3D Data Driven Prediction for Active Contour Models based on Geometric Bounding Volumes

Joachim Denzler, Heinrich Niemann

Universität Erlangen–Nürnberg

Lehrstuhl für Mustererkennung (Informatik 5)

Martensstr. 3, D–91058 Erlangen

Tel.: +49–9131–85-7874, FAX: +49–9131–303811

email: {denzler,niemann}@informatik.uni-erlangen.de

Abstract

Active contour models have proven to be a promising approach for data driven object tracking without knowledge about the problem domain and the object. Problems arise in case of fast moving objects and in natural scenes with heterogeneous background. In these cases, a prediction step is an essential part of the tracking mechanism.

In this paper we describe a new approach for modelling the contour of moving objects in the 3D world. The key point is the description of moving objects by a simplified geometric model, the so–called bounding volume. The contour of moving objects is predicted by estimating the movement and the shape of the bounding volume in the 3D world and by projecting its contour to the image plane. Stochastic optimization algorithms are used to estimate shape parameters of the bounding volume. The 2D contour of the bounding volume is used to initialize the active contour, which then extracts the contour of the moving object. Thus, an update of the motion and model parameters of the bounding volume is possible. No task specific knowledge and no a priori knowledge about the object is necessary. We will show that in the case of convex polyhedral bounding volumes, this method can be applied to real–time closed–loop object tracking on standard Unix workstations. Furthermore, we present experiments which prove that the robustness for tracking moving objects in front of a heterogeneous background can be improved.

1 Introduction

In the past five years many different techniques in the context of active contour models (snakes) have been introduced which are mainly based on the work of (Kass et al., 1988). Besides applications in the area of contour segmentation (Ronfard, 1994, Xu et al., 1994), snakes have been successfully applied to contour tracking, also in real–time closed loop object tracking. In (Leymarie and Levine, 1993) snakes have been applied to tracking deformable objects in the 2D image plane. (Vieren et al., 1995) has proposed an approach to initialize snakes with application to tracking cars on highways. Learning the snake deformations from examples has been mentioned by (Blake et al., 1995) and (Lai and Chin, 1996). (Curwen and Blake, 1992) have presented an approach for real–time tracking with snakes.

For homogeneous background and without partial occlusions of the object, active contour models are a promising approach for real–time object tracking due to the inherent local processing of an image around the contour elements of the active contour. In natural scenes with a lot of structure and therefore a lot of strong edges in the image, strong background edges may be a better minimum than the object's contour itself. Thus, during the energy minimization, parts of or the whole active contour can be caught by the background, which results in loosing the moving object. Some work has been done which introduces special task specific knowledge to reduce the influence of background edges directly (Delagnes et al., 1995). Other researchers limit the possible deformations during the energy minimization by learning the possible shapes of the object and its deformations (Cootes et al., 1995, Ngo et al., 1995). This is a successful approach if the changes of the shape of the contour can be described by scaling, rotation, and some small amount of skewing, which normally limits the modelling of the motion to movements on a plane. If the contour of the object deforms due to changing views, this approach fails, except if 3D knowledge about the object is available.

For tracking moving objects it is a common technique to introduce a prediction step which models the coherence in time during tracking. A *prediction step* (or *prediction*) for an active contour is a mechanism, which predicts the shape of the active contour at time t_0 by using information about the shape of the active contour at time t, where $t \leq t_0$. This information might result from a 2D or 3D motion estimation for contour elements or for the global contour, or simply by taking the active contour at time $t_0 - \triangle t$ as an initialization for the active contour at time t_0.

For active contour models in principle three mechanisms for prediction are possible. First, the prediction can be done by modelling the coherence in time directly in the energy function of the snake (*implicit prediction*). Secondly, one can apply a separate prediction step to get a good starting position for the following energy minimization (*explicit prediction*). Finally, one can combine the first two prediction methods. That means, one predicts a good starting position for the energy minimization and additionally includes this starting position in the energy term of the snake, to make solutions of the minimization near this starting position more likely. One example of the explicit prediction is (Terzopoulos and Szeliski, 1992), where the motion of the snake is modelled in 2D and the motion parameters are estimated by a Kalman–Filter. A disadvantage of 2D motion description in the image plane is that by correctly estimating the motion, appearing edges of the object due to rotation cannot be predicted. Thus, 3D knowledge about the object and its motion should be available but for generality reasons, no explicit problem specific 3D models (like generic car models, (Koller, 1992)) should be taken into consideration.

