MODEL–FREE, STATISTICAL DETECTION AND TRACKING OF MOVING OBJECTS

Mark Ross

University of Koblenz
Institute of Computational Visualistics
Koblenz, Germany

ABSTRACT

A novel statistical approch for detection and tracking of objects is presented here, which uses both edge and color information in a particle filter. The approach does not need any prior models of the objects of interest or of the scene. It starts with homogenous regions as tracking primitives and creates complex objects by merging similar moving regions. Even partially occluded objects in a sequence captured by a moving camera can be tracked efficently and robust.

1. INTRODUCTION

Tracking of moving objects, e. g. people or vehicles, in an image sequence is one of the basic tasks in computer vision. The resulting trajectory or the course of the object state can be either of interest in its own or used as the input for a higher level analysis. Applications include surveillance and recognition systems and advanced human-computer interaction. However, designing robust tracking algorithms is difficult, requiring mechanisms to deal with problems like background clutter, discontinuous motion, multiple and occluding objects, and many others.

In the recent years many different approaches [1, 2, 3] have been developed; the most interesting one seems to be the Condensation or Particle Filter algorithm [4], which has been used and extended many times [5, 6, 7, 8, 9].

Particle Filtering [4] was developed to track objects in clutter, in which the posterior density and the observation density are often non-Gaussian. The key idea of particle filtering is to approximate the probability distribution by a weighted sample set. Each sample consists of an element which represents the hypothetical state of an object and a corresponding probability. The state of an object may be control points of a contour [4], the position, shape and motion of an elliptical region [6], or specific model parameters [7].

Here we present an extension of the Particle Filter which uses both contour and color information as object state for tracking. In contrast to [5], who also use color and edges, our approach is completely free of a-priori models. It initializes objects automatically by the use of a color segmentation [10]. Although objects in general consist of more than one segment, the particle filter starts with tracking of single segments as tracking primitives. After establishing the motion of these segments, similar moving segments are merged to an object, which can be tracked in the same way as single segments.
Our method is based on the following assumptions:

- Each object must be visible and distinguishable in color from adjacent background parts in the initial frame.
- Objects are approximately rigid, so a motion model of affine transformation can be applied.
- There are only slight color changes of objects in successive frames.

2. NEW APPROACH

As our approach is free of a-priori models, it starts with tracking of segments as tracking primitives, which are merged to complex objects after their motion is established. A segment S is a topologically connected set of pixels, which are similar in color. So a segmentation $\mathcal{S}_t$ is a partitioning of a frame F_t at time t into segments. Here we use the nonlinear filter of [11] and the CSC color segmentation method [10].

2.1. Template model

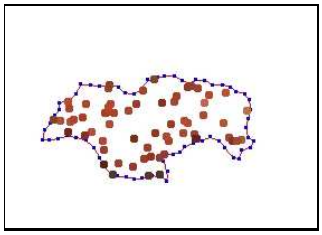
The template $q(S) = (E(S), C(S))$ of a segment S consists of a set $E(S)$ of edge points and a set $C(S)$ of color points and is the representation of segment S used for the tracking. The set $E(S) \subset \mathbb{N}^2$ of edge points is obtained by equidistant sampling of the outer border of segment S; the set $C(S) \subset \mathbb{N}^2 \times [0, 255]^3$ of uniformly distributed color sample points (position and RGB-color) is obtained by random sampling of all pixels belonging to S. Describing the color of a segment by a set of sample points is more accurate than its mean color as in [1] or a color histogram as in [6], because it contains the spatial distribution of color. Fig. 1 shows an example of the view q of a single segment.

