Active Rays: A New Approach to Contour Tracking

J. Denzler, H. Niemann
Lehrstuhl für Mustererkennung (Informatik 5)
Universität Erlangen–Nürnberg
Martensstr. 3, D–91058 Erlangen, Germany
email: {denzler,niemann}@informatik.uni-erlangen.de

denzler@informatik.uni-erlangen.de

The following paper will appear in the
Proceedings on the 3^{rd} German–Slovenian Workshop
on
Speech and Image Analysis

Ljubljana, April $24^{th} - 26^{th}$, 1996

Active Rays: A New Approach to Contour Tracking

J. Denzler, H. Niemann
Lehrstuhl für Mustererkennung (Informatik 5)
Universität Erlangen–Nürnberg
Martensstr. 3, D–91058 Erlangen, Germany
email: {denzler,niemann}@informatik.uni-erlangen.de

Abstract

In this paper we describe a new approach to contour extraction and tracking, which is based on the principles of active contour models and overcomes its shortcomings. We formally introduce *active rays*, describe the contour extraction as an energy minimization problem and discuss what active contours and active rays have in common.

The main difference is that for active rays a unique ordering of the contour elements in the 2D image plane is given, which cannot be found for active contours. This is advantageous for predicting the contour elements' position and prevents crossings in the contour. Furthermore, another advantage of this approach is that instead of an energy minimization in the 2D image plane the minimization is reduced to a 1D search problem. The approach also shows any–time behavior which is important with respect to real–time applications. Finally, the method allows for the management of multiple hypotheses of the object's boundary.

First results on real image sequences shows the suitability of this approach for real–time object tracking. The contour tracking can be done within the image frame rate (25 fps) on standard Unix workstations (HP 735).

Keywords: active contour models, tracking, real–time

1 Introduction

The field of real–time computer vision has become more and more important in the past 10 years. Due to the increasing hardware performance and to new strategies for the processing of images and of image sequences (active vision, [2]), applications, which work in a closed loop between sensing and action, have been developed recently [6, 7, 11]. Especially, for real–time object tracking many algorithms can be found in the literature [3, 5, 8, 13]. One class of algorithms is the so called active contour model (snake) [**?**], which allows for data driven contour segmentation, extraction, and tracking. They are well suited for real–time applications due to the inherent local processing of an image

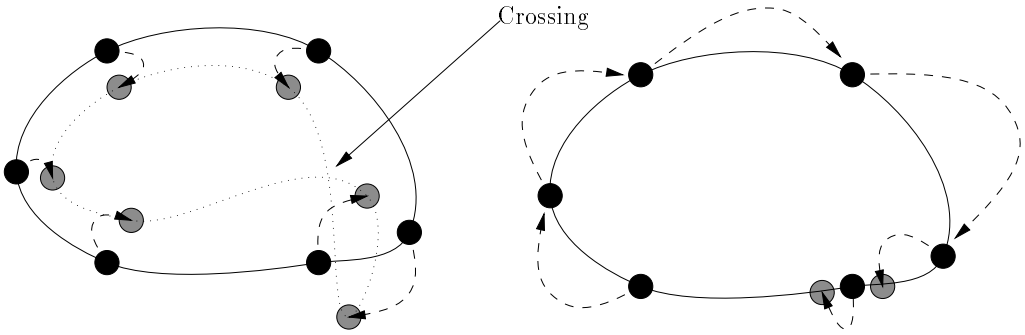

Figure 1: Two main problems of active contours during tracking resulting from the missing ordering in the 2D image plane: Crossings in the 2D contour may occur (left). The snake elements are not fixed at logical features on the object's contour, but they may move around the contour as they like (right).

nearby the snake elements. In front of a homogeneous background moving objects can be robustly tracked in real–time [4, 14]. Introducing a prediction step, also some amount of inhomogeneous background can be allowed. Finally, this approach is insensitive with respect to different camera devices and changes in the camera parameters (focus, zoom, aperture); it is also robust to changing lighting conditions, even during the tracking.