In our work we propose a new approach for an explicit prediction step, which is based on a geometric modelling of the moving object's coarse 3D shape. This is called a *bounding volume*. Neither task specific knowledge nor a priori object models are necessary. The shape of the bounding volume and its position in the 3D world is estimated by comparing the 2D contour of the 3D bounding volume against the 2D active contour, which is extracted by a standard active contour approach. This information can then be the input into a Kalman–Filter, which estimates the 3D motion from 3D positions.

In Sect. 2 the motivation and a formal definition of the concept of 3D bounding volumes are given. In Sect. 2.3 we will first show how 3D bounding volumes can be applied to the problem of predicting active contours. Then, by restricting the possible geometric bounding volumes to convex polyhedra we show that the computational complexity is reduced and thus a prediction is possible in real–time using off–the–shelf hardware. Experiments and promising results are presented in Sect. 4. The paper finishes with a conclusion and an outlook on future work in Sect. 5.

2 3D Bounding Volumes

2.1 General Idea

In the introduction the need for a prediction step for active contour models has been motivated. Also the problems of a prediction in the 2D image plane have been discussed. Thus, for data driven tracking where no a priori models of the object are available one has to look for a description of the object which enables to predict the 2D contour of the object by estimating the 3D position and the coarse shape of the object itself. In computer graphics the term "bounding volume" is well known. The bounding volume of an object is the smallest volume, which completely contains the object. These bounding volumes can be applied to 2D contour prediction. The idea is the following:

1. initially extract the 2D contour of the moving object by the snake's energy minimization (for example (Kass et al., 1988)),
2. estimate the parameters of the 3D bounding volume (i.e. the location in 3D and its shape), such that the projected contour $\langle \boldsymbol{v}_i \rangle_{1 \leq i \leq m}$ of the bounding volume best matches against the extracted active contour $\langle \boldsymbol{v}'_j \rangle_{1 \leq j \leq n}$, see equation (7) and (8),
3. take the parameters calculated by step 2 to update 3D knowledge about the motion and the shape of the object, see Sect. 4,
4. calculate the object's position in 3D space and project the contour into the 2D image plane, see Sect. 2.3,
5. take this 2D contour to initialize the snake,

6. do the normal snake energy minimization, i.e. the active contour will be attracted by edges in the image under the influence of its internal energy (details of this step can be found in (Denzler and Niemann, 1995a, Kass et al., 1988)),

7. go to step 2

Three questions arise from this procedure: 1) What kind of bounding volume do we need? 2) Can we apply this procedure in real–time tracking? And 3) how accurate must the approximation of the object's contour by the contour of the bounding volume be to improve the object tracking? The first and second question will be answered in the next section. For an answer to the third question one fact must be taken into consideration. An active contour model tends to shrink. Thus, a coarse initialization around the true object contour is sufficient in most cases, as has been demonstrated by the automatic initialization of the snake described in (Denzler and Niemann, 1995a, Vieren et al., 1995). The experiments in Sect. 4 will give an answer to the third question.

2.2 Formal Description

Let $M(\boldsymbol{a})$ be the set of 3D points of a bounding volume, parameterized by a vector $\boldsymbol{a}$:

$$M(\boldsymbol{a}) = \left\{ \left({}^{w}x_i(\boldsymbol{a}), {}^{w}y_i(\boldsymbol{a}), {}^{w}z_i(\boldsymbol{a})\right)^T \mid i = 1, \ldots, n \right\} \tag{1}$$

The upper left w denotes that coordinates ${}^{w}x_i$, ${}^{w}y_i$ and ${}^{w}z_i$ of the point i refer to the 3D world. These points may be corners, edge points or generally surface points of the bounding volume. For example, for a rectangular solid, shown in Figure 1, a parameter vector $\boldsymbol{a}$ might be $\boldsymbol{a} = (l, w, h)^T$, with l, w and h being the length of the edges of the rectangular solid. In general no restrictions for the object's shape are made. The set $M(\boldsymbol{a})$ then contains the following points:

$$M((l,b,h)^T) = \left\{ \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} l \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ b \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ h \end{pmatrix}, \begin{pmatrix} 0 \\ b \\ h \end{pmatrix}, \begin{pmatrix} l \\ 0 \\ h \end{pmatrix}, \begin{pmatrix} l \\ b \\ 0 \end{pmatrix}, \begin{pmatrix} l \\ b \\ h \end{pmatrix} \right\} \tag{2}$$

