Recursive Tracking of Image Points Using Labelled Graph Matching*

S. Chandrashekar
Signal and Image Proc. Institute
Dept. of Electrical Engg–Systems
University of Southern California
Los Angeles, CA 90089-2564

C. von der Malsburg
Lehrstuhl für Biophysik
Institut für Neuroinformatik
Ruhr-Universität Bochum
Germany

R. Chellappa
Department of Electrical Engineering
Center for Automation Research
University of Maryland
College Park, MD 20742

Abstract—This paper addresses the problem of obtaining the image-plane trajectories of feature points from a sequence of images. The matching of points between successive images in the sequence is treated as a labelled graph matching problem, and solved by minimizing a weighted sum of three cost functions. The nodes in the graph are the feature points to be matched, which are labelled using Gabor wavelets. Information gathered from the previous image frames is processed by a Kalman filter, and used to predict the positions of the match points in the incoming image, and thereby initialize the graph matching. Results on synthetic and real image sequences are presented.

1 INTRODUCTION

The analysis of visual motion from a sequence of images is one of the basic problems in computer vision. The so-called feature-based methods (e.g. [1, 9]) for solving this problem rely on the recognition of the same set of corresponding features (such as points or lines) in two or more images. Given these “image-plane” trajectories of selected features, various algorithms can be applied to estimate the relevant motion parameters.

In this paper, we present a scheme for obtaining the trajectories of salient points (feature points) in a long monocular sequence of images. We combine the method of recursive filtering used for motion analysis with the technique of labelled graph matching as applied in [3, 2, 7] for obtaining feature correspondences for pattern recognition.

The problem of estimating the image-plane trajectory of a point is formulated as a recursive tracking problem, based on a plant model and an observation model. The parameters relating to the position and motion of the point are contained in a state vector, whose time-evolution is represented by the plant model. In this work, a linear model is assumed for the motion of the feature points in the image plane, and each point is processed separately. More complex 3-D motion models can be used if further knowledge is available about the manner in which the image sequence was generated. The measurement vector contains the observed feature point positions. The observation model represents the relationship between the state vector and the measurements. Together, the two models constitute a state space representation of the problem, suitable for solution using a Kalman filter.

Feature point correspondence is posed as a labelled graph matching problem, with the feature points treated as nodes of a labelled graph. This technique has been applied in [2] for face recognition, with encouraging results. The problem of trajectory analysis is somewhat similar to the object recognition problem in the sense that both usually require finding a correspondence between distinct features in two or more images, or between a stored pattern and a test pattern. In both cases, labelled graph matching provides the required invariance to limited amounts of distortion, unlike correlation-based methods which are sensitive to distortion.

The performance of the labelled graph matching procedure depends on the manner in which feature points are

*Partially supported by the Office of Naval Research under the grant N00014-89-1-1588.
selected and labelled. The images in the sequence have to be suitably processed to extract features and to obtain a rich description (labelling) of the features so that the probability of errors in matching is minimal. A feature point can be labelled in several ways, based on the intensity distribution in a neighbourhood around it. Image gray levels cannot be used directly, since they seldom remain constant over significant time intervals. To provide a better characterization of image points, we use Gabor wavelets (sometimes called Morlet functions [5]), which are more robust in the presence of noise and fluctuations in illumination. In addition, they have several desirable properties such as variable resolution and optimal localization in the spatial and frequency domains [4, 8].

The matching of feature points is interleaved with the recursive estimation of the trajectories. Current information about the trajectories is used to predict the future positions of feature points, thereby reducing the search time for finding matching points. Feature points are not assumed to have already been extracted in all the images in the sequence; instead, selected feature points in the first image in the sequence are "tracked" over successive images in the sequence by labelled graph matching between consecutive image frames. Thus feature point extraction in all images but the first is done automatically.

2 FORMULATION FOR RECURSIVE SOLUTION

The idea here is to formulate the estimation of a point's trajectory as a recursive tracking problem, based on a plant model and a measurement model.

The motion of a point \( p = (x, y)^T \) in the image is modelled by the following equation:

\[
p(t) = p(0) + \dot{p}(0) t + \ddot{p}(0) t^2/2! + \cdots + p^{(n)}(0) t^n/n!,
\]

where \( n \) is small compared to the number of frames in the sequence. (In other words, it is assumed that derivatives of \( p(t) \) higher than the \( n \)th are negligible for all \( t \).) Typically, one would select \( n = 1 \) or \( n = 2 \).