But this contour tracking method still has some limitations. Strong background edges near the object's contour are also good minima for the energy minimization of the active contour. Thus, in natural scenes special task specific constraints [12, 16] are necessary which increase the computation time and reduce the real–time performance. Without such task specific constraints tracking may fail. Another problem is the missing order of the snake elements in the 2D image plane. During the energy minimization crossings in the contour might occur [17] which result in an incorrect contour extraction (see Figure 1, left). Algorithms exist [17], which can handle this case, but also increase the computation time of the algorithm. Another problem arises from the missing order in the 2D image plane. Even if no crossing of the contour occurs, one cannot find any logical correspondence between active contour elements and points at the contour of the moving object. This means, one cannot predict in principle to which point a single snake element moves. Only the movement and — for some amount — the distortion of the whole active contour can be predicted (see Figure 1, right: the shape of the contour remains the same, although all the snake elements have moved around the contour). Finally, no work is known which adds some any–time behavior to active contours. Of course, an iterative minimization can be seen as an any–time algorithm. Reducing the iteration steps in one image results in a less accurate extraction of the object's contour. But within the next images the snake elements might remove from the sphere of influence of the object's contour and thus lose the moving object. This problem is explained in Figure 2. A 1D contour is shown at three different times (t_1, t_2, t_3), moving along the x–axis. The snake element at time t_1 (black circle) is in the sphere of influence of the 1D contour. It reaches the minimum after some iterations by moving downhill the contour. At time t_2 the contour element is

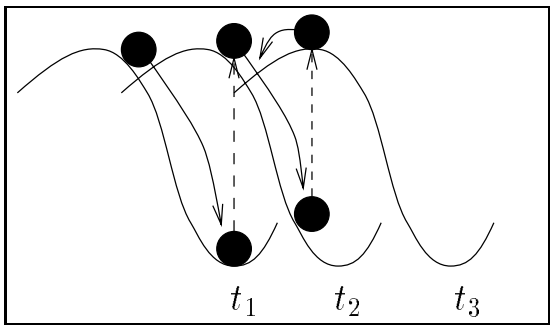

Figure 2: Problem, when reducing the number of iteration steps during the energy minimization while tracking a 1D signal (time t_1, t_2, t_3).

again in the sphere of influence. This allows for moving downhill to the minimum. Now the iteration number is reduced, which means that the energy minimization stops before the real minimum is reached. Then, at time t_3 the snake element (gray circle) is not in the sphere of influence of the 1D contour and cannot reach the true minimum. Thus, the contour is lost.

In this paper we propose a new approach to contour extraction and tracking, which we call *active rays*. These active rays contain principles of active contour models (i.e., energy minimization, local processing of the image, contour representation of the moving object) and overcome the mentioned shortcomings. To be more precise, active rays

- show any–time behavior,
- reduce the 2D energy minimization of snakes to a 1D search problem,
- allow for using multiple hypotheses for the object's boundary, and
- have a fixed 2D order of the contour elements.

In Sect. 2 we formally introduce active rays and we present an energy description which shows the common parts of active contour models and active rays. In Sect. 3 we apply active rays to contour tracking and show the possible any–time behavior. First experiments and results for real–time object tracking can be found in Sect. 4. The paper ends with a summary and discussion (Sect. 5) and an outlook to future work (Sect. 6).

2 Active Rays

This section is organized into three parts. First, a formal description for active rays is given. In the second part we formulate the contour extraction as an energy minimization problem and we compare active contour models and active rays. Finally, we motivate the use of multiple hypotheses and the any–time behavior of our proposed method.

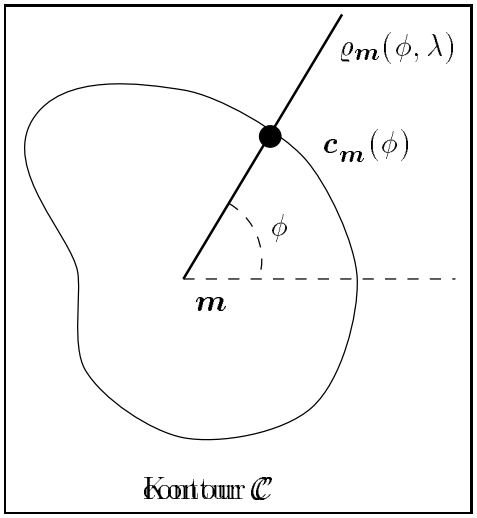

Figure 3: Principle of one active ray.

