PROC. ICIP 2002. VOL I, PP. 565-568, IEEE. ROCHESTER, USA, SEPTEMBER 2002.

SPEEDING UP SSD PLANAR TRACKING BY PIXEL SELECTION

José Miguel Buenaposada, Luis Baumela

Departamento de Inteligencia Arti£cial, Universidad Polit´ecnica de Madrid
Campus de Montegancedo s/n, 28660 Boadilla del Monte (Madrid). SPAIN
jmbuena@dia.fi.upm.es, lbaumela@fi.upm.es

ABSTRACT

In this paper we present a method to estimate in real-time the position and orientation of a previously viewed planar patch. The algorithm is based on minimising the sum of squared differences between a selected set of pixels obtained from a previously stored image of the patch and the current image of it. First a linear model for projectively tracking a planar patch is introduced, then, a procedure to accelerate the tracking process is presented. This procedure is based on using a small set of informative points in the tracking process. In the experiments conducted we show the gain in performance and compare different procedures to select the set of points used in tracking.

1. INTRODUCTION

Tracking planar patches is a subject of interest in computer vision, with applications in augmented reality [1], mobile robot navigation [2], face tracking [3], or the generation of super-resolution images [4]. Traditional approaches to tracking are based on £nding correspondences in successive images. This can be achieved by computing optical ¤ow [5] or by matching a sparse collection of features [6]. In ¤ow-based methods, a velocity vector is computed for each pixel, while in feature-based methods, image features, such as points and lines are matched across all frames in the sequence. Feature-based methods minimise an error measure based on geometrical constraints between a few corresponding features, while direct methods minimise an error measure based on direct image information collected from all pixels in the image, such as image brightness.

The tracking method presented in this paper belongs to the £rst group of methods. It is based on minimising the sum-of-squared differences (SSD) between a selected set of pixels obtained from a previously stored image of the tracked patch (image template) and the current image of it. It extends Hager's SSD tracker [7] by introducing a projective motion model and a procedure to select the set of pixels used in tracking.

This work was funded by the Spanish Ministry of Science and Technology under grant number TIC1999-1021.

2. SSD PLANAR TRACKING

Let P be the image of a planar object. Assuming no changes in the scene illumination, the following constancy equation holds:

$$I(\bar{x}, t_0) = I(f(\bar{x}, \bar{\mu}), t_n) \forall \bar{x} \in P, \quad (1)$$

where $I(\bar{x}, t_0)$ is the template image of P and $I(f(\bar{x}, \bar{\mu}), t_n)$ is the recti£ed image at time t_n, with motion model $f(\bar{x}, \bar{\mu})$ and motion parameters $\bar{\mu}$.

The motion parameter vector $\bar{\mu}$ can be estimated from equation (1) by minimising the difference between the template and the recti£ed image:

$$\min_{\bar{\mu}} ||\mathbf{I}(f(\bar{x}, \bar{\mu}), t_n) - \mathbf{I}(\bar{x}, t_0)||^2, \quad (2)$$

where $\mathbf{I}(\bar{x}, t)$ is a column vector constructed scanning P.

This minimisation problem can be solved linearly by computing $\bar{\mu}$ incrementally while tracking. We can achieve this by making a Taylor series expansion of (2) about ($\bar{\mu}$, t_n) and computing the increment, $\delta\mu$, between two time instants [7]:

$$\delta\bar{\mu} = -(M^\top M)^{-1} M^\top [\mathbf{I}(\bar{x}, \bar{\mu}_n) - \mathbf{I}(\bar{x}, \bar{\mu}_0)]$$

where M is the Jacobian matrix of the image and dependence of $\mathbf{I}$ on t has been dropped for convenience.

While tracking, matrix M must be recalculated in each frame, as it depends on $\bar{\mu}$. This is computationally expensive, as M is of dimension $N \times n$, being N is the number of template pixels and n the number of motion parameters. In the sequel we will factor M in order to simplify this computation.

