Probabilistic Measures for Motion Segmentation

Venu Madhav Govindu
HIG-25, Simhapuri Layout
Visakhapatnam, AP 530047
venu@narmada.org

Abstract

The first contribution of this paper is a probabilistic approach for measuring motion similarity for point sequences. While most motion segmentation algorithms are based on a rank-constraint on the space of affine motions, our method is based on spectral clustering of a probability measure for motion similarity which can be applied to any parametric model. The probabilistic framework allows for incorporation of informative priors for the noise and camera motion. Similarly spatial and temporal priors can also be subsumed leading to useful segmentation techniques. Our second contribution is a tensor-decomposition technique enables us to infer motion affinity from higher dimensional representations. Results are presented on real image sequences.

1. Introduction

During the past decade, a large body of work has clarified our understanding of the geometry of image formation [2] which has resulted in various structure and motion estimation algorithms. A basic assumption for these methods is that the image sequence constitutes a single motion, typically due to a moving camera observing a rigid scene. However, most real situations would consist of sequences that contain objects undergoing different motions. This motivates the development of motion segmentation methods that identifies groups of features or objects that have the same motion. Most of these motion segmentation methods have assumed an affine camera observing feature correspondences and perform a rank-based factorisation ([1, 9] and references therein). While having the advantage of treating all data simultaneously, a common limitation of such approaches is that tracks have to either visible over the entire sequence or have to be "approximated" using rank criteria. Moreover an algebraic rank-criterion does not allow for either an incorporation of information about the camera motion or the segmentation of other parametric models that are not affine (e.g. epipolar geometry). Consequently our motivation is to develop a generative model that allows for a probabilistic interpretation of the motion coherence of features. Given measurements of the motion "similarity" (or likelihood) of tuples of points in the form of an affinity matrix [1], spectral clustering methods allow for a straightforward segmentation of the data. We very briefly state the spectral clustering method here, the reader is referred to [10, 3] for details. Given a positive symmetric matrix $\mathbf{P}$, its *normalised Laplacian* is $\mathbf{D}^{-\frac{1}{2}}(\mathbf{D}-\mathbf{P})\mathbf{D}^{-\frac{1}{2}}$ where $\mathbf{D}$ is diagonal with $\mathbf{d}_i = \sum_j \mathbf{p}_{ij}$. The segmentation is achieved by thresholding (with threshold of zero), the second smallest eigenvector of the normalised Laplacian.

As will be developed in subsequent sections, using a probabilistic framework allows for an intuitively satisfactory measure of motion similarity for a given parametric motion model. Since we can incorporate any parametric motion model into our scheme, it is vastly more general than one that assumes only affine motions. Moreover, a probabilistic framework allows for easy incorporation of prior knowledge of the camera motion, observation noise etc. which increases the accuracy of the method. While the spectral clustering methods are applicable to tuples of features, many models require more feature correspondences. We address this problem by introducing an approach to decomposing a tensorial representation of motion similarity (the multilinear counterpart of the affinity matrix) into an affinity matrix that is efficiently computed using a randomised algorithm.

Section 2 introduces and develops our probabilistic measure for motion segmentation. This probablistic measure for the affine model is described in Sec. 3 and Sec. 3.1 describes the randomised decomposition of the probability tensor. Section 4 describes experimental results on real sequences and Sec. 5 provides some concluding remarks. We are unable to consider other parametric models (epipolar geometry and space-time affine models in particular) in this paper due to space constraints.

[1]The term affinity matrix denotes a matrix of measurements of similarity for clustering and should not be confused with the term affine which denotes a particular parametric motion model.

2. Probabilistic Formulation

The motivation for our approach can be best illustrated using a simple one-dimensional example where the intuitive interpretation of the probability measure is easiest to see. In the following, for pedagogical purposes, we adopt a simple rigid translation model on one-dimensional points. Extension to more complex models like affine motion follow naturally and will be taken up in subsequent sections.

