smallest eigenvalue, i.e. $\lambda_{|V|-1}$.

To summarise, the steps of the clustering algorithm are as follows:

1.- Compute the matrix $H = \mathcal{R} + F$, which corresponds to the connected component under consideration.

2.- Compute the normalised Laplacian $\mathcal{L}$ of H.

3.- Recover the eigenvector ϕ corresponding to the second smallest eigenvalue of $\mathcal{L}$ and partition the connected component based upon the sign of its coefficients.

The above steps are repeated recursively, for every connected component, i.e. graph G in the video sequence, until no further inconsistent matches can be found. It is worth noting that, in Step 3, we have used the vector ϕ to partition V, as an alternative to the vector of cluster membership variables x. This is possible due to the fact that $x = D^{-\frac{1}{2}}\phi$ and, since the degree matrix D is a diagonal matrix whose diagonal elements are all positive, the sign of the coefficients of x and ϕ is the same.

At this point, we note that, in our experiments, we compute the matrix F making use of the function

$$f(\omega_i(k), \omega_j(\ell)) = \exp\Big(-\kappa \big\langle \omega_i(k) - \omega_j(\ell), \omega_i(k) - \omega_j(\ell) \big\rangle \Big) \quad (8)$$

where κ is a constant.

We always commence by estimating all the pairwise correspondences between a number of consecutive frames in the image sequence (typically, we employ 7 frames). Once the set of pairwise correspondences is at hand, we recover all the connected components, i.e. those sets of image-feature vectors that can be transitively matched to one another, and process them independently using the step sequence above. This has a number of advantages. Firstly, this opens-up the possibility of parallelising the algorithm. Secondly, since the algorithm makes use of a local consistency check over the correspondences for the image features in the frames under study, no global optimisation solvers are required.

3. Experiments

In this section, we illustrate the utility of our algorithm for purposes of outlier removal. To do this, we make use of two sequences. The first consists of rigid motion, while the second consists of non-rigid motion.

Rigid motion sequence

The “INRIA Syntim” sequence is an augmented reality video sequence that depicts an indoors scene with synthetically generated penguins. In this case, the scene is rigid and the motion is induced by a simple translation of the camera. Despite its simplicity, the lack of complex motion makes this sequence ideal for a qualitative evaluation of the results provided by the algorithm. Here, we use Harris corners [7] as features and the KD-tree algorithm to recover the set of putative matches taken by our algorithm at input.

In the left-hand panel of Figure 3, we show an example frame from the image sequence. The white dots mark the centroids for the connected components consisting of the pairwise correspondences between Harris corners. To visualize our algorithm on a single connected component, an enlarged connected component of the putative matches is presented in the center panel (the white arrow on the left panel indicates its location on the original sequence). The connected component was computed from the features of five frames. In the plots, every colour is indicative of a feature originating from a different frame. For example, the

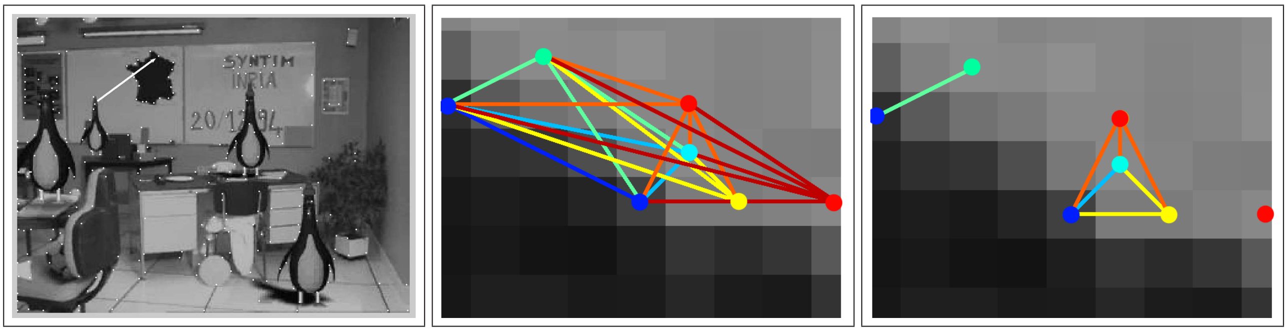


Figure 3. From left-to-right: Tenth frame of the INRIA Syntim sequence, detail of the raw correspondences and detail of the cliques delivered by our outlier removal algorithm.