The rotation $\boldsymbol{R}$ and translation $\boldsymbol{t}$ of the bounding volume maps the points of $M(\boldsymbol{a})$ to the set ${}^{\boldsymbol{R},\boldsymbol{t}}M(\boldsymbol{a})$, which contains the rotated and translated 3D points of the bounding volume. Formally, we write

$$M(\boldsymbol{a}) \stackrel{\boldsymbol{R},\boldsymbol{t}}{\longrightarrow} {}^{\boldsymbol{R},\boldsymbol{t}}M(\boldsymbol{a}) \tag{3}$$

Now, a visibility test must be done. In the literature of computer graphics several algorithms can be found (z–buffer, scan–line, raytracing (Foley et al., 1994)). We define a hiding operator $\mathcal{H}$, i.e.

$${}^{\boldsymbol{R},\boldsymbol{t}}M(\boldsymbol{a}) \stackrel{\mathcal{H}}{\longrightarrow} {}^{\boldsymbol{R},\boldsymbol{t}}M'(\boldsymbol{a}) \subseteq {}^{\boldsymbol{R},\boldsymbol{t}}M(\boldsymbol{a}). \tag{4}$$

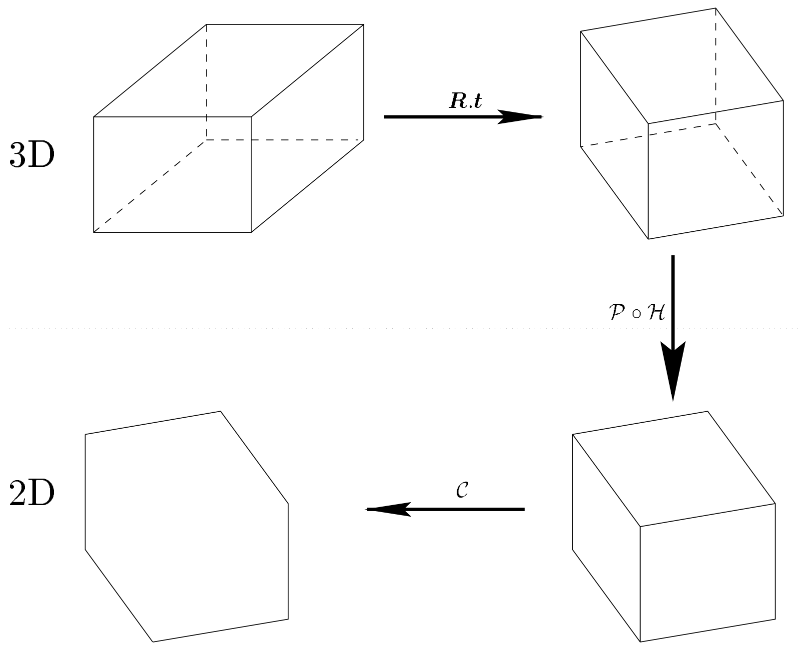

Figure 1: 3D bounding volumes for 2D contour prediction: The mappings $\mathcal{P}$, $\mathcal{H}$, and $\mathcal{C}$ are formally described in the text. By estimating the position of the bounding volume we can calculate the 2D contour in the image plane which is needed to initialize the active contour.

which maps the set ${}^{\boldsymbol{R},\boldsymbol{t}}M(\boldsymbol{a})$ of 3D points into the set ${}^{\boldsymbol{R},\boldsymbol{t}}M'(\boldsymbol{a})$ of visible 3D points. Now, the set ${}^{\boldsymbol{R},\boldsymbol{t}}M'(\boldsymbol{a}) \subset \mathbb{R}^3$ will be projected onto the image plane by perspective projection $\mathcal{P}$

$$ {}^{\boldsymbol{R},\boldsymbol{t}}M'(\boldsymbol{a}) \xrightarrow{\mathcal{P}} {}^{\boldsymbol{R},\boldsymbol{t}}M'_{\mathcal{P}}(\boldsymbol{a}) \subset \mathbb{R}^2 \tag{5} $$

The result is the set ${}^{\boldsymbol{R},\boldsymbol{t}}M'_{\mathcal{P}}(\boldsymbol{a})$ which is equal to the 2D image of the point of the bounding volume. Finally an operator $\mathcal{C}$ will compute the visible 2D contour of the bounding volume, which leads to a set of points in $\mathbb{R}^2$:

$$ {}^{\boldsymbol{R},\boldsymbol{t}}M'_{\mathcal{P}}(\boldsymbol{a}) \xrightarrow{\mathcal{C}} {}^{\boldsymbol{R},\boldsymbol{t}}C_{\mathcal{P}}(\boldsymbol{a}) \subseteq {}^{\boldsymbol{R},\boldsymbol{t}}M'_{\mathcal{P}}(\boldsymbol{a}) \tag{6} $$

The points of the set ${}^{\boldsymbol{R},\boldsymbol{t}}C_{\mathcal{P}}(\boldsymbol{a})$ need to be transformed to a sequence of points $\langle \boldsymbol{v}_i \rangle_{1 \le i \le m}$, with $\boldsymbol{v}_i \in {}^{\boldsymbol{R},\boldsymbol{t}}C_{\mathcal{P}}(\boldsymbol{a})$, ordered counterclockwise to form a representation of this contour.

