Pose refinement of active models using forces in 3D

A D Worrall, G D Sullivan & K D Baker

Dept of Computer Science, University of Reading,
Reading, UK. RG6 2AY

Abstract A new algorithm is described for refining the pose of a model of a rigid object, to conform more accurately to the image structure. Elemental 3D forces are considered to act on the model. These are derived from directional derivatives of the image local to the projected model features. The convergence properties of the algorithm is investigated and compared to a previous technique. Its use in a video sequence of a cluttered outdoor traffic scene is also illustrated and assessed.

1 Introduction

We report a new approach to the problem of recovering an accurate estimate of the pose of a rigid object, given an initial coarse estimate. The method uses an "active" model of the object, the pose of which is successively updated according to "forces" derived by examining local peaks of directional derivatives of the image, under the control of the current estimate. Unlike previous methods using active models [5, 8, 10, 16], we derive a set of elemental forces acting in 3D (rather than in the 2D image). These elemental forces can readily be resolved in the object coordinate frame, so that any object-centred constraints on possible movements are easy to impose. In particular, rigidity constraints are maintained automatically, and environmental constraints restricting the freedom of movement of the object can be imposed directly.

The algorithm was developed as an alternative to the existing "passive" method used in the recently completed VIEWS (Esprit P2157) system for the visual interpretation of traffic scenes. Its main role in the system is to update the position and orientation of models of vehicles in video sequences of images. With each image, we assume an initial estimate of pose and vehicle type; this may be the result either of initial analysis of lines detected by knowledge-free processing of the image [19], or of a kinematic filter based on the prior tracking history [17]. Using update rates of about 5Hz, we typically find that the pose of a tracked vehicle can be predicted to within an accuracy of ± 0.5m and $\pm 10°$. The pose-refinement algorithm seeks to obtain the position of the model which best explains the image data, local to the initial pose estimate. This becomes an observation which is fed into the kinematic filter, and the process iterates.

2 The VIEWS system

The pose refinement stage of the existing VIEWS system uses a potential-maximization method first described in [3] (see also [4]). A scalar "evaluation score" for an object pose is defined, based on the local strengths of image derivatives predicted by the model

Lecture Notes in Computer Science, Vol. 800
Jan-Olof Eklundh (Ed.)
Computer Vision – ECCV '94

lines (see [17] or [1] for recent overviews). A local search is then carried out in the configuration space of the pose to maximize the score. A number of search algorithms have been investigated, with the Simplex algorithm [15] providing a good compromise between efficiency and accuracy. Though model-based, this scheme is "passive", in that the value at a single pose gives no indication of the movement of the model most likely to improve the score.

A major advantage of the VIEWS system is that, once started, it allows objects to be tracked through time in an entirely "top-down" way, using purely local image evidence - *no* knowledge-free early processing is required. This allows computational resources to be devoted entirely to the evolving perceptual interpretation, and completely obviates the need to search for higher level image features (such as straight lines, corners, etc.). This paradigm is retained in the present work, but here the model is "active". Instead of passively defining a potential function to be optimised, each pose allows elemental forces to be computed and aggregated over the whole model. These then actively pull the model towards a better fit with the image. The forces we derive act in 3D on the rigid model, and can therefore be aggregated in a way that is independent of the perspective transformation used to obtain the image.

The 3D forces also allow object-centred constraints (such as the groundplane constraint - GPC [17, 18]) to be included naturally and easily. In the passive method, the GPC was exploited by restricting the search of pose space to the 3 degrees of freedom of a vehicle on a known road surface (x,y on the ground and θ, the rotation of the model about an axis normal to the ground). In the active method we simply resolve the elemental forces along the three freedoms, to give forces in the directions of x and y, and a torque about the vertical axis (θ), and ignore any other residual forces.

This paper gives details of the implementation, and comments on the main parameters of the method. An experimental study is then reported which shows that the new algorithm greatly improves the performance of the VIEWS system - in particular, it locates the pose more precisely and it converges successfully more often from errors introduced arbitrarily in the initial pose estimate.