2.1 A Generative Model

Correspondences in the first and second image are denoted $\mathbf{x}$ and $\mathbf{x}'$ respectively and are assumed to move according to either of two translation models, $\mathbf{t}_1$ or $\mathbf{t}_2$. The observed points in the second image are subjected to Gaussian noise of standard deviation σ. Thus, we have $\mathbf{x}' = \mathbf{x} + \mathbf{t} + n$ where $\mathbf{t} \in \{\mathbf{t}_1, \mathbf{t}_2\}$ and the noise term $n \sim N(0, \sigma^2)$. For measuring the affinity of a pair of points, let the tuple of points be $\mathbf{x}_i$ and $\mathbf{x}_j$, where translations $\mathbf{t}_i$ and $\mathbf{t}_j$ are equal if $\mathbf{x}_i$ and $\mathbf{x}_j$ belong to the same motion model. We denote the boolean condition that $\mathbf{x}_i$ and $\mathbf{x}_j$ belong to the same motion model as $\mathbf{x}_i \parallel \mathbf{x}_j$. Thus $P(\mathbf{x}_i \parallel \mathbf{x}_j)$ denotes the probability that $\mathbf{x}_i$ and $\mathbf{x}_j$ belong to the same motion model. Assuming a given translation $\mathbf{t}$, we have for ith point $n_i = \mathbf{x}_i' - \mathbf{x}_i - \mathbf{t}$ which interpreted probabilistically gives

$$P(\mathbf{x}_i \parallel \mathbf{x}_j | \mathbf{t}) = P(n_i)P(n_j) = e^{-\frac{1}{2\sigma^2}({n_i}^2 + {n_j}^2)} \tag{1}$$

since the noise is assumed to be independent and identically distributed. By substituting the term for n_i in Eqn.1 we have the relationship

$$P(\mathbf{x}_i \parallel \mathbf{x}_j | \mathbf{t}) = e^{-\frac{1}{2\sigma^2}({n_i}^2 + {n_j}^2)} = e^{-\frac{1}{2\sigma^2}\{(\Delta\mathbf{x}_i - \mathbf{t})^2 + (\Delta\mathbf{x}_j - \mathbf{t})^2\}}$$

where $\Delta\mathbf{x}$ denotes $\mathbf{x}' - \mathbf{x}$ for notational convenience. However since we neither know the value of $\mathbf{t}$ nor need to estimate it for our purposes (we are only interested in segmentation of points for now) we can integrate it out by means of the Bayes Theorem, i.e. $P(\mathbf{x}_i \parallel \mathbf{x}_j) = \int P(\mathbf{x}_i \parallel \mathbf{x}_j | \mathbf{t}) d\mathbf{t}$ implying that

$$P(\mathbf{x}_i \parallel \mathbf{x}_j) = \int e^{-\frac{1}{2\sigma^2}\{(\Delta\mathbf{x}_i - \mathbf{t})^2 + (\Delta\mathbf{x}_j - \mathbf{t})^2\}} d\mathbf{t} \tag{2}$$

The term of interest in the exponent of the above equation is $(\Delta\mathbf{x}_i - \mathbf{t})^2 + (\Delta\mathbf{x}_j - \mathbf{t})^2$ which simplified to

$$2\left(\mathbf{t} - \frac{\Delta\mathbf{x}_i + \Delta\mathbf{x}_j}{2}\right)^2 + 2\left(\frac{\Delta\mathbf{x}_i - \Delta\mathbf{x}_j}{2}\right)^2 \tag{3}$$

When this term is substituted back into the integral in Eqn. 2 and after integrating out the term $\mathbf{t}$ we have

$$P(\mathbf{x}_i \parallel \mathbf{x}_j) = e^{-\frac{1}{2\sigma^2}\left(\frac{\Delta\mathbf{x}_i - \Delta\mathbf{x}_j}{2}\right)^2} \tag{4}$$