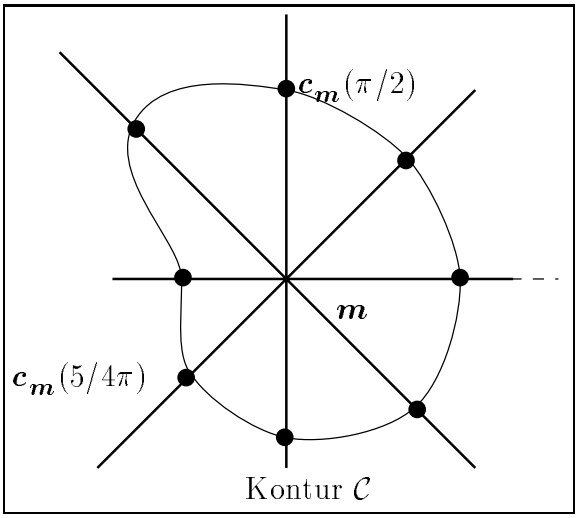

Figure 4: Representation of a contour by active rays.

2.1 Formal Description

An active ray $\varrho_{\boldsymbol{m}}(\phi,\lambda)$ is defined on the image plane (x,y) as a 1D function depending on those gray values $f(x,y)$ of the image, which are on a straight line from the image point $\boldsymbol{m} = (x_m, y_m)^T$ in direction ϕ

$$\varrho_{\boldsymbol{m}}(\phi,\lambda) = f(x_m + \lambda\cos(\phi), y_m + \lambda\sin(\phi)), \quad 0 \le \lambda \le n_\phi, \tag{1}$$

where n_ϕ is given by the image size. The principle is clarified in Figure 3. The angle ϕ is measured counter clockwise.

Now, a contour point in direction ϕ regarding a given reference point $\boldsymbol{m}$ can be described by the parameter $\lambda(\phi) \ge 0$

$$\begin{aligned}\lambda(\phi) &= \operatorname*{argmin}_{\lambda} \left(-\left|\nabla f(x_m + \lambda\cos(\phi), y_m + \lambda\sin(\phi))\right|^2\right) \\ &= \operatorname*{argmin}_{\lambda} \left(-\left|\frac{\partial}{\partial\lambda}\varrho_{\boldsymbol{m}}(\phi,\lambda)\right|^2\right), \quad 0 \le \phi < 2\pi,\end{aligned} \tag{2}$$

i.e., we are looking for points on the active ray with a maximum edge strength. The contour point $\boldsymbol{v}_{\boldsymbol{m}}(\phi)$ (see Figure 3) is then

$$\boldsymbol{v}_{\boldsymbol{m}}(\phi) = (x_m + \lambda(\phi)\cos(\phi), y_m + \lambda(\phi)\sin(\phi)), \quad 0 \le \phi < 2\pi \tag{3}$$

A similar representation is used by the generalized Hough transform. In the discrete case the whole contour can be computed by defining a sampling step size $\triangle\phi$ for ϕ. This allows for different accuracy of the contour representation. An example for a representation of a contour is shown in Figure 4. The sampling step size $\triangle\phi$ is $\pi/4$.

Now, we have to discuss the choice of the reference point $\boldsymbol{m}$. In principle, every point within the object's contour is possible. But to have a unique point, which can be

precalculated by a prediction step, the center of gravity of the contour extracted by the active ray is used in the following, i.e., the equation

$$\boldsymbol{m} = 1/2\pi \int_0^{2\pi} \boldsymbol{v}_{\boldsymbol{m}}(\phi)\, d\phi \tag{4}$$

should hold for the reference point $\boldsymbol{m}$. For convex contours $\boldsymbol{m}$ will also be the center of gravity of the object's contour. What happens, if the chosen reference point is not the center of gravity? Then, we can calculate a new reference point using the formula (4). After that, the new contour representation has to be calculated.