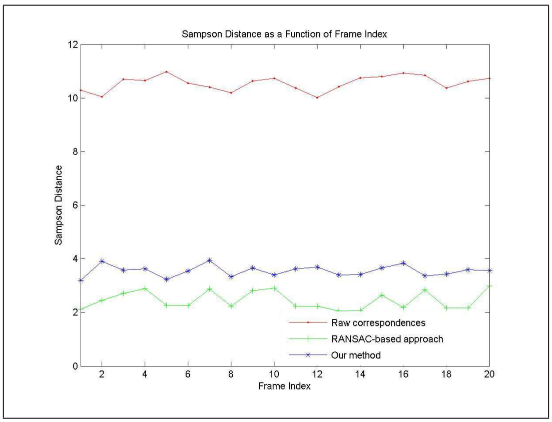


Figure 4. Sampson distance plots for the INRIA Syntim sequence.

blue dots indicate the position of the corners in the frame of reference, e.g., the tenth frame, whereas the red one shows the position of the corners at the fifteenth frame. It can easily be seen that the connected component in the center panel contains local conflicts. The right-hand panel consists of the result of applying our outlier removal algorithm. The original connected component in the center panel is partitioned into three graphs which are the result of two bipartitions of the original graph. Our algorithm has partitioned the connected component so as to isolate the outlier, which in this case is the red-coloured dot on the far-right side of the panels. The other two graphs correspond to two Harris-corner cliques which are consistent with the camera panning motion in the sequence.

To further evaluate our method, we provide a simple comparison of our results to those obtained using RANSAC[1] for a robust estimation of the fundamental matrix. Both methods are applied to the putative matches recovered by the KD-Tree. The RANSAC is applied to these putative matches and then used to estimate the fundamental matrix [9]. Once the fundamental matrix has been recovered, those matches that are outliers related to this estimate are discarded. In our implementation, the threshold for the inliers was 1.25 pixels and, for the estimation of the fundamental matrix, a Levenberg-Marquardt algorithm was used.

In Figure 4, we show the Sampson distance [9] between the features at the i^{th} frame and its point correspondences on the image plane at the frame $i{+}3$ for the first 20 frames of the image sequence. The Sampson distance can be viewed as an approximation to the first-order geometric error for the set of recovered correspondences. It employs the fundamental matrix as a means to measure the accordance of the recovered correspondences with the underlying epipolar geometry. Thus, for our algorithm, we estimate the fundamental matrix from the outlier-free correspondences using the maximum likelihood algorithm in [9]. For the alternative to our method, the computation is straightforward due to the fact that the recovery of the fundamental matrix is an integral part of the RANSAC-based outlier removal algorithm. Hence, for our method, the estimation of the fundamental matrix is a backward step, whereas for the RANSAC-based approach is a forward computation. From the plots, we note that the difference in performance between the two methods is not significative. This is important due to the fact that, despite the lack of constraints on the motion assumed by our method, the detriment in the outlier removal results is marginal.

Non-rigid motion sequence

The second sequence we studied is a real-world image sequence which presents a number of rigid motions, which combined make it largely non-rigid. This sequence, which we call “Vehicles from the Air”, captures two vehicles, moving at different speeds on a highway. The video has been captured from a helicopter and exhibits a large amount of jitter.