In Figure 1 all steps of this approach are summarized. The mappings $\mathcal{H}$ and $\mathcal{C}$ of equations (4) and (6) are time critical for real–time experiments. In the following section we will show how this problem can be solved.

For two contours $\langle \boldsymbol{v}_i \rangle_{1 \le i \le m}$ and $\langle \boldsymbol{v}'_j \rangle_{1 \le j \le n}$ (compare step 2 of the algorithm in Sect. 2.1) a distance function $\mathrm{dist}(\langle \boldsymbol{v}_i \rangle_{1 \le i \le m}, \langle \boldsymbol{v}'_j \rangle_{1 \le j \le n})$, for example

$$ \mathrm{dist}(\langle \boldsymbol{v}_i \rangle_{1 \le i \le m}, \langle \boldsymbol{v}'_j \rangle_{1 \le j \le n}) = \sum_{i=1}^{m} \min_{\boldsymbol{v}'_j} \left\{ \left| \boldsymbol{v}_i - \boldsymbol{v}'_j \right| \right\} + \sum_{j=1}^{n} \min_{\boldsymbol{v}_i} \left\{ \left| \boldsymbol{v}_i - \boldsymbol{v}'_j \right| \right\} \tag{7} $$

is defined. This function measures the correspondence of two 2D contours. Now, for a given active contour $\boldsymbol{v}'(j)$ and a parameter description of a bounding volume, the parameters $\boldsymbol{R}, \boldsymbol{t}$ and $\boldsymbol{a}$ can be computed by

$$(\boldsymbol{R}, \boldsymbol{t}, \boldsymbol{a})^T = \operatorname*{argmin}_{\boldsymbol{R}, \boldsymbol{t}, \boldsymbol{a}} \operatorname{dist}(\langle \boldsymbol{v}_i \rangle_{1 \le i \le m}, \langle \boldsymbol{v}'_j \rangle_{1 \le j \le n}), \text{ where} \quad (8)$$

$$\boldsymbol{v}_i \in {}^{\boldsymbol{R},\boldsymbol{t}}C_{\mathcal{P}}(\boldsymbol{a}) \quad (9)$$

The minimization in (8) results in the parameters $\boldsymbol{R}, \boldsymbol{t}$ and $\boldsymbol{a}$ of that 3D bounding volume, the contour of which best matches — in the sense of equation (7) — the active contour. (compare item 2 at the beginning of Sect. 2.1). To calculate the parameters $\boldsymbol{R}, \boldsymbol{t}$ and $\boldsymbol{a}$ we use the stochastic optimization techniques described in (Ermakov and Zhiglyavskij, 1983).

After this step we have based on the 2D active contour, which has been extracted out of the image, a 3D estimate of the moving object's 3D bounding volume. The only knowledge which is needed for this step is a parametric representation of the 3D bounding volume, which has to be chosen in advance. In our experiments (see Sect. 4) we have taken a block.

In the following section we will show how the parameter estimation problem in equation (8) can be solved in real–time by restricting the description of the bounding volumes to the class of convex polyhedra.

2.3 2D Contour Prediction by 3D Bounding Volumes

In the previous section we have motivated the idea of 2D contour prediction by 3D bounding volumes and we have given a formal description of the various problems which have to be solved. Now, we restrict the possible geometric description to convex polyhedra. Then, no visibility test $\mathcal{H}$ is necessary. Instead, the contour operator $\mathcal{C}$ can be implemented by projecting the corners of the bounding volume to the 2D image plane and computing the convex hull of these 2D corners. A proof, which is rather intuitive and well known, can be found in (Haralick and Shapiro, 1993). As a result, the 2D contour of a convex polyhedron can be calculated by projecting the 3D corners onto the image plane and calculating the 2D convex hull of this set of 2D image points. An advantage of this approach is that first the bounding volume can be represented by a small set of 3D point (the corners). And second, no visibility test (operator $\mathcal{H}$) is necessary. We only need to compute the convex hull of a small set of points (for a block we have 8 points), which can be done in real–time even without specialized hardware on standard Unix workstations. One efficient algorithm for the computation of the convex hull of a set of points can be found in (Ye, 1995).