An important advantage of the active method (over the passive method) is that fewer iterations are needed. Since the computational complexity is comparable, this is likely to lead to a significant improvement in real-time performance, though the system has not yet been implemented with sufficient attention to efficiency to assess and compare final performance.

The method has similarities with other recent work using active models, but differs in many essential respects. A comparison with previous work is deferred to the Discussion section.

3 Method

The central idea of the new algorithm is conveniently explained in terms of springs acting on smooth rods. One rod represents a linear feature of the model, the other represents the ray from the centre of projection to an edge feature found in the image near the projected model. The two rods are attached by a zero-length spring. Since the rods

are smooth, the spring slides along the rods to line up with the mutual perpendicular between the two.

Given an initial estimate of the pose of a rigid object, we search the image for evidence of nearby edges. The object is modelled by a polyhedral facet model, which allows all visible faces to be computed, together with the projections into the image of the boundary lines. Facet boundary lines are commonly associated with discontinuities of grey-level in the image, in a direction approximately perpendicular to the line. To find local evidence about the pose, we simply search a short direction along the normals to each projected boundary line and compute the image derivative in that direction.

In our earlier "passive" system, an evaluation score for a single line was given by the strength of the derivative at, or close to, the corresponding line in the image. This was then converted into an estimate of the probability that such a score would arise by chance (using empirical probability tables derived off-line). The probability scores of all visible boundary lines were then pooled to derive a single scalar representing the "goodness-of-fit" in the image of the model in that pose.

The new scheme uses a similar approach, but sets up elemental forces which act on each (3D) boundary line according to maxima of the image derivatives close to the projected line. In short, we allow a sharp discontinuity in the image (an image edge point) to "pull" any model line which projects near to it in the image. The key novelty in our approach concerns how such forces should act. Because of the perspective projection, an image edge point provides NO information about movement of the model line along the ray at that point. Likewise, a force acting on a model line (in 3D) should have NO action along the model line. We therefore consider the force to act along the mutual perpendicular between the model line and the ray to the image edge point - hence the springs and smooth rods analogy.

A number of schemes for determining the elemental forces have been tried. The simplest, illustrated here, is to weight the image derivative along the normals by a triangle function falling to zero at a fixed distance from the projected model line (this restricts consideration to image evidence near the projected model line). We then identify the strongest weighted derivative, and the ray through this point becomes one of the rods. This rod carries a spring of zero length, which is connected to the (3D) model line. Simple geometry is then used to determine the mutual perpendicular between the rods, so that the elemental force on the model due to the spring is specified in 3D.[1]

A number of alternative methods for deriving the elemental forces have been examined, including weighting the springs by the strength of the image derivative, or aggregating all derivatives along the normal to account for distributed forces. However, these are more costly, and have not yet proved to have any advantage in practice.

A set of elemental forces acting on the model are computed, by sampling along the projections of all boundary lines of the model (in the given pose). Each force is resolved along the three degrees of freedom allowed under the GPC, and aggregated to give linear forces in the x & y directions of the model-centred coordinate system, and a torque

1. An improved algorithm has been implemented that pre-computes the direction of the springs attached to each model line (assuming weak perspective), so that image derivatives can be computed in the required direction and pooled more efficiently. Space precludes a fuller description.