For this sequence, we have used three alternatives for the recovery of the image features used in the matching process. These are the Harris corners [7], Maximally Stable Extremal Regions (MSERs) [11] and Scale Invariant Feature Transform (SIFT) descriptors [10]. For the matching step, we have made use of the KD-Tree [3]. Once the matches are at hand, we impose consistency over the correspondences and remove outliers making use of our algorithm. Here, we have imposed consistency making use of 7

[1] We have used the model fitting and robust estimation routines in http://www.csse.uwa.edu.au/ pk/research/matlabfns/

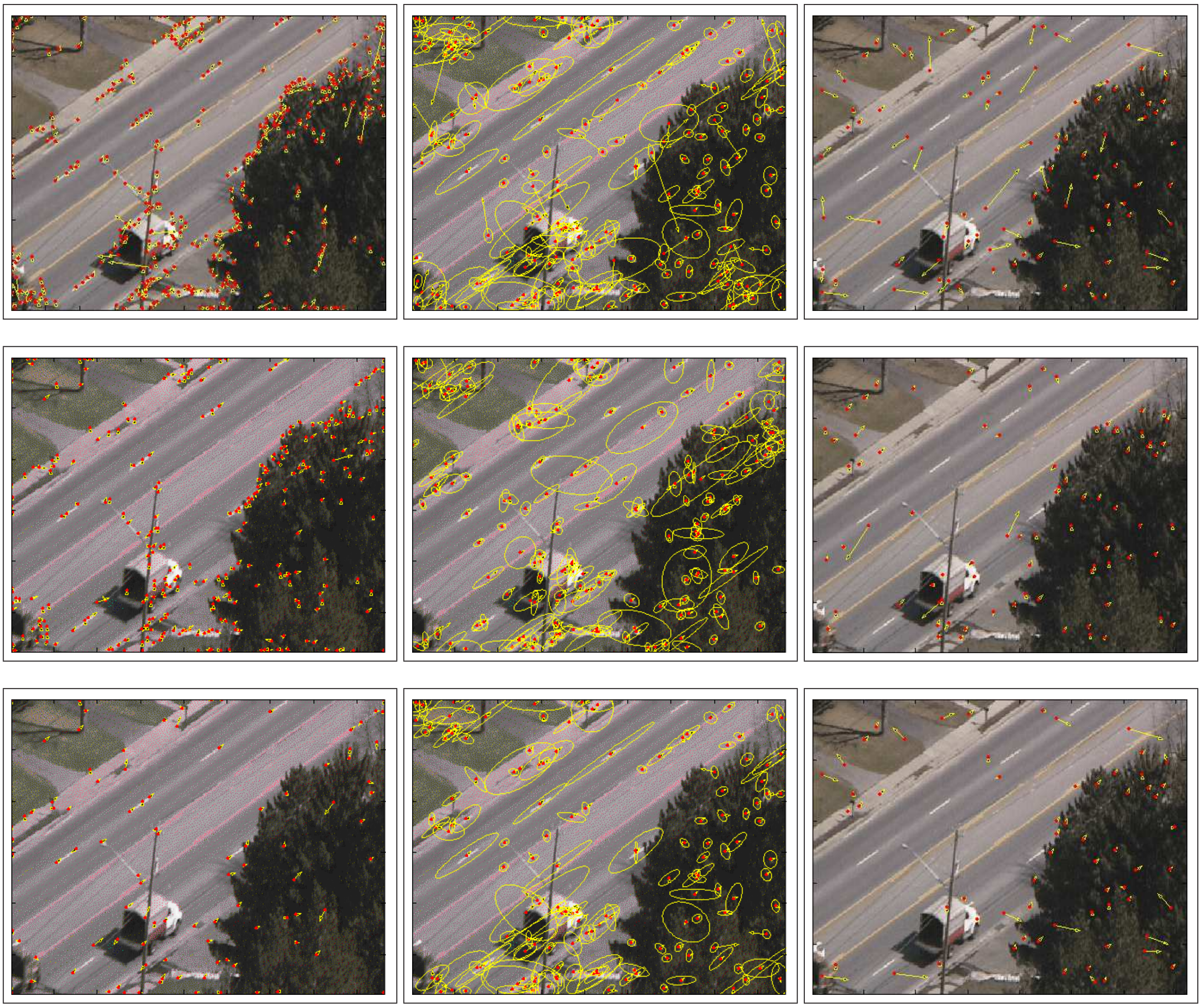


Figure 5. Results for the “Vehicles from the Air” sequence. For every row, each column corresponds to a different image feature(from right-to-left: Harris corners, MSERs and SIFT regions); Top row: Correspondences delivered by the KD-Tree; Middle row: Results for our method; Bottom row: Results for the RANSAC-based method.

consecutive frames.

In Figure 5, from left-to-right, we show the results for the correspondences between the first and the third frames of the sequences yield by the KD-Tree for the Harris corners, MSERs and SIFT descriptors. The red “dot” shows the position of the image feature under study at the frame of reference. The tail of the arrow starts at the red dot and its head points towards the point in which the corresponding feature is at the third frame. In the top row of the figure, we show the initial correspondences as recovered by the