2.2. Motion model

Our motion model is an affine transform $M(\boldsymbol{m}) \in \mathbb{R}^{3,3}$, $\boldsymbol{m} \in \mathbb{R}^6$, with six degrees of freedom which describe translation and rotation in the image plane and horizontal, verti-

13th International Conference on Image Processing (ICIP 2006), Atlanta, GA, USA, oct. 8-11, 2006

(a) Original image, segment 'car body' is bordered white



(b) Template of the segment 'car body'

Fig. 1. Representation of a single segment as a set of contour points and a set of color sample points.

cal, and diagonal scaling [12]. An isotrop scaling is the projective effect of changing the objects distance to the camera; an anisotrop scaling may describe a rotation out of the image plane. A sequence of motion vectors $[\boldsymbol{m}]^n = \boldsymbol{m}_1, \boldsymbol{m}_2, \ldots, \boldsymbol{m}_n$ is called a trajectory of length n. The transformation of a location $\boldsymbol{x} \in \mathbb{R}^2$ by a trajectory $[\boldsymbol{m}]^n$ is a simple matrix multiplication $M(\boldsymbol{m}_n) \cdot \ldots \cdot M(\boldsymbol{m}_1) \cdot \binom{\boldsymbol{x}}{1}$ in homogenous coordinates.

Now, motion is modelled in stochastic terms, so a motion is not fact, but has a certain probability. The (sampled) probability distribution over the space of motions is represented as a particle set P, which is a set of motions. A motion is a tuple $([\boldsymbol{m}], \omega)$ of a trajectory $[\boldsymbol{m}]$ and a weight $\omega \in [0, 1]$ (probability) with

$$\sum_{([\boldsymbol{m}],\omega)\in P} \omega = 1. \tag{1}$$

An initial trajectory $[\boldsymbol{m}]^1 = \boldsymbol{m}_1$ is created by random sampling of a six-dimensional normal distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma}^2)$ with mean $\boldsymbol{\mu} = (0, 0, 0, 1, 1, 0)$ and variance $\boldsymbol{\sigma}^2 \in \mathbb{R}^6$.

2.3. Tracking

A tracking object $o = (q, P)$ is a tupel of a template q and a particle set P, which describes the motion. In a future extension of this approach an object will have a couple of templates (or 'views'), each from a different time and with a certain trustiness.

The tracking of an object $o_{t-1} = (q, P_{t-1})$ at time $t-1$ is done with the Condensation algorithm [4], which includes the four steps

1. Select a Motion: Sample a motion $[\boldsymbol{m}]^{n-1}$ from P_{t-1}.

2. Predict: Calculate a new trajectory $[\boldsymbol{m}]^n$ based on $[\boldsymbol{m}]^{n-1}$.

3. Measure: Evaluate the view q with trajectory $[\boldsymbol{m}]^n$ depending on the current Frame F_t by a weight ω. After normalization of the sum of all weights $([\boldsymbol{m}]^n, \omega)$ will form a new motion in P_t.

4. Normalize: Normalize the weights of P_t, so that they sum to 1.

The prediction of a trajectory $[\boldsymbol{m}]^{n-1}$ has a deterministic part, which depends on the last motion vectors of $[\boldsymbol{m}]^{n-1}$ (e. g. the mean of them), and a stochastic drift, which is random.

The weight ω or quality of a view $q = (C, E)$ with trajectory $[\boldsymbol{m}]$ is composed by a color error e_C and an edge error e_E,

$$\omega = \exp -\frac{e_E}{2\sigma_E^2} \cdot \exp -\frac{e_C}{2\sigma_C^2}. \tag{2}$$

The color error $e_C \in [0, \check{e}_C]$ of a color sample point set $C \in \mathbb{R}^2 \times [0, 255]^3$ and a trajectory $[\boldsymbol{m}]$ is the mean squared difference of the colors $\boldsymbol{c}$ of C to the color in the current frame F_t at location $[\boldsymbol{m}] \cdot \boldsymbol{x}$ for all $(\boldsymbol{x}, \boldsymbol{c}) \in C$, where $[\boldsymbol{m}] \cdot \boldsymbol{x}$ denotes the position $\boldsymbol{x}$ transformed by $[\boldsymbol{m}]$,

$$e_C = \frac{1}{|C|} \sum_{(\boldsymbol{x},\boldsymbol{c})\in C} \min\left(\check{e}_C, (\boldsymbol{c} - F_t([\boldsymbol{m}] \cdot \boldsymbol{x}))^2\right) \tag{3}$$

where $\check{e}_C \in \mathbb{R}$ is a parameter.