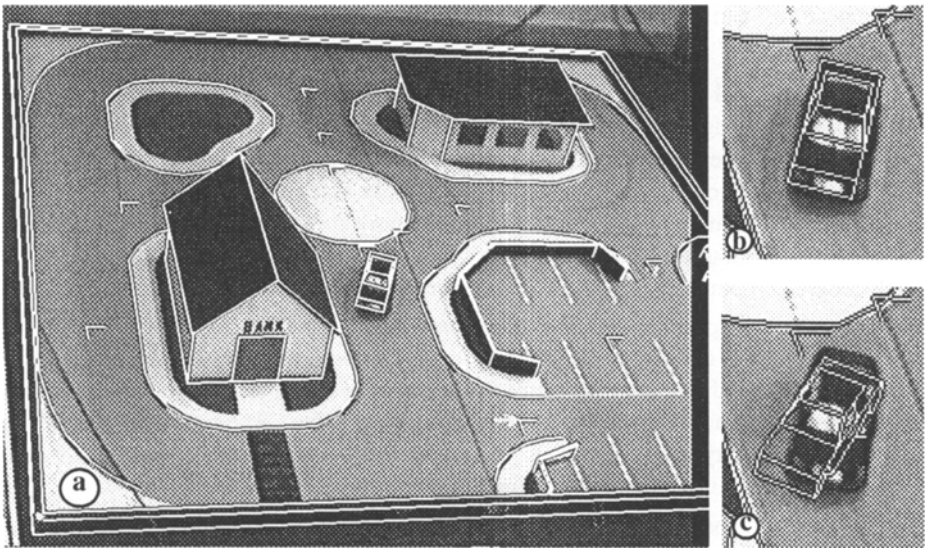

Fig. 1. (a) Image of toy test scene, (b) Close up of the car, with the "true" pose superimposed, (c) Starting pose (here displaced from (b) by 0.5m, 0.5m, 12.5°).

about the vertical. With suitable values chosen for mass, inertia, spring constant, and time step (see below), we allow the forces to displace the model from a stationary state to a new pose, and the process is iterated until some stopping condition is met.

4 Performance

The method is illustrated in Figure 1, which shows an 768*576 pixel image of the toy traffic scene used as a test-bed in VIEWS. We concentrate on the car near the centre, which subtends approximately 50*70 pixels.

A convenient way of assessing the performance of a pose refinement algorithm is to determine which poses become attracted to the correct pose [1]. For a given image of a vehicle we first identify by hand the "true" pose of the model, and this acts as the centre of coordinates in the configuration space (Figure 1(b)). We then perturb the pose by given amounts in the model configuration space (x, y and θ) to form a starting pose (e.g. Figure 1(c)).

The algorithm then runs for a number of iterations and the resulting pose is determined. Figure 2 shows typical results. The configuration space was sampled regularly around the origin at eleven points (per parameter), giving a total of 1331 starting poses, spaced at 200mm and 5°. The total range of starting poses therefore spanned ±1m and ±25° relative to the "true" pose.

Poses are shown in Figure 2 as short vectors ("needles"), whose positions correspond to the (x,y) values of the origin of the object coordinate frame, and whose directions correspond to the orientation of the model (θ). To assist visualization, an overhead view of the model in the "true" pose is shown superimposed on the map of starting poses in Figure 2.

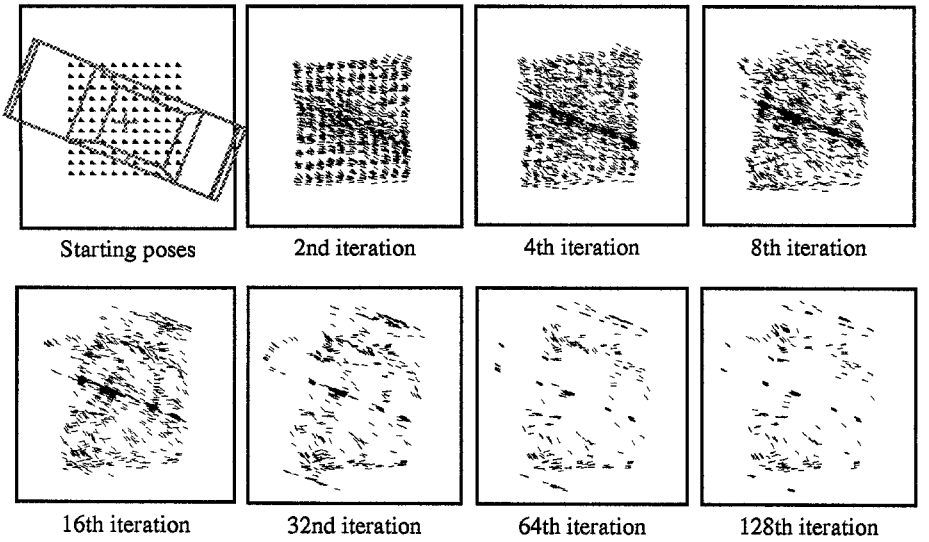

Fig. 2. Starting poses (top left) with the "true" pose superimposed, and poses obtained following successive iterations. Each box represents 4m*4m on the ground.