In the above we have neglected some constant terms that have no significance for our result but the intuitive interpretation of the result is clear. For now, neglecting the effect of observation noise, we note that in the case where points $\mathbf{x}_i$ and $\mathbf{x}_j$ belong to the same motion model, we have $\Delta\mathbf{x}_i - \Delta\mathbf{x}_j = 0$ and $P(\mathbf{x}_i \parallel \mathbf{x}_j) = 1$. In the case where the two points belong to different motion models, we note that the exponential term is a function of $(\mathbf{t}_1 - \mathbf{t}_2)^2$ and thus the probability measure $P(\mathbf{x}_i \parallel \mathbf{x}_j)$ will be lower than 1. Consequently the affinity matrix built out of this probability measure can be used to solve for motion segmentation.

2.2 Prior for Motion Model

In the above analysis, we have derived the probability measure of the likelihood that two points belong to the same motion model. In the process of integrating out the translation term $\mathbf{t}$ we have derived the Maximum Likelihood Estimate (MLE) by being *agnostic* about its likelihood, i.e. we have given equal weightage in the integral to every possible translation value. However this is not the case in a real scenario where external constraints tell us that certain motions are more likely than others. For example, for a moving vehicle we know the maximum possible speed and also know that its more likely that the vehicle is moving at its "average" speed than at higher or lower values. This knowledge can be easily incorporated into our derivation in the form of a prior for the motion $P(\mathbf{t})$. Incorporating this prior into the integral of Eqn. 2 will result in a Maximum A Posteriori (MAP) estimate, $P(\mathbf{x}_i \parallel \mathbf{x}_j) = \int P(\mathbf{x}_i \parallel \mathbf{x}_j | \mathbf{t}) P(\mathbf{t}) d\mathbf{t}$. If we have a Gaussian prior, $P(\mathbf{t}) \sim N(\mu_{\mathbf{t}}, {\sigma_{\mathbf{t}}}^2)$ then the integral can be solved for analytically. We should point out that this is not a severe constraint since most well-behaved motion priors can be easily expressed in the form of a mixture of Gaussians, say $P(\mathbf{t}) = \Sigma_k m_k N(\mu_k, {\sigma_k}^2)$ where m represents the mixing proportion of each Gaussian function $N(\mu_k, {\sigma_k}^2)$. It will be noted that due to the linearity of the integral operator, such a mixture of Gaussians model will still result in an analytic form for the integral. In general, we would like to use the MAP estimate when we have some knowledge of the form of the prior for the motion model. In the absence of any useful knowledge of the motion prior we take recourse to the MLE and do not use a prior. While learning a meaningful prior from image sequences is very useful, it remains beyond the scope of this paper. Similarly the use of spatial priors is not dealt with here due to space constraints.

In this section we have derived a measure for the likelihood of a tuple of points moving according to the same motion model. The influence of the priors of noise in the observations and priors for the motion is also described. The probability measure derived in the case of the one-dimensional translation model easily carries over to other more complex motion models as will be described in the next section.

3. Affine Motion Model

In this section we derive probability measures for points moving according to an affine motion model. Following the pedagogic example of one-dimensional points in translation, we can easily modify the probability measure to that of an affine motion model. If we denote point correspondences in two images as $\mathbf{p}$ and $\mathbf{p}^{'}$, then under an affine motion we have $\mathbf{p}^{'} = A\mathbf{p} + \mathbf{t}$ where A is a 2×2 affine matrix and $\mathbf{t}$ is a two-dimensional translation parameter. Under observation noise, the more general relationship is $\mathbf{p}^{'} = A\mathbf{p} + \mathbf{t} + \mathbf{n}$ where $\mathbf{n}$ is the noise term. We assume in subsequent analysis that the noise term is Gaussian distributed with zero mean, i.e. $\mathbf{n} \sim N(\mathbf{0}, \Sigma_n)$ where Σ_n is a 2×2 covariance matrix. Rewriting the relationship in vector form we have

$$\left[\begin{array}{c} x^{'} \\ y^{'} \end{array} \right] = \left(\begin{array}{cc} a_{11} & a_{12} \\ a_{21} & a_{22} \end{array} \right) \left[\begin{array}{c} x \\ y \end{array} \right] + \left[\begin{array}{c} t_x \\ t_y \end{array} \right] + \left[\begin{array}{c} n_x \\ n_y \end{array} \right] \quad (5)$$