The edge error $e_E \in [0, \check{e}_E]$ of an edge point set $E \in \mathbb{R}^2$ and a trajectory $[\boldsymbol{m}]$ is the mean squared distance of the transformed edge points $[\boldsymbol{m}] \cdot \boldsymbol{x}$, $\boldsymbol{x} \in E$, to the true edges in the current frame F_t and is calculated as

$$e_E = \frac{1}{|E|} \sum_{\boldsymbol{x}\in E} \min\left(\check{e}_E\,,\, D_t^2([\boldsymbol{m}] \cdot \boldsymbol{x})\right) \tag{4}$$

where $D_t([\boldsymbol{m}] \cdot \boldsymbol{x})$ is the distance of position $\boldsymbol{x}$ transformed by the trajectory $[\boldsymbol{m}]$ to the nearest edge in image F_t and $\check{e}_E \in \mathbb{R}$ is a parameter. The distance image $D : \mathbb{N}^2 \to \mathbb{R}$ is obtained by applying the distance transform on the segmentation based edge image of F_t, so the borders of all segments (of a minimum size) form the edge image. In comparison to [4] the distances are determined without any seeking, which leads to an efficient algorithm.

2.4. Complex objects

After tracking single segments (objects consisting of one segment), complex objects are created by merging 'statistically similar' moving objects, which are close to each other.

The similarity of two objects $o = (q, P)$ with $q = (E, C)$ and $o' = (q', P')$ with $q' = (E', C')$ is defined as

$$sim(o, o') = \sum_{\substack{([\boldsymbol{m}],\omega)\in P,\\ ([\boldsymbol{m}'],\omega')\in P'}} \omega \cdot \omega' \cdot sim([\boldsymbol{m}], [\boldsymbol{m}']) \cdot near(E, E'). \tag{5}$$

with spatial closeness $near(E, E')$ and similarity $sim([\boldsymbol{m}], [\boldsymbol{m}'])$ defined below.

The merging can be regarded as finding subgraphs in a graph, where each object is a node and with edges between nodes, iff the similarity of the respective objects is greater

than a certain threshold. Each maximum subgraph with more than one node forms a new complex object.

Let E, E' be two sets of edge points. The spatial closeness $near(E, E') \in [0, 1]$ is

$$near(E, E') = 1 - \max(0, \min(1, near'(E, E'))) \quad (6)$$

where

$$near'(E, E') = \frac{\left(\min_{\boldsymbol{x} \in E, \boldsymbol{x}' \in E'} (\boldsymbol{x} - \boldsymbol{x}')^2\right) - d_{\mathrm{Min}}}{d_{\mathrm{Max}} - d_{\mathrm{Min}}} \quad (7)$$

with parameter $d_{\mathrm{Min}}, d_{\mathrm{Max}} \in \mathbb{R}$, $0 \leq d_{\mathrm{Min}} < d_{\mathrm{Max}}$. Let

$$\overline{\boldsymbol{x}}_i = \frac{1}{3}(\boldsymbol{x}_i + \boldsymbol{x}_{i-1} + \boldsymbol{x}_{i-2}) \text{ with } \boldsymbol{x}_j = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix} \cdot \boldsymbol{m}_j \quad (8)$$

be the smoothed translation of a trajectory $[\boldsymbol{m}]^n = \boldsymbol{m}_1, \boldsymbol{m}_2, \ldots, \boldsymbol{m}_n$ at position i, $3 \leq i \leq n$. The similarity $sim([\boldsymbol{m}], [\boldsymbol{m}']) \in [0, 1]$ of two trajectories $[\boldsymbol{m}]^n$, $[\boldsymbol{m}']^n$ of length n with smoothed translations $\overline{\boldsymbol{x}}_i$ and $\overline{\boldsymbol{x}}'_i$, $3 \leq i \leq n$, is