As the iterations proceed we see a rapid attraction of the needles towards the true pose (centred in the Figures and oriented appropriately). Note that identical poses overwrite each other in these needle Figures, so they only appear to occur once. After as few as 16 iterations, very few stable poses remain, and nearly all activity has ceased by the 32nd iteration. The great majority of trials converge correctly, although some small attractors exist away from the "true" pose. These usually correspond to obvious aliases, such as where the top of the windscreen of the model aligns with the bottom of the windscreen in the image.

However, these local traps are very small, and usually account for very few of the results. This is made clearer in Figure 3, which show data from the same experiment represented in a different way. Here the poses are collected into a histograms in (x,y), and orientation is ignored. The starting poses are equally distributed within a 2m*2m square; they rapidly converge to distinct positions, comprising the "true" pose and a few aliases (the main ones of which correspond to confusions between the horizontal structures at the front of the car).

5 Control parameters of the algorithm

There are several free parameters of the algorithm, which greatly affect performance. The experiment shown in Figures 2 and 3 used values set by informal experiment. This section considers the effects of changes from these values.

5.1 Dynamics

The major factor in the dynamic behaviour of the system is the relationship between the spring constant (K), the "mass" of the model (M) and the time step used between

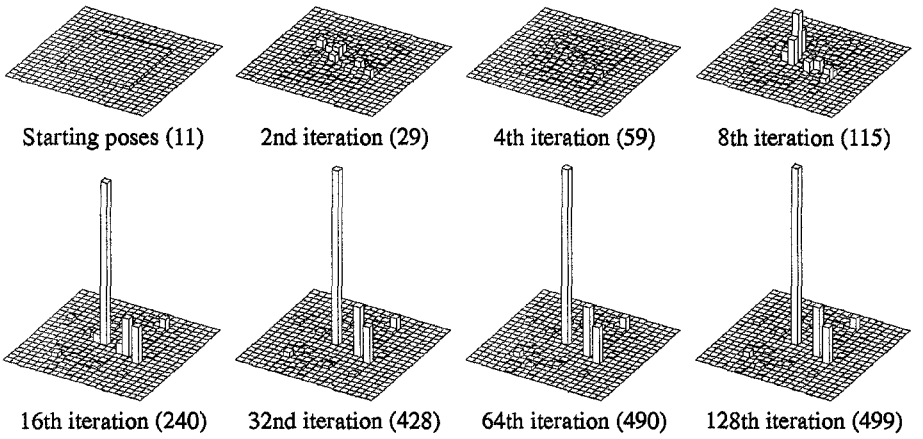

Fig. 3. Alternative representation of the data of Figure 2, showing the number of poses falling into bins centred at the starting poses. The number in the peak is shown as (99).

iterations (δt); these influence the system as:

$$\underline{x}_{i+1} = \underline{x}_i + (K/M)\,\delta t^2 \underline{F}_i$$

A similar expression involving "moments of inertia" is involved in the computation of rotations. To explore the dependency on these terms, it is therefore sufficient to examine the effect of δt alone.

Increasing δt by a factor of 2 the convergence is initially faster, but the final results are less precise. The time-step is too large; the system therefore overshoots and then oscillates around the main attractors. Decreasing δt by a factor of 2 the convergence is very much slower, and iteration 125 seems comparable to iteration 16 of Figure 2. There may also be some increased sensitivity to small aliases.

We conclude that the system is fairly sensitive to the choice of time step, but that our default values perform reasonably well in this case.

5.2 Evidence Range

A second major parameter affecting the algorithm is the lengths of the normals considered in searching for local evidence. We use as default (Figure 2) normal lengths of ± 4 pixels, giving a total of 7 image derivatives, centred on the projected model line. (Note: the image differentiation was carried out with a difference interval of 1 pixel, using bi-linear interpolation of grey values).