3 Object Tracking and Prediction

For applying the prediction method proposed in Sect. 2 to active contour models, first, one has to choose a parametric representation of the bounding volume of the object. In our experiments we have chosen a block. Of course, this can also be done automatically, if the moving object has been identified in a recognition step. This, however, has not been the scope of our investigations yet.

The next step is the motion detection and automatic initialization of the active contour. Assuming a static camera in this initialization step, we compute the difference image between consecutive frames and after a threshold operation a binary image, which contains a region, in which motion has occurred. The border of this region is taken as an initialization of the active contour, which then extracts the contour of the moving object by the normal energy minimization step. A more detailed description of the initialization step including error recognition can be found in (Denzler and Niemann, 1995a).

As soon as the active contour algorithm has extracted the contour of the moving object, the shape and position of the 3D bounding volume is estimated by minimizing (8). For the minimization we use stochastic optimization techniques as already mentioned in Sect. 2. As a result one gets $\boldsymbol{R}, \boldsymbol{t}$ and $\boldsymbol{a}$ of the 3D bounding volume. The parameter $\boldsymbol{R}$ and $\boldsymbol{t}$ can be taken to estimate the 3D motion of the object, for example, by using a Kalman Filter. Thus, for the next image a prediction of the 3D bounding volume's position and its 2D contour is possible. The 2D contour is then taken to initialize the active contour. The active contour again extracts the contour of the moving object and an update of the shape and position of the 3D bounding volume is possible.

A problem, which has already been mentioned in Sect. 2.1, is the correspondence of the parametric representation of the bounding volume and the real shape of the moving object. Can our proposed method improve tracking with active contours, even if the real shape of the moving object differs from the estimated bounding volume? And why should the representation of the bounding volume be more useful than just the contour ? The answer to the two questions will be given by the experimental results in Sect. 4. Regarding an answer to the second question one has to take into consideration that once we have 3D knowledge about the object we can compute information about the 3D motion (e.g. "the object is moving toward the observer") and we can handle partial occlusion (see Figure 2).

4 Experiments

In this section we present results for tracking a moving object by using the proposed method (Sect. 2 and Sect. 2.3) for contour modelling and projection. The 3D position and shape parameters of the bounding volume, calculated at time t by equation (8), are taken directly as the initialization of the contour at time $t+1$. This means, that at present no motion estimation (see step 3 in Sect. 2.1) is done. This is subject of our future work, for example estimating the motion parameters by a Kalman–Filter.

After initialization of the active contour a standard active contour model is a applied, based on the variational calculus (Kass et al., 1988), which has been modified to satisfy real–time constraints: the iterative energy minimization has been simplified, a new energy term and blowing forces known from the balloon model have been introduced. More details can be found in (Denzler and Niemann, 1995b). A moving toy train is tracked in real–time by a low–cost multimedia pan/tilt camera device. The speed of the toy train is about 1.4