The quantities to be estimated i.e. the position of the point and its derivatives are contained in a state vector \( s \), defined by

\[
s \triangleq \begin{bmatrix}
p(t) \\
\dot{p}(t) \\
\ddot{p}(t) \\
\vdots \\
p^{(n)}(t)
\end{bmatrix}
\]

The plant model describes the time evolution of the state vector. Using (1), which expresses the assumption that \( p^{(m)}(t) = 0 \) for \( m > n \), it can be written as

\[
s(t) = Fs(t) + w(t)
\]

where \( w \) is a noise term included to take into account modelling errors, and the matrix \( F \) is of the form

\[
F = \begin{bmatrix}
0 & I_2 & 0 & 0 & \cdots & 0 \\
0 & 0 & I_2 & 0 & \cdots & 0 \\
\vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\
0 & 0 & \cdots & \cdots & \cdots & I_2 \\
0 & 0 & \cdots & \cdots & \cdots & 0
\end{bmatrix}
\]

where \( I_2 \) is the \( 2 \times 2 \) identity matrix. Equation (3) has to be discretized, in order that it can describe the evolution of the state vector from one sampling instant to another. The discrete version of the plant model can be obtained by integration of (3) over the sampling interval (i.e. interframe period), and the result is

\[
s(k) = F s(k - 1) + w_k
\]

where the matrix \( F \) is given by

\[
F = \begin{bmatrix}
I_2 & t & I_2 & \cdots & \cdots & t \cdot I_2 \\
0 & I_2 & t & I_2 & \cdots & \cdots & (n-1) \cdot I_2 \\
\vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\
0 & 0 & \cdots & \cdots & I_2 & t \cdot I_2 \\
0 & 0 & \cdots & \cdots & I_2 & 0
\end{bmatrix}
\]

and \( w_k \) is a discretized plant noise term, whose covariance \( Q_k \) is manipulated to ensure filter convergence.

The input data to the estimator are in the form of image point positions, one for each sampling instant. The measurement model, which shows the relationship between the state vector \( s \) and the observation (measurement) vector \( z \) is given by

\[
z(k) = H s(k) + v_k
\]

where

\[
H = \begin{bmatrix}
I_2 : 0 : 0 : \cdots : 0
\end{bmatrix}
\]

and \( v_k \) is the measurement noise, which in most cases is equal to the spatial quantization noise.

The formulation is now ready for solution by a Kalman filter. Details of Kalman filtering can be found in [6].

3 FEATURE POINT LABELLING

We use a form of wavelet decomposition called Morlet transform [5] converting the image into a "resolution pyramid" by filtering with Gabor-like kernels of the form

\[
G(x, k) = e^{(ik \cdot x)} e^{-\frac{|k|^2 \cdot x^2}{2 \sigma^2}}
\]
Each kernel has two vector arguments, 2-D position \( x \) and wave number \( k \). The filtering step consists of convolving the image \( I(x) \) with the kernels \( G(x, k) \) i.e.
\[
f(x, k) = \int I(x') G(x - x', k) dx'.
\] (10)

By varying \( k \) in magnitude and orientation (phase), we get a vector of labels \( f(x, k) \) at each pixel \( x \). The magnitude represents local spatial frequency (or resolution) of the feature detector. In our experiments, we found it sufficient to use four levels of resolution, and four orientations at each level. Since the kernels consist of complex numbers, this results in a label vector of 16 complex numbers for every point in the image. This vector will be referred to as a "jet"\(^1\). Matching two points can now be done by comparing their jets, and can be accomplished with greater reliability than what would be possible by using image intensities alone.

4 FEATURE POINT MATCHING

Feature point matching between two images \( I_1 \) and \( I_2 \) is performed using the principles of labelled graph matching, which have been successfully applied in [2] for performing distortion-invariant pattern recognition. Let us assume that feature points in \( I_1 \) are available, and that we wish to find the matching points in \( I_2 \). Feature points are treated as nodes in a labelled graph, with the (vector) labels being the jets obtained by convolution with the Gabor wavelet kernels. Neighbouring feature points in \( I_1 \) are linked to form a topological graph. This can be done automatically, using the interpoint distance as a basis for linkage; in our work this is done by inspection. Matching then consists of dynamically assigning image points in \( I_2 \) to the given feature points in \( I_1 \)\(^2\). This assignment is guided by three criteria (a) similarity of the jets of potential match points (b) preservation of the local topology of the graphs of the feature points in the two images and (c) nearness of the match points to their predicted locations. The matching is treated as minimization of a compound cost function of the form
\[
\sum_i [C_S(i, i') + \alpha C_T(i, i') + \beta C_D(i, i')]
\] (11)

and \( \alpha \) and \( \beta \) are weighting parameters. The prime on the second argument of the functions \( f' \) refers to fact that it is the match point in the second image corresponding to the \( i \)th point in the first image. These costs may be computed in many different ways. Some discussion of this can be found in [2]. For instance, we could use the following cost functions:
\[
C_S(i, i') = \sum_j ||jet(i) - jet(i')||^2
\] (12)
\[
C_T(i, i') = \sum_{j \in N(i)} [d(i, j) - d(i', j')]^2
\] (13)

where \( jet(i) \) is the vector of labels of the point \( i \), \( N(i) \) is the set of neighbours of \( i \), and \( d(\ldots) \) measures the Euclidean distance between image points. Computation of the diffusion cost will be explained in the next section. If the \( C_S \) or \( C_D \) cost terms for a point are inordinately high after minimization, it is assumed that the point has been "lost" due to occlusion or other causes, and it is not tracked any further.

