Learning Efficient Linear Predictors for Motion Estimation

Jiří Matas[1,2], Karel Zimmermann[1], Tomáš Svoboda[1], and Adrian Hilton[2]

[1] Center for Machine Perception Czech Technical University Prague, Czech Republic
[2] Centre for Vision, Speech and Signal Proc.
University of Surrey Guildford, England

Abstract. A novel object representation for tracking is proposed. The tracked object is represented as a constellation of spatially localised linear predictors which are learned on a single training image. In the learning stage, sets of pixels whose intensities allow for optimal least square predictions of the transformations are selected as a support of the linear predictor.

The approach comprises three contributions: learning object specific linear predictors, explicitly dealing with the predictor precision – computational complexity trade-off and selecting a view-specific set of predictors suitable for global object motion estimate. Robustness to occlusion is achieved by RANSAC procedure.

The learned tracker is very efficient, achieving frame rate generally higher than 30 frames per second despite the Matlab implementation.

1 Introduction

We formulate real-time object or camera tracking as establishing correspondence in a short-baseline pair of images followed by robust motion estimation. In real-time tracking, computation time together with the relative object-camera velocity determine the maximum displacement for which features must be matched. Local features (corners, edges, lines) or appearance templates have both been widely used to estimate narrow baseline correspondences [1,2,3].

Recently more discriminative features have been introduced to increase robustness to changes in viewpoint, illumination and partial occlusion allowing wide-baseline matching but their are too computationally expensive for tracking applications since their include combinatorial search [4,5,6,7].

We propose a novel object representation for tracking. The tracked object is represented as a constellation of spatially localised linear predictors. The predictors are learned using a set of transformed versions of a single training image. In a learning stage, sets of pixels whose intensities allow for optimal prediction of the transformations are selected as a support of the linear predictor.

The approach comprises three contributions: learning object specific linear predictors which allow optimal local motion estimation; explicitly defining the trade-off between linear predictor complexity (i.e. size of linear predictor support) and computational cost; and selecting an view-specific set of predictors suitable for global object motion estimate. We introduce a novel approach to

P. Kalra and S. Peleg (Eds.): ICVGIP 2006, LNCS 4338, pp. 445–456, 2006.

learn a linear predictor from a circular region around the reference point which gives the best local estimation, in the least square sense, of the object motion for a predefined range of object velocities. Spatial localisation robust to occlusions is obtained from predicted reference points motions by RANSAC. The approach makes explicit the trade-off between tracker complexity and frame-rate.

Tracking by detection [8,9] establishes the correspondences between distinguished regions [6,4] detected in successive images. This approach relies on the presence of strong, unique features allowing robust estimation of large motions by matching across wide-baseline views. Detection approaches also allow automatic initialisation and re-initialisation during tracking. Methods dependent on distinguished regions are not able to track fast, saccadic motions with acceptable accuracy due to their low frame-rate.

Displacement estimation methods achieve higher frame rates but are not able to reliably estimate large inter-frame motions. The methods usually utilise gradient-base optimization algorithms minimising a criteria function. The method of Lucas and Kanade [2,1] minimise color dissimilarity. They assume that total intensity difference (dissimilarity) is a convex function in some neighbourhood. Thus, the motion is estimated by a few iterations of the Newton-Raphson method, where the difference image is multiplied by the pseudo-inverse of the image gradient covariance matrix. Black and Jepson [10] robustifies this idea by tracking in eigen-space. Jepson et al. [11] extend tracking by online adaptation to non-rigid changes using EM-algorithm.

The method of Lucas and Kanade [2,1] was also extended by Cootes [12] and applied to tracking by Jurie [13,14] who learn a linear approximation of the relationship between the local dissimilarity image and displacement. Assuming that there exists a neighbourhood where displacement can be found directly from gradients of image intensities, online tracking is performed by multiplying the difference image by a matrix representing the linear function. This is computationally efficient because no gradient or pseudo-inversion are required. Recently this approach [15] has been extended to more general regression functions, where displacements are estimated by RVM. Such methods can learn a larger range of pose changes but tracking is more complex resulting in a lower frame-rate. Since we believe that the computational cost grows faster than the region of convergence, the main attention is focused on Jurie's [13] linear predictor. Although, all proposed methods are simply extendible to an arbitrary motion predictor.