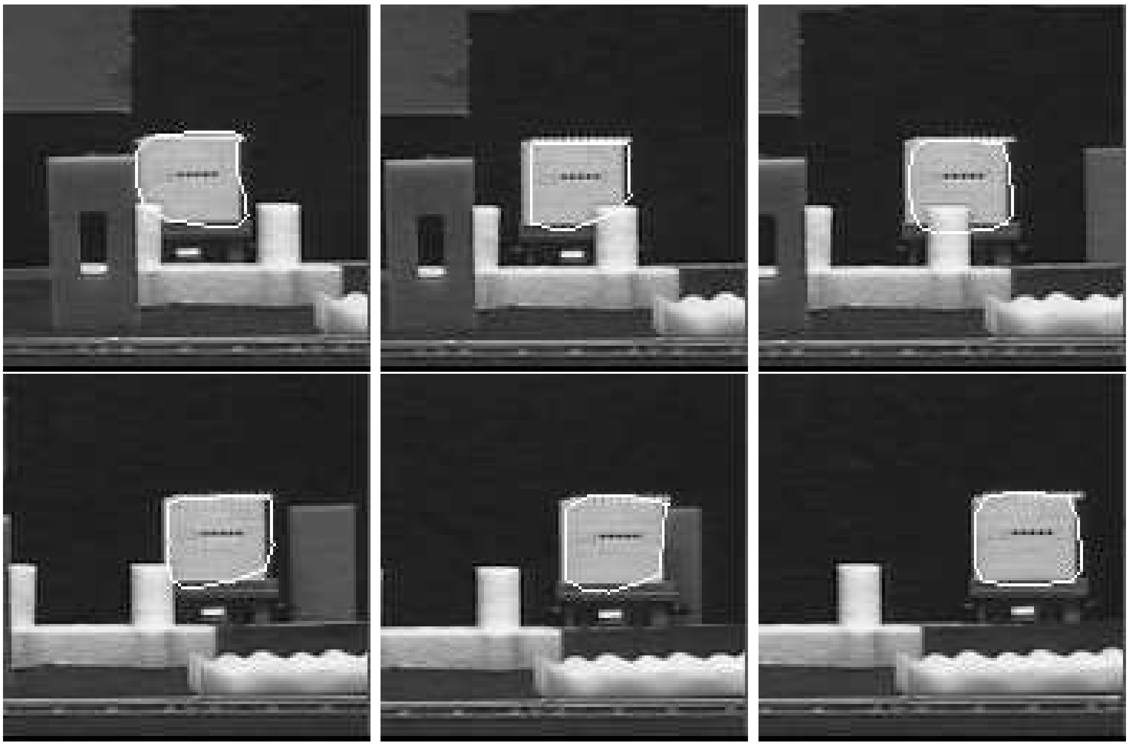

Figure 2: Results for tracking a moving object with the proposed 3D prediction method based on bounding volumes. The resulting active contour extracts the object's boundary.

cm/sec, which is equal to a displacement in the image plane of about 2–7 pixels. All algorithms are implemented in C++ and run on an SGI Onyx (two R4400 processors) with a Sirius frame grabbing board. The board does no preprocessing operations. No specialized hardware is used.

It is worth noting that at present the estimated 3D bounding volume at time t is directly used to initialize the active contour at time $t+1$. This means, that for the results presented in this paper no parameter estimation of the 3D motion, for example, using a Kalman–Filter, is done. This will be integrated in our future work. Instead, the search space for the calculation of the parameters $\boldsymbol{R}, \boldsymbol{t}$ and $\boldsymbol{a}$ of the 3D bounding volume (compare (8)) is limited to a small interval around the estimated parameter vector of the previous image.

In Figure 2 results for tracking the toy train in front of heterogeneous background are shown. Without a prediction step, the active contour is caught by one of the strong background objects. Also the partial occlusion would cause an error in the tracking. As one can see, both problems can be handled by the proposed 3D bounding volume prediction method.

In Figure 3 another sequence is shown in which the object's contour changes its shape due to a changing view. In this case, as already mentioned in the introduction, 3D knowledge is necessary to correctly predict the shape of the active contour.

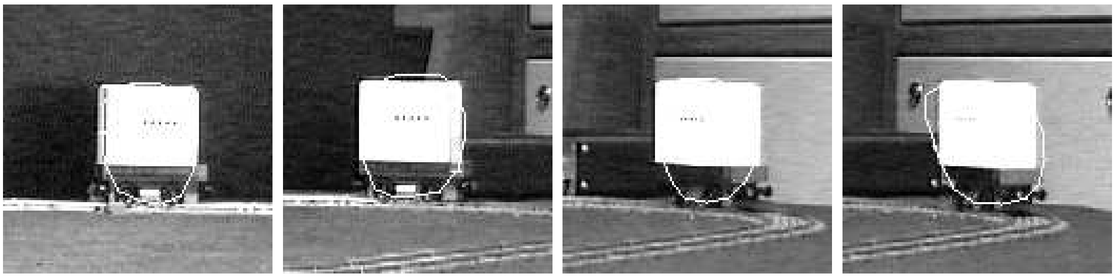

Figure 3: Results for tracking a moving toy train. The contour of the toy train is changing due to a changing view. The 3D bounding volume correctly models the shape of the toy train's contour. The calculated rotation angle around y–axis for the 3D bounding volume during the tracking can be seen in Figure 4.

Figure 4: Calculated angle around the y–axis of the 3D bounding volume (by equation (8)) for the image sequence shown in Figure 3.

Finally, in Figure 5 the results for the estimation of the 3D bounding volume (see equation (8)) are shown: the 2D contour of the bounding volume, which is taken as an initialization of the active contour (see item 5 of the algorithm, described in Sect. 2.1), a wire frame model of the bounding volume itself (a cube) and the bounding volume drawn into the image.