2.2 Definition of an Energy Term

Equation (2) leads to single contour points without taking into account the global shape of the underlying contour. Thus, errors might occur for real images due to noise in the image or background edges near the object. Without coupling neighboring contour points the function λ normally will not correctly represent the contour. This can be seen in Figure 5, where the function λ for the contour in Figure 6 (middle) is shown. For the angles $\phi \in [4/3\pi, 3/2\pi\,[$ a strong edge is extracted, which does not belong to the contour which corresponds to $\lambda(\phi)$, $\phi \notin [4/3\pi, 3/2\pi\,[$. In Figure 5 this results in four values of λ (x–axis: $\frac{4}{3}\pi - \frac{3}{2}\pi$), which are outliers in this function plot.

A common technique to solve this problem is defining an internal energy to connect the contour elements of the active ray (see equation 3). For snakes a common definition of the internal energy $E_i(\boldsymbol{v}(s))$ of an active contour element $\boldsymbol{v}(s)$ is (cf. [**?**])

$$E_i(\boldsymbol{v}(s)) = \frac{\alpha(s)|\boldsymbol{v}_s(s)|^2 + \beta(s)|\boldsymbol{v}_{ss}(s)|^2}{2}, \tag{5}$$

with $\boldsymbol{v}_s(s)$ and $\boldsymbol{v}_{ss}(s)$ being the first and second derivatives of $\boldsymbol{v}(s)$. This energy definition weighted by $\alpha(s)$ and $\beta(s)$ describes the membrane and thin plate behavior of a snake [**?**]. For an active ray the same behavior can be produced by defining

$$E_i(\boldsymbol{v}(s)) = E_i(\boldsymbol{v}_{\boldsymbol{m}}(\phi)) = \frac{\alpha(\phi)\left|\frac{d}{d\phi}\boldsymbol{v}_{\boldsymbol{m}}(\phi)\right|^2 + \beta(\phi)\left|\frac{d^2}{d\phi^2}\boldsymbol{v}_{\boldsymbol{m}}(\phi)\right|^2}{2}. \tag{6}$$

This is clarified in Figure 7 which shows the common ground of active contours and active rays. In the case of an active ray, a reference point is given. Thus, a better definition of the internal energy $E_i(\boldsymbol{v}_{\boldsymbol{m}}(\phi))$ is

$$E_i(\boldsymbol{v}_{\boldsymbol{m}}(\phi)) := E_i(\phi) = \frac{\alpha(\phi)\left|\frac{d}{d\phi}\lambda(\phi)\right|^2 + \beta(\phi)\left|\frac{d^2}{d\phi^2}\lambda(\phi)\right|^2}{2}. \tag{7}$$

The function $\lambda(\phi)$ is given by equation (2). Now we need an external energy. Let us take the usual image gradient, i.e.,

$$E_e(\boldsymbol{v}_{\boldsymbol{m}}(\phi)) = -|\nabla f(\boldsymbol{v}_{\boldsymbol{m}}(\phi))|^2 = -\left|\frac{d}{d\lambda}\varrho_{\boldsymbol{m}}(\phi,\lambda)\right|^2 \tag{8}$$

Equations (7) and (8) both need computations only for a 1D signal, compared to the energy definitions of an active contour, which needs a 2D minimization. This is advantageous for real–time applications.