The computation cost of tracking is a trade-off between the time required for displacement estimation and the distance moved between successive frames. Therefore, we propose a tracking method which explicitly models the trade-off between tracker complexity and frame-rate. Given the expected maximum velocity of the object we learn the optimal support of linear predictors for frame-rate tracking.

It is desirable to have efficient tracking and motion estimation to limit the object movement between successive estimates. In this paper, we extend the computationally efficient tracking using linear models of motion proposed by Jurie et al. [13], whose linear predictors use a support around pixels with high gradient values. Instead, our approach learns the support suitable for estimation of the linear motion model. Given a circular region around a reference point we learn the k best pixels to estimate the linear motion from synthesized training images with known motion giving the optimal linear predictor support. Selection of predictors, suitable for the global object motion is performed online. This approach tracks a view-specific set of reference points using the optimal supports for efficient tracking with a known relationship between maximum object-camera velocity, motion estimation accuracy and computation time.

The rest of the paper is organised as follows. Section 2 introduces learning of linear predictors templates and reference points set, respectively. Section 3 describes tracking and the optimal size of the template neighbourhood. Following Section 4 shows the experiments and the last Section 5 summarises the results and conclusions.

2 Motion Estimation

In this section we introduce a method for learning a linear predictor as well as a subset of a given size from a circular region around the reference point, which minimise a training error. This subset is called a *linear predictor support* and the size is called *complexity of linear predictor*.

The input to our system is a single image of the object to be tracked. This image is used to synthesise a set of training images under the motion model to be tracked. In this work we assume planar object surfaces giving a homography for object motion estimation. The local linear approximation of the motion model for each image neighbourhood allows more general non-planar surfaces and perspective projection. Combining particular motion of regions into a global motion estimate imposes constraints on the object surface shape and motion model. Section 2.1 presents the learning of object specific linear predictors for local motion estimation. Section 2.2 describes predictor complexity estimation optimal with respect to the maximum object velocity.

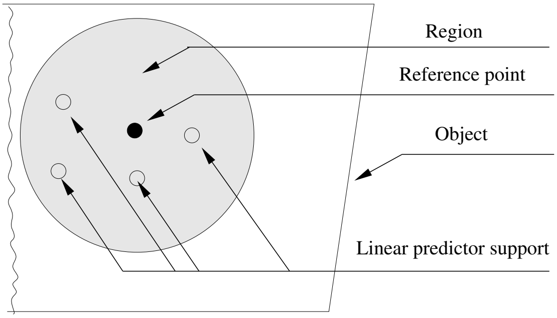

Fig. 1. Terminology: Reference point and a circular region around it. The linear predictor support is a learned set of pixels from the region.

2.1 Learning of Linear Predictors

In this section we present a method for learning a reference point specific linear predictor of a given complexity for estimation of the local motion. The set of pixels sampled in the predicted region is optimised to give the best k pixel predictor support for estimating the object motion using a linear approximation [1]. Optimisation is performed with respect to a set of synthesised training examples (i.e. perturbations) of the predicted region under known motion. The resulting subset gives efficient motion computation.

We are looking for a linear mapping $\mathtt{H} : \mathcal{R}^k \to \mathcal{R}^2$, from which we can estimate the displacement $\mathbf{t}$ (2-vector) from the difference $\mathbf{d}$ (k-vector) between the template and observation on the support,

$$\mathbf{t} = \mathtt{H}\mathbf{d}. \tag{1}$$

The ($2 \times k$ matrix) matrix $\mathtt{H}$ is estimated by the least squares method. A set of training examples are generated from a single input image by perturbing the observed object surface with random affine deformation. The range of possible affine deformations considered is given by the expected maximum relative velocity between the camera and object together with the camera frame-rate. Given m training examples, represented by $2 \times m$ matrix $\mathtt{T}$ and $k \times m$ matrix $\mathtt{D}$ such that columns are the corresponding pairs of displacements and intensity differences, the least-squares solution is:

$$\mathtt{H} = \mathtt{T}\mathtt{D}^{+} = \mathtt{T}\mathtt{D}^{\top}(\mathtt{D}\mathtt{D}^{\top})^{-1} \tag{2}$$