By collecting the unknown motion parameters into a single vector $\mathbf{a} = [a_{11}, a_{12}, t_x, a_{21}, a_{22}, t_y]$ we can rewrite Eqn. 5 as

$$\left[\begin{array}{c} x^{'} \\ y^{'} \end{array} \right] = \underbrace{\left(\begin{array}{cccccc} x & y & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x & y & 1 \end{array} \right)}_{=\mathbf{M}} \mathbf{a} + \left[\begin{array}{c} n_x \\ n_y \end{array} \right] \quad (6)$$

which can be compactly rewritten as $\mathbf{p}^{'} = \mathbf{Ma} + \mathbf{n}$. Note here that $\mathbf{p}^{'}$ and $\mathbf{M}$ are known quantities based on point correspondences. Following Eqn. 1, for $P(\mathbf{p}_i \parallel \mathbf{p}_j | \mathbf{a})$ we have $P(\mathbf{n}) = e^{-\frac{1}{2}\mathbf{n}^T {\Sigma_n}^{-1} \mathbf{n}}$ where $\mathbf{n}$ can be obtained from Eqn. 6, i.e. $\mathbf{n} = \mathbf{p}^{'} - \mathbf{Ma}$. Rewriting the noise terms we have

$$\begin{aligned} \mathbf{n}^T {\Sigma_n}^{-1} \mathbf{n} &= \left(\mathbf{p}^{'} - \mathbf{Ma}\right)^T {\Sigma_n}^{-1} \left(\mathbf{p}^{'} - \mathbf{Ma}\right) \\ &= \mathbf{a}^T \mathbf{M}^T \Sigma_n \mathbf{Ma} - 2\mathbf{p}^{'T} {\Sigma_n}^{-1} \mathbf{Ma} + \mathbf{p}^{'} {\Sigma_n}^{-1} \mathbf{p}^{'} \\ &= \mathbf{a}^T \mathbf{Ba} + \mathbf{c}^T \mathbf{a} + \mathbf{d} \end{aligned} \quad (7)$$

where terms $\mathbf{B}$,$\mathbf{c}$ and $\mathbf{d}$ are derived from the correspondences and the noise covariance. The above derivation is for the probability measure for a single point. However since we assume that the noise is Gaussian and iid, the noise terms are additive in the exponents. In other words, when we consider the joint probability of two noise terms from points $\mathbf{p}_i$ and $\mathbf{p}_j$ the terms $\mathbf{B}$,$\mathbf{c}$ and $\mathbf{d}$ would be modified to the arithmetic sum of the individual terms, e.g. $\mathbf{B} = \mathbf{B}_i + \mathbf{B}_j$. The reader will notice that given this conditional probability we can derive the analytic probability measure by integrating out the motion parameters $\mathbf{a}$, i.e. we have for $P(\mathbf{p}_i \parallel \mathbf{p}_j) = \int P(\mathbf{p}_i \parallel \mathbf{p}_j | \mathbf{a}) d\mathbf{a}$, the form

$$\int P(\mathbf{p}_i \parallel \mathbf{p}_j | \mathbf{a}) d\mathbf{a} = \int e^{-\frac{1}{2}\{\mathbf{a}^T \mathbf{Ba} + \mathbf{c}^T \mathbf{a} + \mathbf{d}\}} d\mathbf{a} \quad (8)$$

The analytic form for this equation is easily derived by completing the squares in variable $\mathbf{a}$ in a manner similar to that in Eqn. 3. However in contrast with the translation model since the affine motion model has 6 unknown parameters, two correspondences $\mathbf{p}_i$ and $\mathbf{p}_j$ are not sufficient to solve for the affine motion as they provide only four constraints. An even greater constraint is that for the measure to be accurate we need a correct prior for the affine model since we are integrating over 6 dimensions. In the absence of a known prior for the motion model, we use the following estimator for the affinity between sets of points.