M can be written as

$$M(\bar{\mu}) = \begin{pmatrix} \nabla_x I(\bar{x}_1, \bar{\mu}_0)^\top f_x(\bar{x}_1, \bar{\mu})^{-1} f_\mu(\bar{x}_1, \bar{\mu}) \\ \nabla_x I(\bar{x}_2, \bar{\mu}_0)^\top f_x(\bar{x}_2, \bar{\mu})^{-1} f_\mu(\bar{x}_2, \bar{\mu}) \\ \vdots \\ \nabla_x I(\bar{x}_N, \bar{\mu}_0)^\top f_x(\bar{x}_N, \bar{\mu})^{-1} f_\mu(\bar{x}_N, \bar{\mu}) \end{pmatrix}, \tag{3}$$

where $\nabla_x I$ is the template image gradient, f_x is the derivative of the motion model with respect to the pixel coordinates and f_μ is the derivative of the motion model with respect to the motion parameters.

Depending on the motion model, M may be factored into the product of two matrices [7],

$$M(\bar{\mu}) = \begin{pmatrix} \nabla_x I(\bar{x}_1, \bar{\mu}_0)^\top \Gamma(\bar{x}_1) \\ \nabla_x I(\bar{x}_2, \bar{\mu}_0)^\top \Gamma(\bar{x}_2) \\ \vdots \\ \nabla_x I(\bar{x}_N, \bar{\mu}_0)^\top \Gamma(\bar{x}_N) \end{pmatrix} \Sigma(\bar{\mu}) = M_0 \Sigma(\bar{\mu}) \tag{4}$$

a constant matrix M_0 of dimension $N \times k$ and a matrix Σ of dimension $k \times n$, that depends on $\bar{\mu}$. As M_0 can be precomputed, this factorisation reduces the on line computation to

$$\delta\bar{\mu} = -(\Sigma^\top M_0^\top M_0 \Sigma)^{-1} \Sigma^\top M_0^\top [\mathbf{I}(\bar{x}, \bar{\mu}_n) - \mathbf{I}(\bar{x}, \bar{\mu}_0)]. \tag{5}$$

Matrix M_0 is the Jacobian of the template image. It is our *a priori* knowledge about target structure, that is, how the grey level value of each pixel changes as the object moves. It represents the information provided by each template pixel to the tracking process. Note that we can not track any object, as in order to solve (4), a non singular $M_0^\top M_0$ matrix is needed.

3. PROJECTIVE MODEL FOR PLANAR TRACKING

In this section we are going to introduce a projective model of target motion. In order to do this, we need to obtain the Jacobian matrix decomposition that arises from this model.

Let $\bar{x} = (u, v)^\top$ and $\bar{x}_h = (r, s, t)^\top$ be respectively the Cartesian and Projective coordinates of an image pixel. They are related by:

$$\bar{x}_h = \begin{pmatrix} r \\ s \\ t \end{pmatrix} \rightarrow \bar{x} = \begin{pmatrix} r/t \\ s/t \end{pmatrix} = \begin{pmatrix} u \\ v \end{pmatrix}; \; t \neq 0.$$

The equation f that describes the motion of a planar region is then a 2D projective linear transformation,

$$f(\bar{x}_h, \bar{\mu}) = H\bar{x}_h = \begin{pmatrix} a & d & g \\ b & e & h \\ c & f & 1 \end{pmatrix} \begin{pmatrix} r \\ s \\ t \end{pmatrix},$$

where the motion parameters are $\bar{\mu} = (a, b, c, d, e, f, g, h)^\top$.