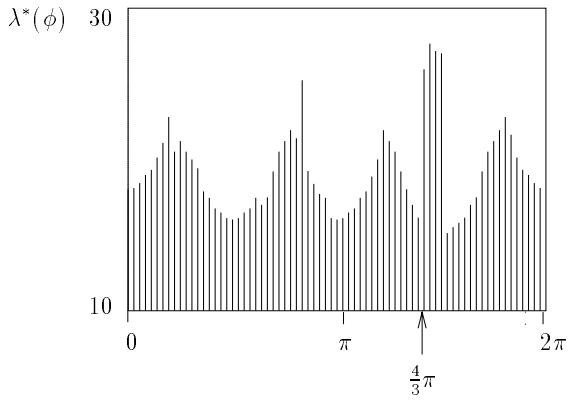

Figure 5: The function λ for the contour shown in Figure 6: one point of the x–axis is equal to five degree.

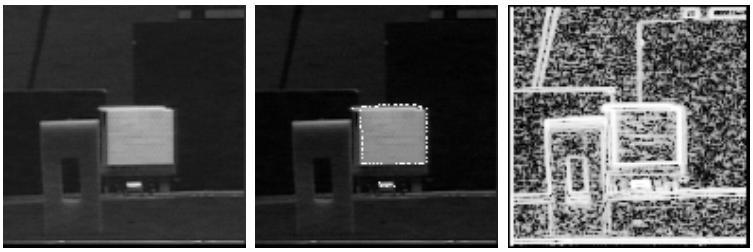

Figure 6: Example of a contour representation by active rays: The image (left), the active ray representation (middle) and the underlying external energy (right), computed by a Sobel operator.

2.3 Energy Minimization

After a formal description of an active ray and its internal and external energy, we can go on formulating the contour extraction as an energy minimization problem. For this, the same formalisms as for active contour models can be applied (for example, the Greedy algorithm [15], dynamic programming [1], etc.).

The total energy of an active ray, which is uniquely given by the function $\lambda(\phi)$ and a reference point $\boldsymbol{m}$, can be defined as

$$E = \int_0^{2\pi} \left[E_i(\lambda(\phi)) + E_e(\lambda(\phi))\right] d\phi \tag{9}$$

$$= \int_0^{2\pi} \left[\frac{\alpha(\phi)\left|\frac{d}{d\phi}\lambda(\phi)\right|^2 + \beta(\phi)\left|\frac{d^2}{d\phi^2}\lambda(\phi)\right|^2}{2} - \left|\frac{d}{d\lambda}\varrho_{\boldsymbol{m}}(\phi,\lambda)\right|^2 \right] d\phi \tag{10}$$

Now, we are looking for a function $\lambda(\phi)$, which minimizes this energy E, i.e., we have to

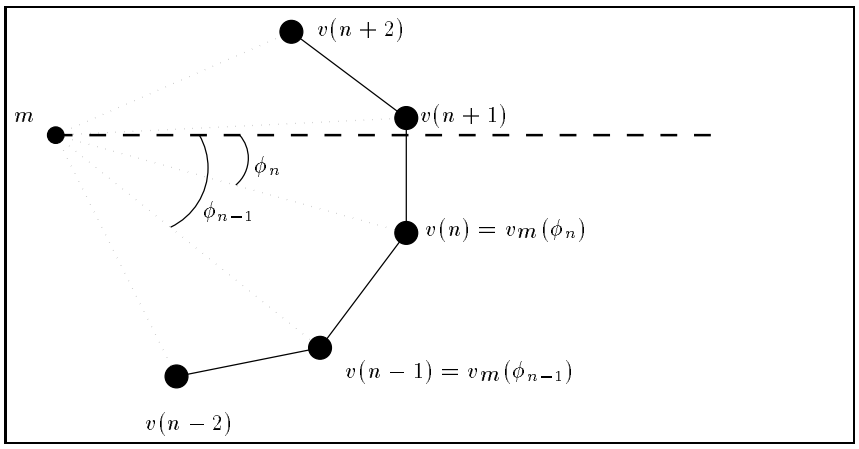

Figure 7: Calculation of the internal energy of a snake and an active ray. The same energy definition can be used for both models.

solve the Euler–Lagrange differential equation

$$\alpha(\phi)\frac{d^2}{d\phi^2}\lambda(\phi) - \beta(\phi)\frac{d^4}{d\phi^4}\lambda(\phi) + \frac{d}{d\lambda}\left|\frac{d}{d\lambda}\varrho_{\boldsymbol{m}}(\phi,\lambda)\right|^2 = 0. \tag{11}$$

This can be done in the discrete case by using an iterative algorithm [**?**].