The results answer the third question, which has been asked in Sect. 2.1: As one can see in Figure 5, the 3D bounding volume need not accurately approximate the real bounding volume of the object. It is sufficient, to get an approximation, which 2D contour best matches against the extracted active contour. Then, strong background edges near the moving object and even partial occlusions (see Figure 2) can be handled. Also Figure 4 shows that the calculated 3D position of the 3D bounding volume corresponds to a 3D motion (rotation of $\triangle\phi \approx 80$ degree between image 20 and 140). Thus, a Kalman–Filter

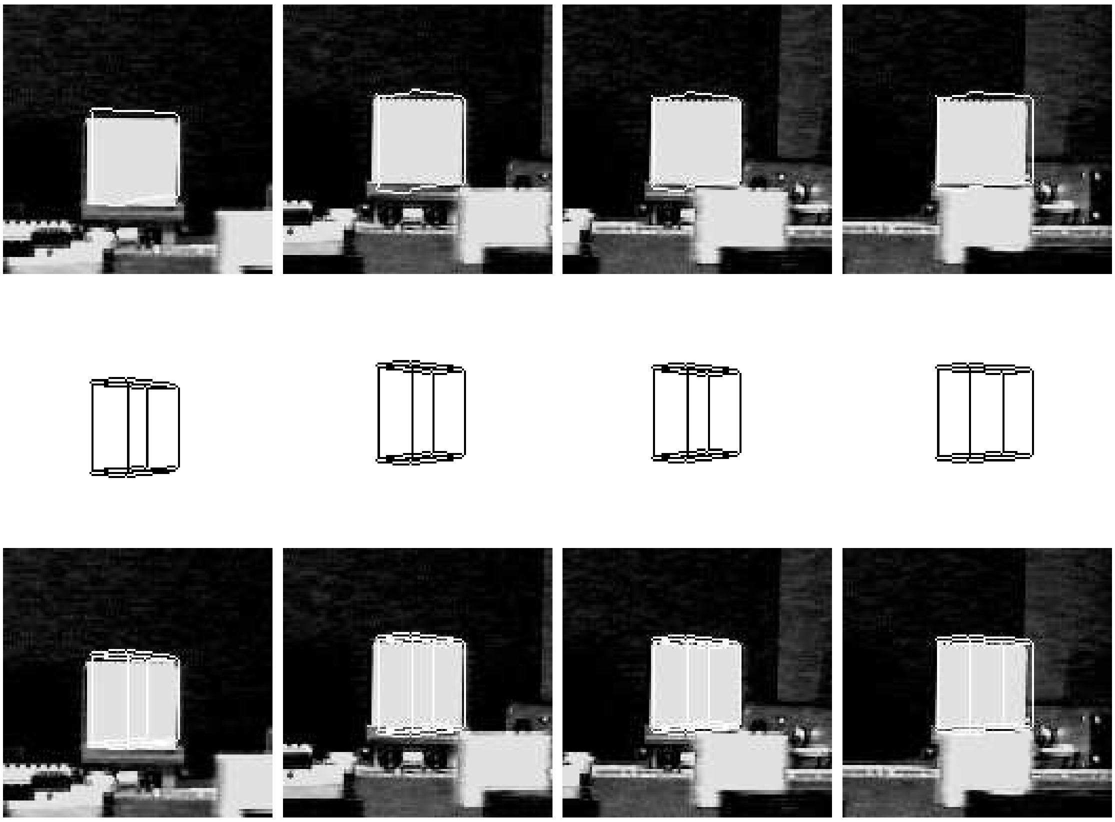

Figure 5: Results of the 3D bounding volume estimation during the tracking: the 2D contour of the bounding volume (first row), a wire frame model of the bounding volume (second row), and the bounding volume drawn into the image (third row).

can be applied to estimate the motion of the 3D bounding volume. Knowledge about the motion instead of the position will further improve the prediction. This will be investigated in our future work.

Since this approach is independent of the chosen active contour model, it can be applied as a prediction step to any tracking algorithm, based on active contour models.

5 Conclusion

In this paper we presented a framework for predicting 2D contours of objects moving in 3D without a priori knowledge of the object's shape in 3D. This data driven approach — the so called bounding volumes — can especially be applied to active contour models, where a prediction of the 2D contour of an object based on the 3D shape and motion is necessary. We have also shown that by limiting the class of bounding volumes to convex polyhedra the prediction can be done in real–time. The results have proven that this method can be

Figure 6: Results for the estimation of bounding volumes for three moving cars (right), which have been tracked by active contours (left).

integrated in a real–time object tracking system. Strong background edges near the object and even partial occlusions as well as changing views of the object can be handled. The results also have shown that the 3D bounding volume need not accurately approximate the real shape of the moving object.