The Jacobian matrix decomposition of this motion model can be expresses in terms of the elements of equation (3):

$$\nabla_{x_h} I(\bar{x}_h, \bar{\mu}_0)^\top = \left(\frac{\partial I}{\partial u}, \frac{\partial I}{\partial v}, -\left(u\frac{\partial I}{\partial u} + v\frac{\partial I}{\partial v} \right) \right) \tag{6}$$

$$f_x(\bar{x}_h, \bar{\mu})^{-1} = H^{-1} \tag{7}$$

$$f_\mu(\bar{x}_h, \bar{\mu}) = \begin{pmatrix} r & 0 & 0 & s & 0 & 0 & t & 0 \\ 0 & r & 0 & 0 & s & 0 & 0 & t \\ 0 & 0 & r & 0 & 0 & s & 0 & 0 \end{pmatrix} \tag{8}$$

Introducing (6), (7) and (8) into (3) M can be factored according to (4):

$$f_x(\bar{x}, \bar{\mu})^{-1} f_\mu(\bar{x}, \bar{\mu}) =$$
$$H^{-1} \left(\begin{array}{ccc|c} & & & \\ rI_{3\times3} & | & sI_{3\times3} & | \quad \dfrac{tI_{2\times2}}{0_{1\times2}} \\ & & & \end{array} \right) =$$
$$\left(\begin{array}{ccc} rH^{-1} & | & sH^{-1} \end{array} \right. \left| \begin{array}{c} \\ \\ tH_{12}^{-1} \end{array} \right) = \Gamma(\bar{x}_h)\Sigma(\bar{\mu}),$$

where H_{12}^{-1} is the matrix composed with the first two columns of H^{-1}, $I_{q\times q}$ is the $q \times q$ identity matrix and

$$\Gamma(\bar{x}_h) = \begin{pmatrix} rI_{3\times3} & | & sI_{3\times3} & | & tI_{3\times3} \end{pmatrix},$$
$$\Sigma(\bar{\mu}) = \begin{pmatrix} H^{-1} & 0 & 0 \\ 0 & H^{-1} & 0 \\ 0 & 0 & H_{12}^{-1} \end{pmatrix}.$$

With this factorisation we can projectively track a planar patch with the computational cost of inverting a 8×8 matrix on each frame.

4. TEMPLATE PIXEL SELECTION

Only areas of high image contrast provide information about template motion (see Fig. 1, only the white pixels on the left image provide information for tracking). If in equation (5) we use all template pixels, most of the computational effort would be devoted non informative pixels.

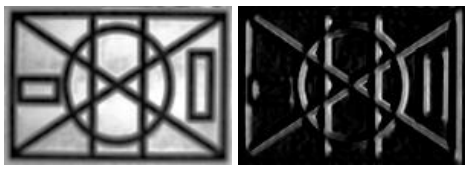

Fig. 1. Images of a template (left) and of $I(\bar{x}, \bar{\mu}_n) - I(\bar{x}, \bar{\mu}_0)$ (right), where parameter $\bar{\mu}$ represents a horizontal displacement.

In this section we will further improve the tracking procedure presented in the previous section by reducing the number of template pixels used for solving equation (5).

This improvement comes not only from having a smaller matrix M_0, but mainly from diminishing the number of pixels warped to compute $I(\bar{x}, \bar{\mu}_n)$.

The Jacobian matrix M of image I can be expressed as:

$$M = (I_{\mu_1}, I_{\mu_2}, \cdots, I_{\mu_n}),$$

where $I_{\mu_i} = \frac{\partial I(\bar{x},\bar{\mu})}{\partial \mu_i}$ is a column vector with an entry for every pixel in I. It represents the changes in image brightness induced by motion μ_i (see Fig. 2). Thus, M relates variations in motion parameters to variations in brightness values. Note that equation (5) works in the opposite direction, i.e. it uses M to compute motion from observed changes in brightness values.

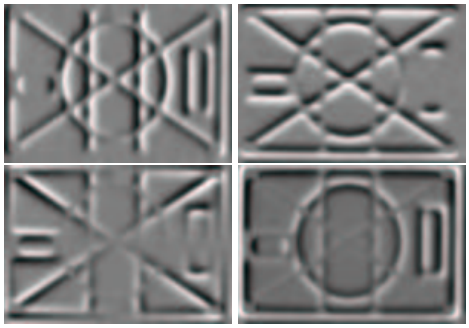

Fig. 2. Jacobian matrix for a translation (x, y), rotation (θ) and scale (s) motion model. In reading direction each image represents respectively I_x, I_y, I_θ, I_s.