2.4 Multiple Hypotheses and Any–Time Behavior

In the introduction we have noted that the approach of active rays allows for multiple hypotheses. For this, we have to look for the i best solutions of equation (2), which means, that for each ray in direction ϕ we get a set $\Lambda(\phi)$

$$\Lambda(\phi) = \left\{ \lambda_k(\phi) | \lambda_k(\phi) = \underset{\lambda,\lambda\neq\lambda_l,l<k}{\operatorname{argmin}} \left(-\left|\frac{\partial}{\partial\lambda}\varrho_{\boldsymbol{m}}(\phi,\lambda)\right|^2 \right), 0 \le k < i \right\} \tag{12}$$

of possible solutions for the contour instead of one single contour element. Then, also multiple boundary elements, lying on one ray, can be handled, which is necessary, if concave contours shall be tracked.

The principle of any–time behavior of active rays can be summarized as follows. If more than one object should be tracked, or the objects are moving very fast, then increase $\triangle\phi$ of the angle ϕ. If you have more time or you need a more accurate contour representation, reduce $\triangle\phi$ by filling some lost angles into the active ray. For example, start with the angles $\pi/2$, π, $3/2\pi$ and 2π. Then, if time is remaining, add the angles $i/4\pi, i \in \{1,3,5,7\}$, then the angles $i/8\pi, i \in \{1,3,5,7,9,11,13,15\}$, etc.

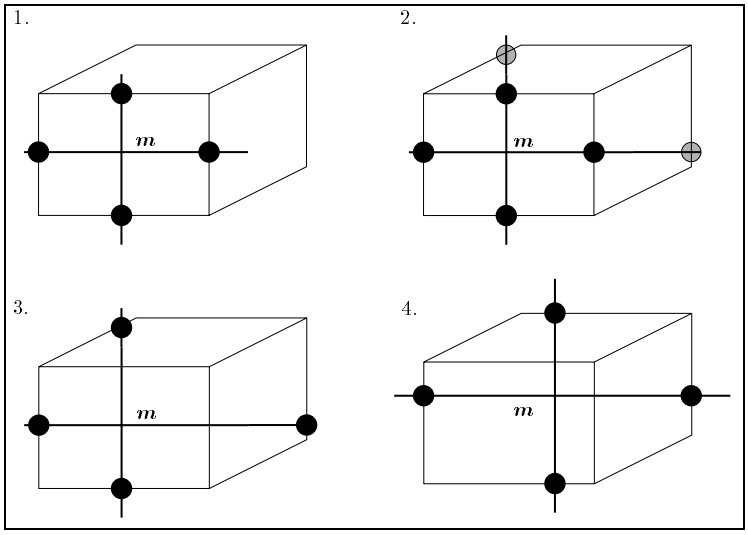

Figure 8: Principle of hypotheses management for contour points.

3 Active Rays for Object Tracking

In the previous section an introduction and a formal definition of active rays, together with an energy minimization scheme has been given. For object tracking several aspects have to be examined: The initialization of an active ray, the strategy for choosing a suitable number and directions of the active rays, as well as a strategy for managing multiple hypotheses. Finally to reduce the computation time certain search intervals have to be selected. Some of these points will be discussed in the following, some others will be subject to our future work.

To initialize an active ray, one point within the contour of the moving object must be found. This can be done by using a static camera and computing the difference image. The center of gravity of the difference is well suited as a first reference point $\boldsymbol{m}$. Also, some information about the size of the moving object in different directions can be computed.

Shooting an active ray from this reference point may result in finding strong edges within this object which are not contour edges (step 1 in Figure 8). This is especially a problem, if no information about the size of the object is available. Then, one possible approach is to let the active ray grow for each image, until a new contour point is found which satisfies (2) (step 2 in Figure 8). If this new hypothesis within the next images is verified, take it as the new contour element, update the reference point $\boldsymbol{m}$, and search for the next hypothesis (step 3 and 4 in Figure 8). Due to the lack of space the verification step cannot be described any further in this paper.