We have investigated the effect of increasing and decreasing the length of the normals by a factor of 2. Since the spring lengths will now vary (on average) by the same factor and this will affect convergence rates, the spring constants were also changed to compensate. Performance is fairly robust to such changes, though larger normal lengths lead to fewer, stronger attractors - this may or may not be desirable, depending on the application. It is interesting to note that the pattern of aliases changes, and also that the attractors appear more diffuse for larger normals probably indicating oscillation.

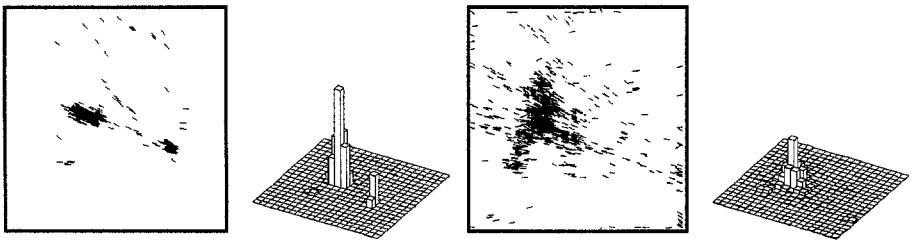

Fig. 4. Comparison between Active (left) and Passive (right) methods. using 30 iterations or simplex search (see text).

We conclude that the system is relatively insensitive to the search range, and that the default values are reasonable.

6 Comparison between Passive and Active methods

The algorithm was developed as a direct replacement for the passive system for model-based pose-refinement used in the VIEWS traffic tracking system (see e.g. [1,17]). In this use we need also to supply a stopping criterion for the search algorithm. With the passive search, using the Simplex algorithm, we terminate when either: (i) the 4 positions considered at one time (1 + the number of dimensions of the configuration space) differ by less than some threshold value, or (ii) after 100 iterations. In either case the resulting pose is that with the highest evaluation score found in the search.

The choice of stopping criterion in the active algorithm is less obvious. We have explored continuing until the movement in a single time step is smaller than some threshold, but this fails if the pose oscillates, and also may fail prematurely if convergence starts off slowly. In the current implementation it is a very small additional cost to evaluated the vehicle at each new pose, and this suggests an alternative approach. We refine the pose for 30 iterations of the active algorithm, but then use as our result the pose having the highest evaluation score encountered in the search - usually this occurs very near the end of the search.

Informal experiments have shown that the new algorithm appears to perform at least as well as the old. Figure 4 shows the two methods applied to the image of Figure 1, using a similar experimental technique as that in the previous experiments, but with the 11*11*11 search space spanning ±0.5m, ±0.5m and +/12.5°. Figure 4 is therefore drawn with twice the resolution. Note that performance in Figure 4 (left) differs from (say) Figure 2 (32nd iteration), because here the pose with the best evaluation score encountered in the search is retained, whereas before we showed the final pose.

In comparison to the new algorithm, the passive technique shows far more uncertainty in the results. It is likely that this is due to the fact that the evaluation function has a fairly flat plateau near its peak, so that noise in the system leads to spurious local peaks. This fact would also account for the increased spread observed in the results for the active system (Figure 4, left), since here we show the pose with the best evaluation score. One oddity is the conspicuous alias in the active results, which is not present in the passive system. This remains to be investigated.

Fig. 5. Three sub-images from test sequence (from frames numbered 5, 64 and 85).

To compare performance between the two algorithms in a more complex outdoor scene, we took one particular sequence from a video of a cluttered traffic scene that has previously caused difficulties for the passive system. These problems have been ascribed to two main causes: (i) the model used is an indifferent fit to the vehicle, and (ii) several items of street furniture obscure the vehicle and provide spurious image detail which may disrupt tracking.