KD-Tree, before any outlier removal is performed. In the middle row, we show the results for those correspondences after removing outliers by our method for the three alternatives of image feature vectors. The bottom row shows the results for the RANSAC-based alternative.

From the figure, we can see that our algorithm outperforms the alternative. It has not only removed the outliers, clearly visible in the plots for the non-outlier free correspondences, but it has also recovered a much denser cloud of correspondences and inlier feature points than the RANSAC-based approach. The RANSAC alternative has removed inliers that does not agree with the fundamental matrix recovered under rigid motion assumptions. To see this more clearly, we show in Figure 6 details of the feature cloud corresponding to the car in the sequence. In the left-hand column of the figure we present the results for our algorithm. The right-hand column corresponds to the inliers yield by the alternative. From the figure, we note that, specially in

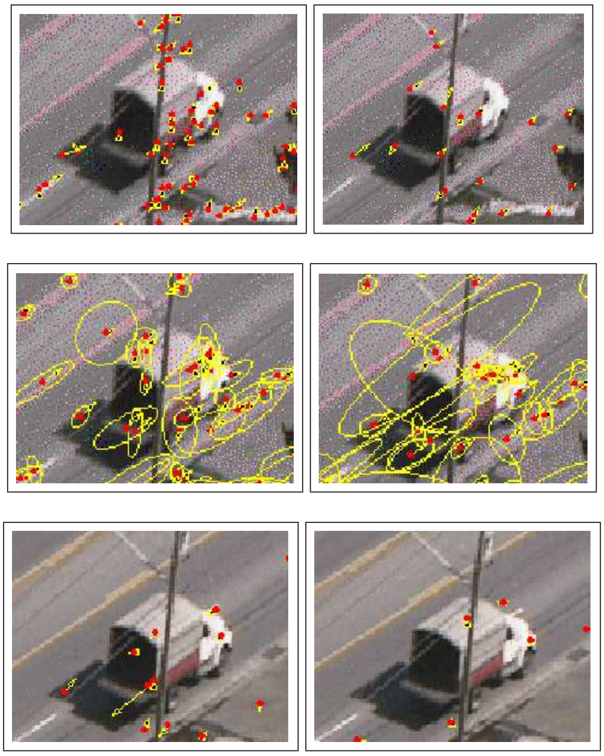


Figure 6. Details for the "Vehicles from the Air" sequence for the three image descriptors used in our experiments. In the left-hand row we show, from top-to-bottom, the results yield by our method when Harris corners, MSERs and SIFT descriptors are used, respectively. The right-hand column shows the results for the alternative.

the case of the Harris corners, the inlier cloud recovered by the algorithm is clearly denser and better localised over the moving car.