For each active ray $\varrho_{\boldsymbol{m}}(\phi, \lambda)$ and the contour element $\boldsymbol{v}_{\boldsymbol{m}}(\phi)$ one can define a search interval $I(\phi)$ for λ for the next image. This search interval can be computed by a prediction step, or it may depend on $\lambda(\phi)$ of the previous image or the neighboring elements. So, only a small part of $\varrho_{\boldsymbol{m}}(\phi, \lambda)$ must be examined to find the maximum, and equation (2)

Figure 9: Results for tracking a car on a highway with active rays (images 4, 24, 44, 64, 84, 104 of a sequence of 123 images taken at video rate): the sampling step size $\triangle\phi$ is $\pi/18$.

gets

$$\lambda(\phi) = \underset{\lambda \in I(\phi)}{\operatorname{argmin}} \left(- \left| \frac{\partial}{\partial \lambda} \varrho_{\boldsymbol{m}}(\phi, \lambda) \right|^2 \right), \quad 0 \leq \phi < 2\pi. \tag{13}$$

In the introduction the missing ordering of snake elements in the 2D image plane has been mentioned as a problem for predicting the motion of the single contour elements. Only the motion of the complete contour can be predicted. For active rays, one has defined an implicit ordering in the 2D image plane, given by the angle of the rays and the reference point. This can be used, if a 2D contour has been predicted, by estimating the 3D parameters of the moving object. Then, the reference point $\boldsymbol{m}$ of the predicted contour as well as the rays in arbitrary directions can be calculated in advance and verified for the real image data. The results for the real data can then be used to update the predicted object parameters.

4 Experiments and Results

Preliminary experiments for extracting and tracking a contour with active rays have been conducted. Neither multiple hypotheses nor a prediction step has been applied yet. This will be the subject of our work in the near future. All experiments are done off–line.

In Figure 9 results for tracking a moving car on a highway are shown. The sampling step size $\triangle\phi$ has been $\pi/18$. The computation time for extracting the contour for all 123 images of this sequence has been 1.98 sec, i.e., 16 msec/image on a HP 735/99 MHz. The reference point $\boldsymbol{m}$ has been chosen manually in the first image. For all other images the computed center of gravity for the active ray in image t has been taken as the initial reference point for image $t+1$. In Figure 10 the results of the same sequence for $\triangle\phi = \pi/9$ can be seen. Results of the computation time for different sampling rates can be found in Figure 11.

Figure 10: Results for tracking a car on a highway with active rays (images 4, 44, 84 of a sequence of 123 images taken at video rate): the sampling step size $\triangle\phi$ is $\pi/9$.

$\triangle\phi$	time/image (msec)
$\pi/180$	38
$\pi/36$	19
$\pi/18$	16
$\pi/9$	16

Figure 11: Computation time for one image for different sampling step sizes $\triangle\phi$.

5 Conclusion

In our contribution we have presented a new approach to contour tracking, called active rays. The basic ideas come from active contours, which have been proven to be a promising approach to data driven real–time contour tracking.

Active rays have the following advantages over active contour models:

- For active rays an ordering in the image plane is given by a reference point $\boldsymbol{m}$ and an angle ϕ. Thus no crossings occur and predicting the position of the contour elements is possible.
- All optimization problems are reduced to 1D search problems.
- Active rays provide a mechanism, to select the required accuracy of the contour approximation. This leads to an any–time behavior, which is an important aspect of real–time applications.
- Active rays provide a mechanism to manage multiple hypotheses which is useful to detect contours which appear due to changing views of the object.

We have presented a formal description of active rays and an energy formulation for the contour extraction. In addition we have shown the common parts of active contours and active rays. The experiments have proven that this new approach is well suited for an accurate contour extraction and tracking in real–time.