Given a set of k-tuple $(k \geq 4)$ correspondences we linearly solve for the affine motion model and estimate the probability measure according to the noise model, $\mathbf{n} \sim N(0, \Sigma_n)$. If all points in the k-tuple belong to the same motion model, the residual error will be low and the probability measure will be high. The converse is true for the case where the points belong to different motion models as the linear estimate will be poor. An alternate way to interpret this approach is the probability measure thus derived is similar to that in Eqn. 8 with a strong prior for the affine motion placed at $\mu_{\mathbf{a}} = \mathbf{a}^\star$ where $\mathbf{a}^\star$ is the linear affine estimate using the k-tuple of points. The resultant representation for the probability measure will be a tensor of dimensions k. However the spectral clustering techniques are based on the affinity of tuples of points. Thus we require an appropriate technique that will convert a k-dimensional tensorial representation to a two-dimensional probability matrix. Recent work in the analysis of multilinear decompositions have extended standard linear algebra ideas like the singular value decomposition (SVD) and the eigen-decomposition to higher dimensional representations [4, 5] and [7, 8] apply these ideas to problems in computer vision.

3.1 Decomposition of the Probability Tensor

In the following we provide a brief sketch of tensor-based ideas that are used in our scheme, the reader is referred to [4, 8] for a detailed treatment. In the case of matrices $\mathbf{P}$, the singular value decomposition $\mathbf{P} = \mathbf{U}_1\mathbf{S}\mathbf{U}_2{}^T$ provides an orthogonalisation of the row and columns spaces in matrices $\mathbf{U}_1$ and $\mathbf{U}_2$. For an N-dimensional tensor $\mathcal{P}$, the "flattened" matrix representation $\mathbf{P}_{(n)}$ can be obtained by varying the index along dimension n while holding all other dimensions fixed. Each such instance gives a vector which is a column in $\mathbf{P}_{(n)}$. The n-mode product of a tensor $\mathcal{P}$ and a matrix $\mathbf{U}$ can now be defined as follows $\mathcal{P} \times_n \mathbf{U} = \mathbf{P}_{(n)}\mathbf{U}$. Using this representation we can rewrite the matrix SVD as $\mathbf{P} = \mathbf{U}_1\mathbf{S}\mathbf{U}_2{}^T = \mathbf{S}\times_1\mathbf{U}_1 \times_2 \mathbf{U}_2$. By extension, the tensorial decomposition is given by $\mathcal{P} = \mathcal{S}\times_1\mathbf{U}_1 \times_2 \mathbf{U}_2 \cdots \times_n \mathbf{U}_n$, where $\mathcal{S}$ is known as the core-tensor and is the multilinear equivalent of the singular value matrix. The mode matrices $\mathbf{U}_n$ are obtained by an SVD of the flattened matrix $\mathbf{P}_{(n)}$ and setting $\mathbf{U}_n$ to the left matrix of the SVD.