We have so far has assumed that the matching is done at all resolutions simultaneously, as is done in [2]. In our work we use a hierarchical multi-scale approach, using the matches obtained at coarser levels to guide the matches at finer levels. Here "coarse" and "fine" refer to the magnitude of the spatial frequency of the Gabor kernel, the highest frequency corresponding to the finest resolution. At coarser levels, the search is conducted in a larger neighbourhood, sampling it sparsely, and at finer levels in a smaller, densely sampled neighbourhood.

5 INTERLEAVING MOTION ESTIMATION AND CORRESPONDENCE

The matching process described above will require that the location of match points in \( I_3 \) be known approximately to start with. If this is not the case, the search region for match points will be very large, resulting in extremely high computation time, and highly increased probability of false matches. This is where the strength of the recursive filtering approach lies; given the current state vector \( \dot{s}_i \) corresponding to \( i \)th feature point, the position \( p_i \) of the point in the image can be predicted at the next time instant with a known uncertainty. In standard estimation-theoretic notation, this can be expressed as
\[
\hat{p}_i(k|k-1) = H F \dot{s}_i(k-1|k-1).
\] (14)

The covariance, or uncertainty, in this prediction can be shown to be:
\[
C_i(k|k-1) \overset{\Delta}{=} C_{\text{cov}}(p_i(k|k-1)) = H(F P_i(k-1|k-1) F^T + Q_k) H^T
\] (15)

233
The predicted feature point positions can then be used to initialize the matching process, and the covariance of the prediction to control the “diffusion” i.e. search of match points during the matching. In other words, the search for a match point is conducted in a region around its predicted location, the size of this region being proportional to the uncertainty in the prediction. Further, the diffusion cost term $C_D$ in (11) is chosen so as to favour matches close to their predicted locations. To be precise, it is selected to be the sum of the “Mahalanobis distances” from the predicted locations, i.e.

$$C_D(i) = \sum_i d_i(k)^T C_i(k/k-1)^{-1} d_i(k)$$

where

$$d_i(k) \triangleq \hat{p}_i(k|k-1) - z_i(k)$$

The matching procedure yields the measurement $z_i(k)$, which is then used to perform a “measurement update” on the state vector $s_i$. This is done for all the feature points. The system is then ready to process the next image in the sequence.

6 EXPERIMENTAL RESULTS

The method was tested on a number of image sequences, both real and synthetic, with different patterns of image motion. Results for two real image sequences are discussed in this section. In the motion model, a value of $n = 1$ was used, corresponding to constant image-plane velocities for the feature points. This assumption is not very restrictive, since a well-designed recursive algorithm has the ability to track the states even in the presence of model deviations. Experimental results are shown in Figs. 2 and 1. The trajectories determined by the method are shown superimposed on the first and last images in the sequence, to enable the observer to validate them. Also shown are the selected feature points in the first image and the topological graph linking them.

The first sequence, consisting of ten 512x512 images, was taken by a camera mounted on a PUMA robot arm. Its motion is approximately a rotation around the optical axis of the camera. The scene is the interior of a room with several objects, and several polygonal patterns on the floor, walls and ceiling. The corners of these polygons are ideal choices for feature points, since they are precisely localizable, and can be expected to have Gabor profiles easily distinguished from those of neighbouring points. The second real sequence containing 16 images of dimensions 512x512 was obtained by a camera moving forward along its optical axis. The original sequence has 151 closely sampled image frames, from which we selected every 10th image. The scene is the top of a table with several small objects on it. The features in this case are not so well defined. For instance, the tops of the pencils have been chosen as feature points, although they are several pixels wide. This is not seen to be a serious problem, because the multi-scale nature of the Gabor wavelet representation makes it possible to treat extended features as point features, and match them within the same framework.

ACKNOWLEDGMENTS

The robot arm sequence was provided by the Computer Science Department at the University of Massachusetts. The table-top image sequence was provided by Dr. B. Sridhar. The authors also wish to thank B. S. Manjunath for helpful discussions.

References


234
Figure 1: The 1st, 8th and 16th frames of a sequence taken by a camera translating along its optic axis are shown in (a), (b) and (c). The graph topology used for matching is shown in (d). The resulting trajectory estimates are shown in (e) and (f), superimposed on the 1st and 16th frames, respectively.
Figure 2: The 1st, 5th and 10th frames of a sequence taken by a camera attached to a robot arm are shown in (a), (b) and (c). The graph topology used for matching is shown in (d). The resulting trajectory estimates are shown in (e) and (f), superimposed on the 1st and 10th frames, respectively.