4. Conclusions

In this paper, we have shown how the problem of removing outliers for motion analysis can be cast in a graph theoretic setting. The algorithm presented here imposes consistency constraints on the set of putative matches recovered by a matching algorithm in the literature and makes no assumptions on the nature of the motion in the scene. Further, it is not limited to a predetermined number of views and allows the use of an arbitrary number of frames. We have illustrated the utility of the algorithm for outlier removal. Further, the results yield by our algorithm compare favorably with those recovered using an alternative. In future work, we intend to explore the utility of the method for purposes of motion segmentation and tracking in sequences which exhibit non-rigid motion. Our algorithm is quite general and can be used for other outlier correspondence removal tasks.

References

[1] E.H. Adelson and J. Bergen. Spatiotemporal energy models for the perception of motion. *Journal of the Opt. Soc. Am. A*, 2:284–299, 1985. 1

[2] S. Avidan, Y. Moses, and Y. Moses. Centralized and distributed multi-view correspondence. *Int. Journal of Comp. Vision, to appear*, 2006. 1, 2

[3] J.L. Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 8(9), 1975. 6

[4] H. Chui and A. Rangarajan. A new point matching algorithm for non-rigid registration. *Computer Vision and Image Understanding*, 89(2-3):114–141, 2003. 2

[5] T. Cootes, C. Taylor, D. Cooper, and J. Graham. Active shape models - their training and application. *Computer Vision and Image Understanding*, 61(1):38–59, 1995. 1

[6] V. Ferrari, T. Tuytelaars, and L. Van Gool. Wide-baseline multiple-view correspondences. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2003. 1, 2

[7] C. J. Harris and M. Stephens. A combined corner and edge detector. In *Proc. 4th Alvey Vision Conference*, pages 147–151, 1988. 5, 6

[8] R. Hartley and R. Vidal. The multibody trifocal tensor: Motion segmentation from 3 perspective views. In *Proc. of the IEEE Conf. on Comp. Vision and Pattern Recognition*, pages I:769–775, 2004. 1

[9] R. Hartley and A. Zisserman. *Mutliple view goemetry in Computer Vision*. Cambridge University Press, 2000. 1, 6

[10] D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004. 6

[11] J. Matas, O. Chum, U. Martin, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *Proceedings of the British Machine Vision Conference*, pages 384–393, 2002. 6

[12] P. Perona and W. T. Freeman. Factorization approach to grouping. In *Proc. ECCV*, pages 655–670, 1998. 2

[13] A. Rangarajan, H. Chui, and F. L. Bookstein. The softassign procrustes matching algorithm. In *Information Processing in Medical Imaging*, pages 29–42, 1997. 2

[14] S. Sarkar and K. L. Boyer. Quantitative measures of change based on feature organization: Eigenvalues and eigenvectors. *Computer Vision and Image Understanding*, 71(1):110–136, 1998. 2

[15] L. Shapiro and J. M. Brady. Feature-based correspondance - an eigenvector approach. *Image and Vision Computing*, 10:283–288, 1992. 1

[16] L. Shapiro and J. M. Brady. Rejecting outliers and estimating errors in an orthogonal-regression framework. *Phil. Trans. Roy. Soc. A*, 350:403–439, 1995. 2

[17] J. Shi and J. Malik. Normalized cuts and image segmentations. In *Proc. of the IEEE Conf. on Comp. Vision and Pattern Recognition*, pages 731–737, 1997. 2, 4

[18] P. Torr and D. W. Murray. The development and comparison of robust methods for estimating the fundamental matrix. *International Journal of Computer Vision*, 24:271–300, 1997. 1, 2

[19] J. Wills and S. Belongie. A feature based method for determining dense long range correspondences. In *European Conference on Computer Vision*, volume 3, pages 170–182, 2003. 2

[20] A. Yuille and N. M. Grzywacz. A mathematical analysis of the motion coherence theory. *International Journal of Computer Vision*, 3:155–175, 89. 1