Let us call $I_{\bar{\mu}}^{\top}(\bar{x})$ the row in M corresponding to image pixel $I(\bar{x})$. Each row entry is the derivative of image pixel $I(\bar{x})$ with respect to a model parameter μ_i $(\forall i = 1 \ldots n)$. Intuitively, a pixel with a small $||I_{\bar{\mu}}(\bar{x})||$ provides almost no information for solving (5). So, a good pixel for tracking is one with a large $||I_{\bar{\mu}}(\bar{x})||$. Given two image pixels $I(\bar{x}_1)$ and $I(\bar{x}_2)$, one of them is redundant if $I_{\bar{\mu}}(\bar{x}_1) \approx I_{\bar{\mu}}(\bar{x}_2)$. So, a good set of pixels for tracking is one such that $M^{\top}M$ is not singular.

Selecting the "best" set of m pixels is a combinatorial search problem, as all $\binom{m}{N}$ sets of pixels should be considered in order to select the most informative one. In the context of image registration, Dellaert selects m pixels randomly from the top 20% of pixels with highest $||I_{\bar{\mu}}(\bar{x})||$ [8]. In our experiments we have found that the best set of pixels for tracking is the one with highest $||I_{\bar{\mu}}(\bar{x})||$, lowest redundancy and most even distribution on the image. In the sequel we will present a procedure to select a set of pixels with high $||I_{\bar{\mu}}(\bar{x})||$ and low redundancy.

If we consider each row vector $I_{\bar{\mu}}(\bar{x})$ as a point in n-dimensional space, then the points in the convex hull of this cloud are those with highest $||I_{\bar{\mu}}(\bar{x})||$ and lowest redundancy. Let us call this set of points the *Jacobian cloud*. Computing the convex hull of a Jacobian cloud with thousands of points in a 8-dimensional space (the projective motion model has 8 parameters) can be time consuming. On the other hand, as can be seen in Fig. 3 (right), the distribution of points for this model is highly correlated, with two space directions representing 99.96% of the total variance in the cloud. So, a good approximation to the convex hull of the cloud would be to compute the convex hull of its projection onto the two main directions (see Fig. 3, left).

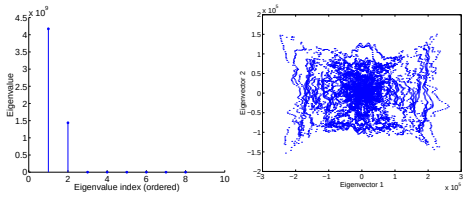

Fig. 3. Eigenvalues of the Jacobian cloud's covariance matrix (left) and view of the projection of the Jacobian cloud onto the two principal directions (right).

If we choose the points from the outer convex hulls (like peeling off an orange) then only the pixels in the strongest edges of the image would be selected. In order to achieve a more even spatial distribution of the selected pixels we choose all pixels of a randomly selected set of convex hulls from the outer 30% of them (see Fig. 4).

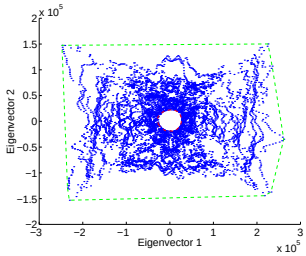

Fig. 4. Points in the outer 30% convex hulls in projected space.

5. EXPERIMENTS

We have conducted our experiments on a GNU/Linux system with an AMD K7 750MHz. We acquire images from a Sony VL500 digital camera with IEEE 1394 interface. In the experiments performed we track a template of 149×104 pixels, shown on the left in Fig. 1.

In the £rst experiment we study the gain in throughput achieved by tracking the template using only 695 pixels, instead of the 15.496 pixels of the full template. As can be

seen in Fig. 5, using pixel selection the system runs one order of magnitude faster.

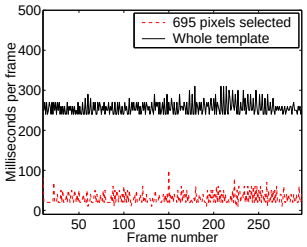

Fig. 5. Gain in system throughput achieved by using pixel selection.