The supporting set need not include all the pixels from a predicted region. For example in uniform image areas pixels will add no additional information to the transformation estimation whereas pixels representing distinct features (edges, corners, texture) will be important for localisation. We therefore want to select the subset of k pixels for a predicted region which provides the best local estimate of the motion according to the linear model defined in equation (1). The quality of a given subset of the pixels can be measured by the residual error of the transform estimated on the training data:

$$e = \|\mathtt{H}\mathtt{D} - \mathtt{T}\|_F \tag{3}$$

For k pixels from the radius s we have $\binom{\pi s^2}{k}$ possible subsets of pixels. Explicit evaluation of the training error for all possible subsets is prohibitively expensive, we therefore subset is selected by randomised sampling.

The above analysis considers a single linear function $\mathtt{H}$ approximating the relationship between the observed image difference and object motion for a predicted region. This allows motion estimation upto a known residual error. For a given region of radius R the linear model gives an approximation error $r << R$

[1] Estimation of the optimal k with respect to the object maximum velocity is described in Section 2.2.

such that 95% of the estimated motions are within r of the known true value. Typically in this work $R \approx 20 - 30$ pixels and the resulting $r \approx 2 - 5$ pixels for a planar homography. Given a set of local motion estimates for different regions a robust estimate of the global object motion is obtained using RANSAC to eliminate the remaining 5% of outliers Section 3.

To increase the range across which we can reliably estimate the object motion we can approximate the non-linear relationship between image displacement and motion by a piece-wise linear approximation of increasing accuracy. For a given region we learn a series of linear functions $\mathtt{H}_0, \ldots, \mathtt{H}_q$ giving successive 95% approximation errors $r_0, \ldots, r_q$ where $r_0 > r_1 > \ldots > r_q$. This increases the maximum object velocity without a significant increase in computational cost.

2.2 Learning of Predictor Complexity

In this section we analyse the complexity of motion predictor in order to maximize framerate. To achieve real-time tracking, we generally want to utilise the observations at each frame to obtain a new estimate of the motion. This requires a trade-off between tracking complexity and estimation error due to object motion. Here we assume a maximum object velocity and optimise the motion estimation for tracking at frame-rate.

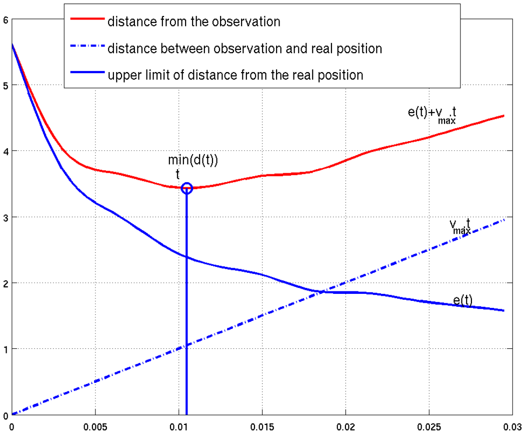

Fig. 2. Distance $d(t)$ from the real position of the object and its minimum

For a single linear predictor the error of displacement estimation decreases with its complexity (i.e. the number of pixels k selected from the predicted region). However, as k increases the error converges to a constant value with decreasing negative gradient. The error will only decrease when new structural information about the local variation in surface appearance is added. In uniform regions the variation is due to image noise and will not decrease localisation error. The computation cost increases linearly with the number of pixels used,

k. Therefore, we seek to define an optimal trade-off between computation time and motion estimation error.

Since the time needed for displacement estimation is a linear function of the number of pixels $t = ak$, the displacement error $e(t)$ is also a decreasing function of time. During the displacement estimation, the object moves away from the observation. The distance $d(t)$ from the real position of the object in the worst case is

$$d_{max}(t) = e(t) + v_{max}t, \tag{4}$$

where v_{max} is the maximum velocity of the object in pixels. Figure 2 shows the characteristic of the maximum distance and the motion estimation error $e(t)$ with increasing number of pixels k or time.