In our case for k-tuples of points, we have a probability tensor represented by $\mathcal{P}$, where the entry $p_{i_1,\cdots,i_k}$ is the probability measure for the set of points indexed by $\{i_1,\cdots,i_k\}$. Here the k-dimensional tensor is of dimensions $N \times N \cdots \times N$ where N is the total number of points. Moreover, we note that the probability measure is is invariant to all index permutations $\pi\{i_1,\cdots,i_k\}$ where π is any element of the set of permutations of k indices. For example, the affine motion probability for points $\{3,4,5,6\}$,$\{3,5,6,4\}$,$\{5,6,4,3\}$ are all identical. In other words, the probability tensor $\mathcal{P}$ has a super-symmetric structure where all the k-mode spaces are identical. Thus the decomposition can be obtained by flattening the tensor along any dimension. We also note that for our purposes we are not interested in the orthogonal representation $\mathbf{U}$ alone but want to decompose the tensor into a form where $\mathbf{V}$ is the desired affinity matrix such that the mode product of $\mathbf{V}$ with itself along all dimensions results in the required tensor, i.e. $\mathcal{P} = \mathbf{V} \times_1 \mathbf{V} \times_2 \mathbf{V} \cdots \times_k \mathbf{V}$. Note that this decomposition is valid only for the case where the tensor $\mathcal{P}$ is super-symmetric, the proof of which is beyond the scope of this paper. In general, we note that the affinity matrix $\mathbf{V}$ is of size $N \times N$ and is derived from the left space of the flattened matrix $\mathbf{P}_{(n)}$, i.e. $\mathbf{V} = \mathbf{P}_{(n)}\mathbf{P}^T{}_{(n)}$. The flattened matrix $\mathbf{P}_{(n)}$ is of size $N \times N^{k-1}$, thus the large number of columns is a great computational load. But the problem is further simplified if we consider the following representation. We write matrix $\mathbf{P}_{(n)} = [c_1, c_2, \cdots, c_s]$ where c_i denotes the ith column. Now $\mathbf{V}$ can be written as $\sum_i c_i c_i{}^T$. This suggests that instead of computing the entire tensor and flattening it, we can provide an approximation to the affinity matrix $\mathbf{V}$ by considering the sum $\sum_i c_i c_i{}^T$, where the columns are a small subset of the entire set of columns of the flattened representation $\mathbf{P}_{(n)}$. This procedure can be summarised in the following algorithm which is both efficient and also does not require large-scale memory allocation for the tensor since computations are done *in situ*.

Let $\mathcal{R} = \{1, 2, \cdots, N\}$ be the set of indices for N correspondences and let k be the dimension of the tensor $\mathcal{P}$, i.e. k-tuples are used to compute the probability measure. Also we denote the probability measure for the k-tuples $\{i_1, i_2, \cdots, i_k\}$ as $\mathbf{p}_{\{i_1,i_2,\cdots,i_k\}}$

Randomised Algorithm for Decomposition of Probability Tensor

Set $\mathbf{V}$ to be an $N \times N$ matrix of zeros.
for $\mathbf{T}$ trials do

- Set $\mathbf{v}$ to an N-dimensional column vector of zeros
- Randomly select $(k-1)$ indices $I = \{i_1, i_2, \cdots, i_{(k-1)}\}$ from $\mathcal{R}$
- $\forall i \in \mathcal{R}$ and $i \notin I$ compute $\mathbf{v}(i) = \mathbf{p}_{\{i,I\}}$
- Update $\mathbf{V} \leftarrow \mathbf{V} + \mathbf{v}\mathbf{v}^T$

The resulting matrix $\mathbf{V}$ is the desired affinity matrix measuring the similarity of motion between all pairs of points. An alternate interpretation of this algorithm is that the entry $\mathbf{V}(i,j)$ is given by $\sum_I \mathbf{p}_{\{i,I\}}\mathbf{p}_{\{j,I\}}$ where $i \cap I = \emptyset$ and $j \cap I = \emptyset$, where I are sets of $(k-1)$-tuples drawn from $\mathcal{R}$. This can be seen to be a particular form of marginalisation of the multidimensional probability tensor $\mathcal{P}$ into a two-dimensional affinity matrix.