The results are illustrated in Figure 5, which shows three sub-images from the sequence, as the vehicle, initially stationary as in Figure 5 (left) moves from extreme left in the image to extreme right. A model was instantiated in the first image of the sequence approximately, by eye, and the pose was refined as described above. This pose became the starting pose for a search in the next image in the sequence (taken at 25Hz). After 8 images were treated in this way (during which the vehicle was stationary), a Kalman filter was invoked to model the kinematics of the car, and this generated new predictions in each remaining image. Three examples are shown in Figure 5 corresponding to frames 5, 64 and 85 after the first. Tracking using active models seems excellent.

To compare performance of the active and passive algorithms, we focus on the middle image in Figure 5. The pose recovered during tracking was regarded as the "true" pose, and exhaustive perturbation trials were carried out, as before, using errors up to ± 0.5m, ± 0.5m and $\pm 12.5°$. Figure 6 shows the results obtained for both algorithms, represented as in Figure 4. The active algorithm performs far better: many more cases converge to the "true" pose, and there are fewer aliases. Furthermore, the active algorithm also required fewer iterations than the simplex algorithm, which typically used between 50 and 100 iterations (compared with the 30 always used in the active case).

7 Discussion

We have described a new approach to pose-refinement in model-based object recognition and tracking. The use of "active models" appears to provide a great improvement on our existing method which has used "passive models", though we have yet to optimise our implementation to compare the real-time performance of the two.

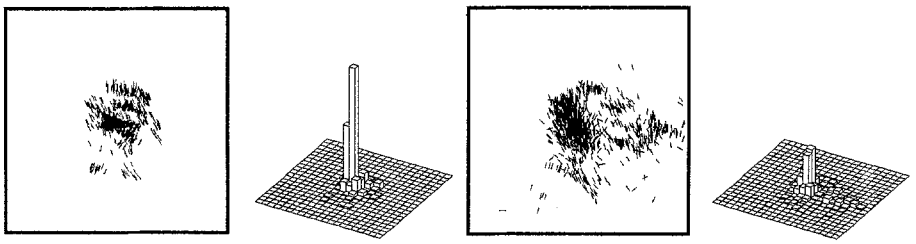

Fig. 6. Comparison between Active (left) and Passive (right) methods for the real traffic scene of Figure 5 using 30 iterations or simplex search (see text).
As Figure 4, but the boxes show 2m*2m on the ground.

7.1 Comparison with previous work on active models

The method owes much to previous work on active models. Kass et al. [11] first used densely sampled active models ("snakes") in which control points are attracted towards image gradients. However, without global structural constraints it is difficult to prevent clusters of control points becoming caught on local image detail.

Several authors have developed methods based on local splines which impose local image smoothness and improve performance considerably [5, 7, 9]. These schemes still lack 3D knowledge of the object's form and consider the forces to act in the image plane. This makes it necessary to solve non-linear equations to discover object rotations. Recent variants of the methods have imposed a limited 3D rigidity by tracking affine-invariant structure [2], but this cannot cope with self-occlusion by the model, and has not been demonstrated in complex images.

Also recently, Taylor and co-workers [6, 9] have developed a method for defining deformable objects (such as faces or hands) in terms of the principal components of data sampled by eye. The active search technique then generates forces which displace the model in its PCA configuration space. Once again though, the forces are computed in the image plane, and any 3D geometrical regularity must be captured implicitly by the PCA representation.

Other workers have explicitly considered rigid objects. Lowe [13, 14] first demonstrated methods for inverting the perspective transformation, using a linearisation technique, to minimise observed errors in the image between predicted and observed lines. Worrall [20] showed that the scheme could be represented as a minimisation of 3D errors, and this parameterisation allows object-based constraints to be represented more easily. The important distinction between these techniques and "active models" is that they rely on matching extended linear features, and this requires (i) a prior stage of feature analysis, and (ii) determination of image-to-object correspondences. The former is computationally expensive, and the latter is unreliable in natural scenes; however, the method has been used successfully to track vehicles in traffic scenes [12].