There are several investigations, which will be done in our future work. At this time, no 3D motion estimation is done. The 2D contour of the 3D bounding volume is only used to initialize the active contour (i.e. the prediction of item 4 of the algorithm in Sect. 2.1 is the identity). This means that the solution for the shape parameters and the position in the 3D world may be ambiguous. Nevertheless, this approach can simply be extended by a Kalman–Filter. Then, the 3D motion parameters can be estimated based on the observations of the position in the 3D world. Also, after some initialization stage the time consuming estimation of the parameters by the stochastic optimization algorithms could be done every $\triangle t$ images (for example twice every second, i.e. $\triangle t = 12$), to verify the motion and position estimation of the Kalman–Filter. This reduces the computation time which is important for real–time applications. Finally, we have to test whether the convex polyhedra approximation is sufficient to track non polyhedral objects. First results for tracking cars on highways can be found in Figure 6, right image, which also show, that the proposed method can be applied to natural scenes. Since the quality of the results of the proposed method depends on the 2D contour extraction in the image, limitation are given in highly structured scenes due to the problems for active contours, which has been already mentioned in the introduction. Nevertheless, this prediction step allows for a data driven contour prediction for objects moving in 3D. Compared to recently published 2D prediction steps, this enables to handle partial occlusion and even the prediction of deforming contours due to changing views to the object.

References

Blake, A., Isard, M., and Reynard, D. (to appear 1995). Learning to track the visual motion of contours. *Artificial Intelligence*.

Cootes, T., Taylor, C., Cooper, D., and Graham, J. (1995). Active shape models — their training and applications. *Computer Vision, Graphics, and Image Processing*, 61(1):38–59.

Curwen, R. and Blake, A. (1992). Dynamic contours: Real–time active splines. In Blake, A. and Yuille, A., editors, *Active Vision*, pages 39–58. MIT Press, Cambridge, Massachusetts, London, England.

Delagnes, P., Benois, J., and Barba, D. (1995). Active contours approach to object tracking in image sequences with complex background. *Pattern Recognition Letters*, 15:171–178.

Denzler, J. and Niemann, H. (1995a). Combination of simple vision modules for robust real–time motion tracking. *European Transactions on Telecommunications*, 5(3):275–286.

Denzler, J. and Niemann, H. (1995b). Evaluating the performance of active contour models for real–time object tracking. In *Second Asian Conference on Computer Vision*, pages II/341–II/345, Singapore.

Ermakov, S. M. and Zhiglyavskij, A. A. (1983). On random search of global extremum. *Probability Theory and Applications*, 28(1):129–136.

Foley, J., van Dam, A., Feiner, S., Hughes, J., and Phillips, R. (1994). *Grundlagen der Computergraphik*. Addison–Wesley, Bonn, Paris, Reading, Massachusetts.

Haralick, R. and Shapiro, L. (1993). *Computer and Robot Vision*, volume 2. Addison–Wesley, Reading, Massachusetts.

Kass, M., Wittkin, A., and Terzopoulos, D. (1988). Snakes: Active contour models. *International Journal of Computer Vision*, 2(3):321–331.

Koller, D. (1992). *Detektion, Verfolgung und Klassifikation bewegter Objekte in monokularen Bildfolgen am Beispiel von Straßenverkehrsszenen*, volume 13 of *Dissertationen zur künstlichen Intelligenz*. infix, St. Augustin.

Lai, K. and Chin, R. (1996). Deformable contours: Modelling and extraction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(1):1–20.

Leymarie, F. and Levine, M. (1993). Tracking deformable objects in the plane using an active contour model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(6):617–634.

Ngo, C., Chan, S., and Lai, K. (1995). Motion tracking and analysis of deformable objects by generalized active contour models. In *Second Asian Conference on Computer Vision*, pages III/442–III/446, Singapore.

Ronfard, R. (1994). Region-based strategies for active contour models. *International Journal of Computer Vision*, 13(2):229–251.

Terzopoulos, D. and Szeliski, R. (1992). Tracking with Kalman snakes. In Blake, A. and Yuille, A., editors, *Active Vision*, pages 3–20. MIT Press, Cambridge, Massachusetts, London, England.

Vieren, C., Cabestaing, F., and Postaire, J. (1995). Catching moving objects with snakes for motion tracking. *Pattern Recognition Letters*, 16:679–685.

Xu, G., Segawa, E., and Tsuji, S. (1994). Robust active contours with insensitive parameters. *Pattern Recognition*, 27(7):879–884.

Ye, Q. (1995). A fast algorithm for convex hull extraction in 2d images. *Pattern Recognition Letters*, 16:531–537.