Next we compare the performance of the pixel selection procedure presented in section 4 with Dellaert's method and with full frame tracking. In Fig. 6 are shown the plots of the RMS tracking residual for full frame tracking and for a tracker using 695 pixels selected with the two methods discussed in this paper.

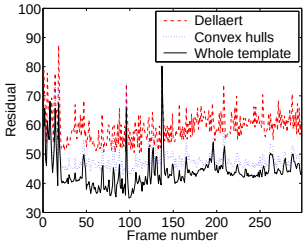

Fig. 6. Tracking residual for different pixel selection procedures.

In the last experiment (see Fig. 7) we study the evolution of the average frame tracking residual as the number of pixels used in tracking increases.

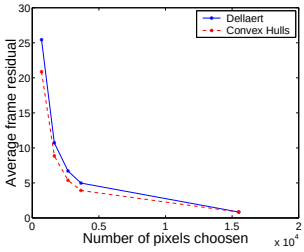

Fig. 7. Evolution of the average tracking residual for different numbers of selected pixels and different selection procedures.

These results show that by adequately selecting the pixels used in tracking, the amount of computation per frame can be reduced in one order of magnitude. The penalty that we pay for this improvement in processing time is an increase of about 20% in the tracking residual.

6. CONCLUSIONS

We have introduced a linear model for projectively tracking a planar patch and a procedure to speed up tracking by selecting only a special set of pixels from the tracked template. The pixel selection procedure introduced increases in one order of magnitude the speed at which frames are processed.

Being able to track a planar patch using a small set of pixels is important not only because of the increase in processing speed, but also because in this way we will be able to track regions of arbitrary shape.

In the present implementation we work at 18 frames per second for the full template tracker using Intel's IPL warping routines. The results shown in the experimental section were obtained using a software not optimised. We are in the process of writing MMX-optimised routines for warping a selected set of pixels.

The spatial distribution of selected pixels on the image is an interesting line of research. We think that the performance of the tracker can be improved by evenly distributing the selected pixels in the image. This issue is specially important in we want to consider tracking with partial template occlusions.

7. REFERENCES

[1] G. Simon, A. Fitzgibbon, and A. Zisserman, "Markerless tracking using planar structures in the scene," in *Proc. International Symposium on Augmented Reality*, October 2000.

[2] F. Lerasle V. Ayala, J.B. Hayet and M. Devy, "Visual localization of a mobile robot in indoor environments using planar landmarks," in *Proceedings Intelligent Robots and Systems, 2000*. IEEE, 2000, pp. 275–280.

[3] M. J. Black and Y. Yacoob, "Recognizing facial expressions in image sequences using local parameterized models of image motion," *Int. Journal of Computer Vision*, vol. 25, no. 1, pp. 23–48, 1997.

[4] C. Thorpe F. Dellaert and S. Thrun, "Super-resolved texture tracking of planar surface patches," in *Proceedings Intelligent Robots and Systems*. IEEE, 1998, pp. 197–203.

[5] M Irani and P. Anandan, "All about direct methods," in *Vision Algorithms: Theory and practice*, W. Triggs, A. Zisserman, and R. Szeliski, Eds. Springer-Verlag, 1999.

[6] P. H. S. Torr and A. Zisserman, "Feature based methods for structure and motion estimation," in *Vision Algorithms: Theory and practice*, W. Triggs, A. Zisserman, and R. Szeliski, Eds. Springer-Verlag, 1999, pp. 278–295.

[7] Gregory D. Hager and Peter N. Belhumeur, "Ef£-cient region tracking with parametric models of geometry and illumination," *IEEE Transactions on Pattern Analisys and Machine Intelligence*, vol. 20, no. 10, pp. 1025–1039, 1998.

[8] F. Dellaert and R. Collins, "Fast image-based tracking by selective pixel integration," in *ICCV99 Workshop on frame-rate applications*, 1999.