Assuming $\dot{e}(t) = \frac{de(t)}{dt}$ is a monotonically decreasing function, equation (4) has a unique solution given by:

$$t^* = \arg\min_t(d(t)) = \dot{e}^{-1}(-v_{max}) \tag{5}$$

The complexity of the tracker which minimises motion estimation error for real-time tracing is $k^* = \frac{t^*}{a}$. The worst expected accuracy error is $e(t^*) + v_{max}t^*$. Similarly, given the required accuracy, the maximum speed of the object could be estimated.

3 Tracking

Motion estimation for each individual prediction support requires a single matrix multiplication using equation (1). The cost of this operation is proportional to the number k of pixels in the regions. Matrix $\mathtt{H}$ is estimated offline in a pre-processing stage using synthesised training examples. Iterative refinement of the linear approximation using a hierarchy of q linear approximations $\mathtt{H}_0, ..., \mathtt{H}_q$ requires $\mathcal{O}(pkq)$ operations, where p is the number of regions and k is the predictor complexity.

Global motion estimation for a set of p regions is estimated using RANSAC to provide robustness to errors in local motion estimates and partial occlusion. In this work we assume planar object surfaces giving image motion defined by a homography with eight degrees-of-freedom. Once the motion of each region is estimated, we use 4-point RANSAC to filter out outliers and compute the correct motion of the object. Note, that this homography is applied to both the reference point positions and the supporting sets.

3.1 Active Region Set

Robust motion estimation in the presence of occlusion requires regions to be distributed across the object surface. It is not possible to find the set of regions suitable for object tracking independently on the object position, because if the object gets closer to the camera some regions can disappear and the global motion estimation can easily become ill-conditioned. In this section we present an online method which automatically selects a subset of p regions, called *active region set*, from all visible regions which provide the most accurate motion estimate and is sufficiently robust.

To optimise the distribution of regions across the surface, we define a coverage measure of the region reference points set X,

$$c(X) = \sum_{\mathbf{x} \in X} d(\mathbf{x}, X \setminus \mathbf{x}), \tag{6}$$

where distance between point $\mathbf{x}$ and set X is defined as the distance from the closest element of the set

$$d(\mathbf{x}, X) = \min_{\mathbf{y} \in X} \|\mathbf{x} - \mathbf{y}\|. \tag{7}$$

Ideally for optimal robustness to occlusion the coverage measure would be maximised. In practice, individual regions have an associated localisation error which must be taken into account. The quality $q(\mathbf{x})$ of individual regions is measured by their mean error $e(\mathbf{x})$ on the training data.

$$q(\mathbf{x}) = \max_{\mathbf{y} \in X} \big(e(\mathbf{y})\big) - e(\mathbf{x}). \tag{8}$$

To find a suitable subset X of regions from all visible regions $\overline{X}$ we seek to optimise the weighted combination of the coverage and quality:

$$f(X) = w\frac{c(X)}{c(\overline{X})} + (1 - w)\frac{q(X)}{q(\overline{X})}, \tag{9}$$

where $w \in [0; 1]$ is the coverage weight. Given the maximum number of regions p we search for the optimal set of regions using the greedy search strategy presented in Algorithm 1.

Figure 3 shows example results obtained for $w = 0, 0.5, and1$. In the case of $w = 0$ the p regions with the minimum error are selected resulting in clustering of regions in one part of the image. Conversely, $w = 1$ results in regions spread across the object with some having a relatively high motion estimation error. Intermediate values of w result in a compromise between region distribution and quality.

```
1. Let $\overline{X}$ be the set of possible regions and $X = \emptyset$ a subset of selected
   regions.
2. Select $\mathbf{x}^* \in \overline{X}$ holds $\mathbf{x}^* = \arg\max_{\mathbf{x} \in \overline{X} \setminus X} f(\mathbf{x} \cup X)$
3. $X = \mathbf{x}^* \cup X$ and $\overline{X} = \overline{X} \setminus \mathbf{x}^*$
4. if $|X| = p$ end, else goto 2
```

Algorithm 1. Active region set estimation

4 Experiments

The proposed method was tested on several different sequences of planar objects. We demonstrate robustness to large scaling and strong occlusions as well as