4. Experimental Results

In this section we present results on two image sequences that illustrate the ability of our motion segmentation scheme. In the computational perception of scene dynamics [6], a preliminary step is to segment the scene into coherently moving "objects". Subsequent analysis is used to label these object various as "active" or "passive" agents or stationary objects etc. from which the perception is inferred. In Fig. 1 we illustrate this problem using a sequence where a moving hand picks up a soda can. Frames from the

(a) Moving Hand (b) Segmentation (c) After Pickup

Figure 1: (a) shows the moving hand at the beginning of the sequence, (c) shows the end of the sequence, after the can has been picked up. (b) shows the correct segmentation obtained over the entire sequence. Note that while the can and the hand are moving coherently after the pickup, the fact that only the hand was moving before the pickup of the can is reflected in the correct segmentation achieved.

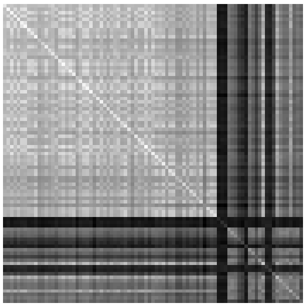

Figure 2: The affinity matrix between adjacent images of the aerial sequence is estimated using the randomised algorithm for tensor decomposition ($\mathbf{T} = 200$ samples). The features belonging to the static background can be correctly inferred from this affinity matrix (the large block).

beginning and end of the sequence are shown in Fig. 1(a) and (c) respectively. If we were to segment the scene using the first few images of the sequence, the moving hand would be segmented out of the rest of the scene. Similarly just using the last few images would result in concluding that both the hand and the soda can constitute a single object as they move coherently. Thus it is important to use the entire sequence to arrive at the correct segmentation.

In our experiment we track feature points (using the KLT tracker) over the entire sequence of 84 images and use the correspondences from every fifth image resulting in 17 sets of correspondences. For motion segmentation we use a simple two-dimensional translation motion model (with the σ of noise set to 1 pixel) and compute the affinity matrix between adjacent frames. Subsequently the segmentation results of each image pair are computed by spectral clustering and the information combined to infer all possible unique labels. The resulting segmentation is shown in Fig. 1(b) and can be seen to show the correct segmentation of the hand, the soda can and the rest of the scene into independent motion segments. The single instance of a point on the table being grouped along with the hand is a result of a tracking error where the tracker confuses a moving shadow with a point on the hand and generates a false track.

In Fig. 3 we illustrate the use of our affine motion segmentation scheme on a long aerial sequence. The sequence consists of a bridge where two vehicles move independently and the camera tracks them for over 700 frames. It must be noted that in such a long sequences the tracks of feature points will be visible only within a limited range of images as they disappear from the field of view as the camera moves ahead. In order to interpret the information in the sequence it is useful to build a mosaic of the scene being viewed. However to correctly estimate the motion between image frames we need to segment the correspondences into those belonging to the background and those on moving objects.

As in the previous example, we use the KLT tracker to track feature points over the sequence of 700 images. Subsequently the sequence is decimated to 35 images. We assume that the warping between the frames is adequately described by an affine transformation and for every adjacent pair of images, we compute the affinity matrix using the method outlined in Sec. 3. For the randomised decomposition algo-

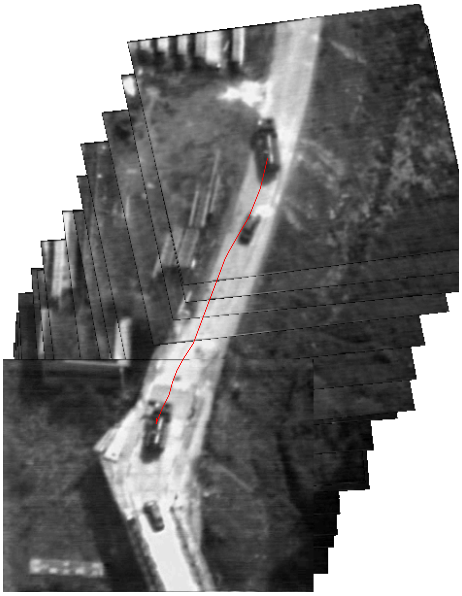

Figure 3: The background mosaic is computed after motion segmentation between adjacent frames. The estimated track of one of the segmented vehicles is superimposed in red.