Stevens [16] used purely "top-down" methods to find image evidence local to predicted features, and pooled the error signals using an iterative form of the Hough transform to identify 3D movements which would minimise image displacement errors. Harris [10] used a very similar scheme, but updated the pose by linearising the rotation

equations around the current pose, and used linear-least-squares methods to solve the over-determined perspective inversion problem (see also [20]). Both methods only considered first order terms in the rotation equations, and as with flexible models, both sought to minimise image-based errors of a sparse set of control points.

The new technique differs from previous methods by the way in which the elemental forces computed from measurements in the 2D image are deemed to act in the 3D object space - using the smooth rods and springs analogy (see Methods). This has two important consequences: (i) the resultant effects of all elemental forces can be computed by using conventional mechanics, and this takes full account of the rigidity of the object, and (ii) forces (and torques) can easily be resolved along the axes of the configuration space to take into account any environmental constraint (such as the ground-plane constraint).

References

1. K.D. Baker, G.D. Sullivan: Performance Assessment of Model-based Tracking. IEEE workshop on Applications of Computer Vision, California, 28-35 (1992).
2. A. Blake, R. Curwen, A. Zisserman: Affine-invariant contour tracking with automatic control of spatiotemporal scale. Proc 4th ICCV, 66-75 (1993)
3. K.S. Brisdon , G.D. Sullivan, K.D. Baker: Feature Aggregation in Iconic Model Matching. Proc Alvey Vision Conference, AVC-88, Manchester, 19-24 (1988).
4. K.S. Brisdon: Hypothesis Verification using Iconic Matching. PhD Thesis, The University of Reading, 1990.
5. T.F. Cootes, C.J. Taylor: Active Shape Models - Smart Snakes. Proc BMVC-92 Springer-Verlag, 266-275 (1992).
6. T.F. Cootes, C .J .Taylor, A. Lanitis, D.H. Cooper, J. Graham: Building and Using Flexible Models Incorporating Grey-Level Information. Proc 4th ICCV, 242-246 (1993)
7. R. Curwen, A. Blake: Dynamic Contours: Real-time active splines. In: A. Blake, Yuille (eds.): Active Vision. MIT Press 1932 pp. 39-57
8. C. Harris: Tracking with Rigid Models. In: A. Blake, A. Yuille (eds.): Active Vision. MIT Press 1992 pp. 59-73
9. C. Harris, C. Stennett: RAPID - A Video rate Object Tracker. BMVC90, 73-77 (1990).
10. A. Hill, T.F. Cootes , C.J. Taylor: A Generic system for Image Interpretation using Flexible Templates. Imag & Vis Comp J, 295-300 (1992).
11. M. Kass, A. Witkin, D. Terzopoulos: Snakes: Active Contours Models. ICCV-1 (IEEE Press) 259-68 (1987).
12. Kollers, Daniilidis, H.H. Nagel: Model-Based Object Tracking in Monocular Image Sequences of Road Traffic Scenes. Int J of Comp Vision, 257-821 (1993)
13. D.G. Lowe: Perceptual organisation and visual recognition. Boston, Kluwer Academic Publishers (1985).
14. D.G. Lowe: Fitting Parameterized 3-D Models to Images. IEEE Trans. on Pattern Analysis and Machine Intelligence. Vol 13, No. 5, 441-450 (1991).
15. W.H. Press, et al.: Numerical Recipes. Cambridge University Press 1986
16. R.S. Stevens: Real-time 3D Object Tracking. Proc Alvey Vision Conf, 85-90 (1989).
17. G.D. Sullivan: Visual interpretation of known objects in constrained scenes. Phil. Trans. Roy Soc (B) 337, 361-370 (1992).
18. G.D. Sullivan: Visual Traffic Understanding using the Ground-plane Constraint. Proc 2nd Intnl Conf on Comp Applic to Eng Systems (Cyprus), 473-478 (1993)
19. T.N. Tan, G.D. Sullivan, K.D. Baker: Recognising Objects on the Ground Plane. BMVC93 BMVA Press, 85-94 (1993).
20. A.D. Worrall, K.D. Baker, G.D. Sullivan: Model-based Perspective Inversion. Image and Vision Computing Journal, 17-23 (1989)