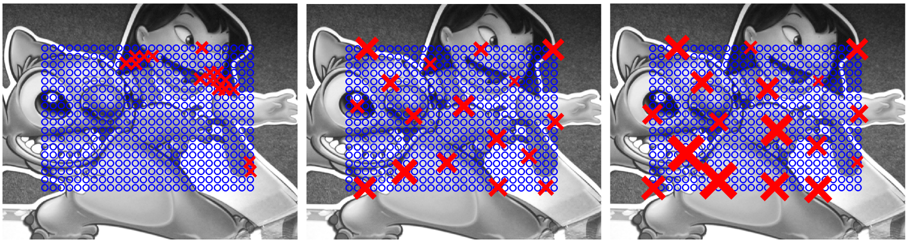

Fig. 3. Object coverage by regions for $w = 0, 0.5, 1$. Blue circles correspond to the all possible regions, red crosses to the selected regions. Size of crosses corresponds to the training error.

saccadic motions (e.g. like shaking), where object motion is faster than 30 pixels per frame. Section 4.1 investigates region suitability and influence of the coverage weight. We show that even the regions which are strong features, in the sense of Shi and Kanade [3] definition, may not be suitable for tracking. Section 4.2 summaries advantages and drawbacks of methods for linear predictor support estimation and Section 4.3 shows real experiments and discuss very low time complexity.

4.1 Active Region Set Estimation

In this experiment, we show influence of coverage weight on active region set and discuss region suitability for tracking. Different region sets selected for different weights are shown at Figure 3. The set of all possible regions is depicted by blue circles. Active region set of the most suitable 17 regions is labeled by red crosses, where size of the cross corresponds to the training error of the particular region. The weight defines the compromise between coverage and quality of the regions. The higher is the weight, the more uniform is the object coverage.

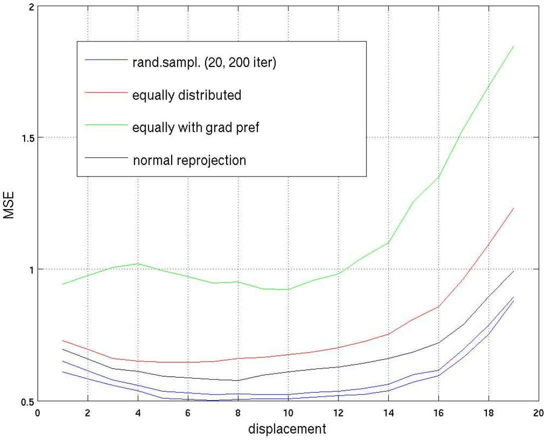

Fig. 4. Comparison of different methods for linear predictor support estimation

In the last case ($w = 1$), we can see that the teeth provide very high tracking error, although they are one of the strongest features due to the high values of gradient in their neighbourhood. The repetitive structure of teeth causes that different displacements correspond to the almost same observations. If the range of displacement had been smaller than teeth period, the training error would have been probably significantly smaller. In this sense, region quality is depends on the expected object velocity (or machine performance).

4.2 Comparison of Different Methods for Linear Predictor Support Estimation

In this experiment we compare several different methods for linear predictor support selection. The experiment was conducted on approximately 100 regions.