rithm of Sec. 3.1, we chose $k = 4$ (i.e. 4 correspondences are selected at a time) and we use 200 samples ($\mathbf{T} = 200$) to estimate the affinity matrix. One instance of the estimated affinity matrix between correspondences is shown in Fig. 2 and we can observe that the affinity matrix correctly reflects the relationship between the large number of feature points that belong to the static background (although the camera is moving) and those that belong to objects that are moving. Once the segmentation of the feature points is achieved using spectral clustering on the affinity matrix, the correct motion between the adjacent frames can be inferred. Moreover once we warp the adjacent frames into a common reference frame, the moving objects can be trivially segmented out since the difference in image intensities would be high for the moving vehicles. We use a simple thresholding of the energy difference between the images registered onto a common reference frame (i.e. mosaic) and use simple morphological operations to compute the centre of the blob corresponding to the moving vehicles which enables us to locate the moving objects on the mosaic. The resulting representation is shown in Fig. 3. A mosaic is built out of the images used and gives us a clear view of the background scene. We also show the vehicles in the first and last frame and the estimated trajectory of one of the vehicles is superimposed in red. As can be seen, our motion segmentation scheme correctly segments the feature points allowing us to build accurate background mosaics and also estimate the motion trajectory of the moving objects.

5. Conclusions

In this paper we have developed a probabilistic measure for the motion similarity between corresponding points. This measure can be computed for any parametric model and the resulting affinity matrix allows for motion segmentation using spectral clustering. We also introduce a randomised algorithm for tensor decomposition that allows us to naturally decompose a probability tensor into its two-dimensional affinity matrix representation. This decomposition method is applicable to any super-symmetric tensor and can be used for other labeling problems as well. Future work will include other more complex parametric models, incorporation of motion priors and a more formal analysis of the decomposition algorithm for super-symmetric tensors.

6. Acknowledgments

Richard Mann provided the *pick-up* image sequence and the use of the publicly available KLT tracker software is acknowledged.

References

[1] J. Costeira and T. Kanade. A multibody factorization method for motion analysis. In *IEEE International Conference on Computer Vision*, pages 1071–1076, 1995.

[2] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[3] D. J. Higham and M. Kibble. A unified view of spectral clustering. In *University of Strathclyde Mathematics Research Report 02*, 2004.

[4] L. Lathauwer, B.D.Moor, and J.Vandewalle. A multilinear singular value decomposition. *SIAM Journal of Matrix Analysis and Applications*, 21(4):1253–1278, 2000.

[5] L. Lathauwer, B. D. Moor, , and J.Vandewalle. On the best rank-1 and rank-(r1; r2; : : : ; rn) approximation of higher-order tensors. *SIAM Journal of Matrix Analysis and Applications*, 21(4):1324–1342, 2000.

[6] R. Mann, A. Jepson, and J. Siskind. The computational perception of scene dynamics. In *European Conference on Computer Vision*, pages 528–539, 1996.

[7] A. Shashua and A.Levin. Linear image regression and classification using the tensor-rank principle. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2001.

[8] M. A. O. Vasilescu and DemetriTerzopoulos. Multilinear analysis of image ensembles: Tensorfaces. In *European Conference on Computer Vision*, pages 447–460, 2002.

[9] R. Vidal and R. Hartley. Motion segmentation with missing data using powerfactorization and gpca. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2004.

[10] Y. Weiss. Segmentation using eigenvectors: A unifying view. In *IEEE International Conference on Computer Vision*, pages 975–982, 1999.