6 Future Research

In Sect. 2 and Sect. 3 we have already mentioned the mechanism of multiple hypotheses, the possibility of an any–time realization and the advantages for the prediction of the contour elements position. These topics are examined in our actual work. We will compare the active ray approach against active contour models on a larger test set, also in a closed–loop real–time application [7].

Finally, another important aspect of any problem in image processing or in general pattern recognition is the ability of self–adaption, or learning. For example, learning 3D models of objects from greylevel images [9], or the deformation of contours in the image plane [10]. Today all state of the art speech recognition systems use a training set to

estimate the parameters of the system. For contour tracking one has to learn the possible deformation of the contour, due to rotation and translation. Then, after inspecting the movement of the object during a training step, one can predict the deformation of the contour even for occlusions. On this aspect a lot of work has to be done. These remarks should only give a coarse idea of how active rays can be trained.

References

1. A. Amini, T. Weymouth, and R. Jain. Using dynamic programming for solving variational problems in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(9):855–867, 1990.
2. A. Blake and A. Yuille, editors. *Active Vision*. MIT Press, Cambridge, Massachusetts, London, England, 1992.
3. C. Brown, D. Coombs, and J. Soong. Real–time smooth pursuit tracking. In A. Blake and A. Yuille, editors, *Active Vision*, pages 123–136. MIT Press, Cambridge, Massachusetts, London, England, 1992.
4. P.A. Couvignou, N.P. Papanikolopoulos, and P.K. Khosla. On the use of snakes for 3–d robotic visual tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 750–751. Ney York City, 1993.
5. R. Curwen, A. Blake, and A. Zisserman. Real–time visual tracking for surveillance and path planning. In G. Sandini, editor, *Computer Vision - ECCV 92*, pages 879–883, Berlin, Heidelberg, New York, London, 1992. Lecture Notes in Computer Science.
6. K. Daniilidis, M. Hansen, C. Krauss, and G. Sommer. Auf dem Weg zum künstlichen aktiven Sehen: Modellfreie Bewegungsverfolgung durch Kameranachführung. In *DAGM 1995, Bielefeld*, pages 277–284, 1995.
7. J. Denzler and H. Niemann. Combination of simple vision modules for robust real–time motion tracking. *European Transactions on Telecommunications*, 5(3):275–286, 1995.
8. C. Harris. Tracking with rigid models. In A. Blake and A. Yuille, editors, *Active Vision*, pages 59–74. MIT Press, Cambridge, Massachusetts, London, England, 1992.
9. J. Hornegger. *Statistische Modellierung, Klassifikation und Lokalisation von Objekten*. Dissertation, Technische Fakultät, Universität Erlangen–Nürnberg, Erlangen, 1996.
10. K.F. Lai and R.T. Chin. Deformable contours: Modelling and extraction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(1):1–20, 1996.
11. N.P. Papanikolopoulos, P.K. Khosla, and T. Kanade. Visual tracking of a moving target by a camera mounted on a robot: A combination of control and vision. *IEEE Transactions on Robotics and Automation*, 9(1):14–34, 1993.
12. R. Ronfard. Region-based strategies for active contour models. *International Journal of Computer Vision*, 13(2):229–251, 1994.
13. S. Venkatesan and C. Archibald. Use of wrist mounted range profile scanners for real-time tracking. *Machine and Vision Applications*, 5:1–16, 1992.
14. C. Vieren, F. Cabestaing, and J. Postaire. Catching moving objects with snakes for motion tracking. *Pattern Recognition Letters*, 16:679–685, 1995.
15. D.J. Williams and M. Shah. A fast algorithm for active contours and curvature estimation. *Computer Vision, Graphics, and Image Processing*, 55(1):14–26, 1992.
16. G. Xu, E. Segawa, and S. Tsuji. Robust active contours with insensitive parameters.

Pattern Recognition, 27(7):879–884, 1994.

17. N. Yokoya and S. Araki. Splitting contour models based on crossing detection. In *Proceedings of the 1995 Real World Computing Symposium*, pages 29–30, Tokyo, 1995.