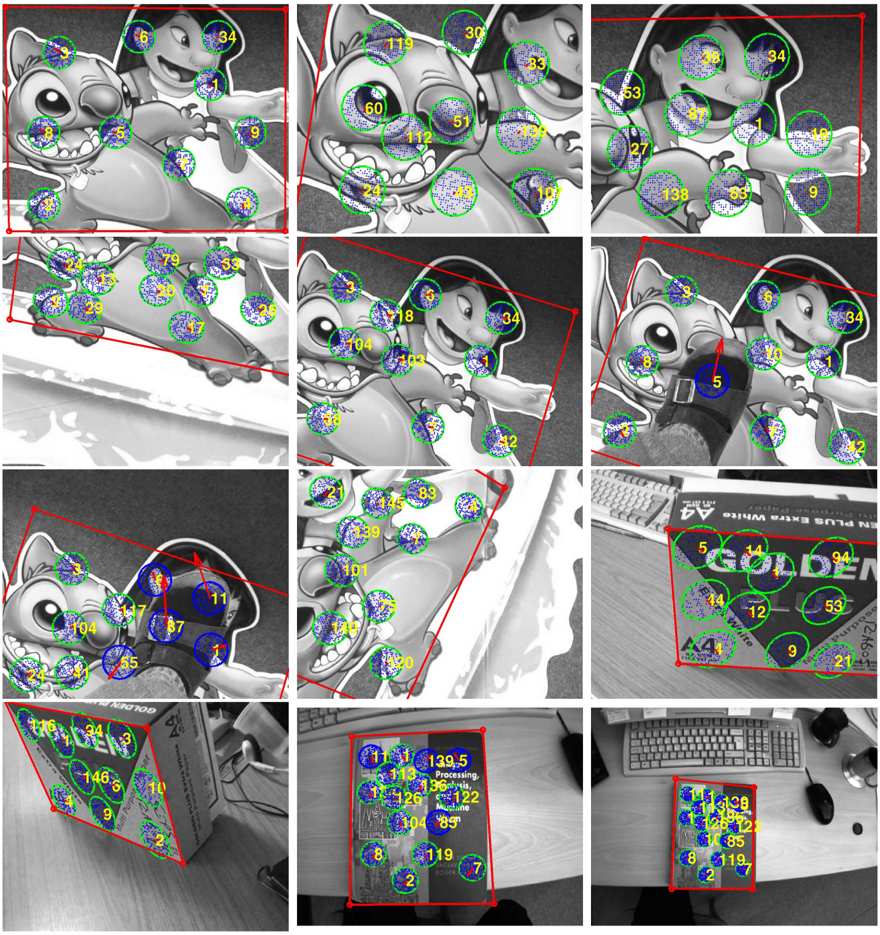

Fig. 5. Different sequences: Blue points represent support set, green circles highlight inliers, red arrows outline particular motion

From each region of 30-pixel radius a subset of 63 pixels was selected supporting by different methods.

Figure 4 compares average errors of tracking on artificial testing examples for different ranges of displacements of the following methods:

- Equally distributed pixels over the region - the support consists of pixels lying on a regular grid.
- Equally distributed with gradient based selection - pixels are divided into the grid-bins. The pixels with the highest gradient from each bin forms the support.
- Normal re-projection - First the least square solution is found for the whole n-pixel region. Each row of the obtained matrix $\mathtt{H}$ corresponds to the normal vector of n-dimensional hyper-plane. Particular components provide an information about pixel significance. The pixels corresponding to the highest components are utilised.
- Randomised sampling - Random subsets are repetitively selected from the region. Those which provide the lowest training error are utilised..

Since the global minimum seems for practical region sizes unreachble, it is necessary to use a heuristic method. Randomized sampling seems as the best choice, because even as few as 20 iterations provide very good results. The more iterations is performed, the closer to the global minimum we can get. In the other hand, randomised sampling requires as many estimation of least square problem

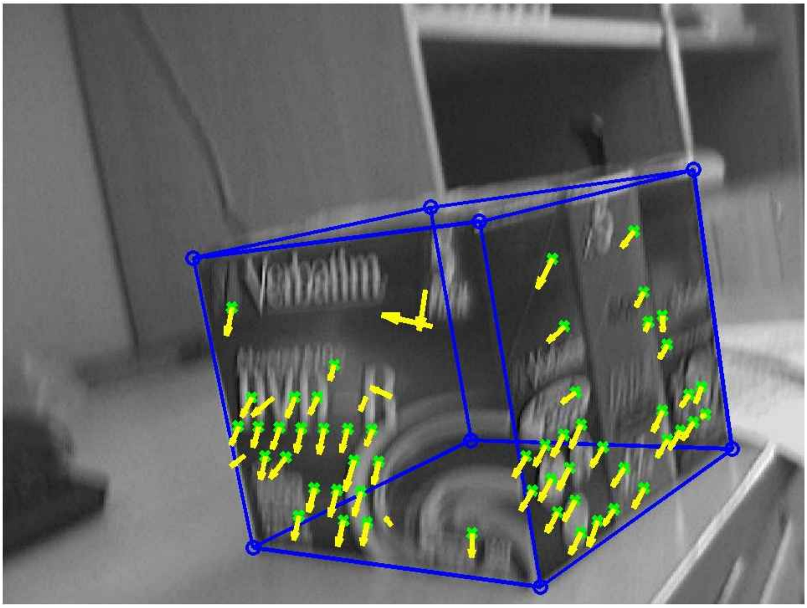

Fig. 6. 3D tracking: Robustness to motion blur achieved by learning

as iterations. If someone looks for a fast heuristic (e.g. for online learning) then normal re-projection method is a natural compromise.

4.3 Tracking

Figure 5 shows tracking of different planar objects including views from the acute angles, partial occlusion, shaking and large range of scales [2]. Figure 6 shows 3D tracking and robustness to the motion blur due to assured in learning stage.

Our slightly optimized matlab implementation runs at 30−140 frames/second. The frame-rate is mainly dependent on the number of tracked regions and the sizes of their complexity. Time required for the particular motion estimation, pose estimation and the active region set selection is approximately the same.

5 Conclusions

We proposed a very efficient tracking method based on linear predictors of displacement. The predictors, learned from a randomly perturbed sample image, predict displacement of reference points from image intensities. The set of predictors changes during the tracking depending on the object pose. The dynamic selection makes the procedure robust against occlusions. The achieved frame rate depends on the object complexity, and it is generally higher than 30 frames per second despite the Matlab implementation.

Perhaps surprisingly, the reference points of the predictors do not often correspond to classical feature points which are mostly anchored at points with high gradient. The strength of method lies in the learning stage. The predictors are learned from the expected maximum velocity. The predictors are linear but strong enough to cover wide range of motions. The linearity allows for efficient learning.

Acknowledgement

Jiří Matas ackowledges support of Grant Agency of Czech Republic Project 201/06/1821 and EC project FP6-IST-004176 COSPAL. Karel Zimmermann and Tomáš Svoboda ackonwledge support of Czech Academy of Sciences Project 1ET101210407. Adrian Hilton acknowledges support EPSRC Platform Grant GR/S46543 at the Centre for Vision, Speech and Signal Processing, University of Surrey and by the EC project FP6-IST-004176 COSPAL.

References

1. Baker, S., Matthews, I.: Lucas-kanade 20 years on: A unifying framework. International Journal of Computer Vision **56** (2004) 221–255
2. Lucas, B., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: IJCAI. (1981) 674–679

[2] We encourage readers to look at the additional material for whole sequences.

3. Shi, J., Tomasi, C.: Good features to track. In: Computer Vision and Pattern Recognition (CVPR'94). (1994) 593 – 600
4. Lowe, D.: Object recognition from local scale-invariant features. In: International Conference on Computer Vision. (1999) 1150–1157
5. Lowe, D.: Distinctive image features from scale-invariant keypoints. IJCV **60** (2004) 91-110.
6. Matas, J., Chum, O., Urban, M., Pajdla, T.: Robust wide-baseline stereo from maximally stable extremal regions. Image and Vision Computing **22** (2004) 761–767
7. Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., van Gool, L.: A comparison of affine region detectors. IJCV **65** (2005) 43–72
8. Gordon, I., Lowe, D.: Scene modelling, recognition and tracking with invariant image features. In: International Symposium on Mixed and Augmented Reality (ISMAR). (2004) 110–119
9. Lepetit, V., Lagger, P., Fua, P.: Randomized trees for real-time keypoint recognition. In: Computer Vision and Pattern Recognition. (2005) 775–781
10. Black, M.J., Jepson, A.D.: Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. In: European Conference on Computer Vision. (1996) 329–342
11. Jepson, A.D., Fleet, D.J., El-Maraghi, T.F.: Robust online appearance models for visual tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence **25** (2003) 1296–1311
12. Cootes, T., Edwards, G., Taylor, C.: Active appearance models. PAMI **23** (2001) 681–685
13. Jurie, F., Dhome, M.: Real time robust template matching. In: British Machine Vision Conference. (2002) 123–131
14. Masson, L., Dhome, M., Jurie, F.: Robust real time tracking of 3d objects. In: International Conference on Pattern Recognition. (2004)
15. Williams, O., Blake, A., Cipolla, R.: Sparse bayesian learning for efficient visual tracking. Pattern Analysis and Machine Intelligence **27** (2005) 1292–1304