$$sim([\boldsymbol{m}], [\boldsymbol{m}']) = 1 - \max(0, \min(1, sim'([\boldsymbol{m}], [\boldsymbol{m}']))) \quad (9)$$

where

$$sim'([\boldsymbol{m}], [\boldsymbol{m}']) = \frac{\frac{1}{n-2} \sum_{i=3}^{n} (\overline{\boldsymbol{x}}_i - \overline{\boldsymbol{x}}'_i)^2 - s_{\mathrm{Min}}}{s_{\mathrm{Max}} - s_{\mathrm{Min}}} \quad (10)$$

with parameter $s_{\mathrm{Min}}, s_{\mathrm{Max}} \in \mathbb{R}$, $0 \leq s_{\mathrm{Min}} < s_{\mathrm{Max}}$.

As a complex object is the aggregation of n objects $o_i = (q_i, P_i)$ with $q_i = (E_i, C_i)$, $1 \leq i \leq n$, it has a template $q = (E, C)$ of contour and colour sample points, where $E = E_1 \cup \ldots \cup E_n$ and $C = C_1 \cup \ldots \cup C_n$. So it can be tracked in the same way as an object consisting of only one segment.

3. RESULTS

Although the development of our system is in the beginning, it shows robust and efficient results.

Only the initialization is based on segmentation, not the tracking itself. Fig. 2 shows the successful tracking of two objects which are similar in color and touch or occlude each other. A segmentation based tracker like [1] would fail here as the segmentation would merge the objects.

As the approach does not use any kind of difference frame techniques it is able to track objects captured with a moving camera, see Fig. 3 and Fig. 4.

Processing the first frame of the sequence of Fig. 3 with an image size 340×275 which includes filtering, segmentation and calculating the contour and colour sample points of the segments takes about 350 ms on a standard office PC[1]; all this can be speed up by parallel processing. Tracking of objects in the following frames with the particle filter takes about 350 ms, too. As tracking of an object works independently from the others, processing could easily be distributed on multi processors. So a processing in real-time is possible.

[1] i686

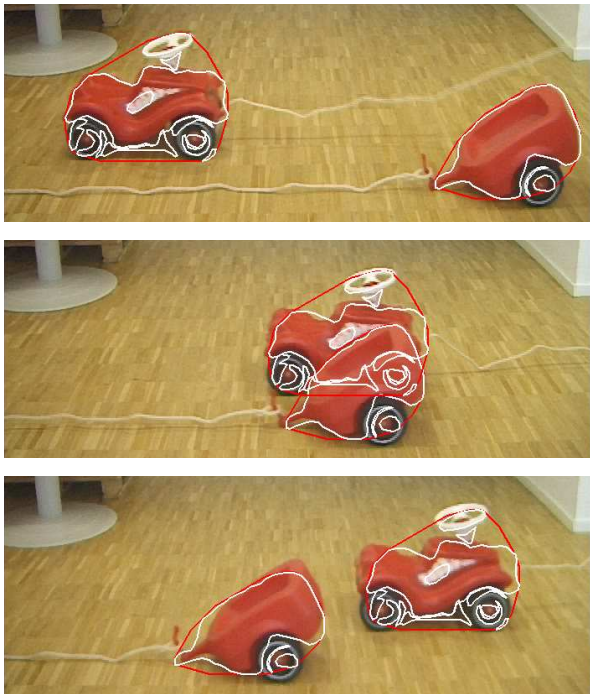


Fig. 2. Sequence 'Bobby-Car', tracking of two objects with identical color and partial occlusion.

4. SUMMARY AND FUTURE PROSPECTS

We presented a statistical object tracker, which is able to track moving objects captured with a moving camera (see Fig. 3). The new ideas of our approach are

- using both color and edges for the tracking template,
- automatic initialization of objects without a-priori models,
- using color sample points instead of histograms,
- and combining tracking primitives to complex objects.

This leads to an efficient and robust system, which is able to track multiple objects in clutter and handles partial occlusions (see Fig. 2) without the need of any a-priori scene or object model.

The main problem of this approach is the lack of any reinitialization of segments. So the tracker can not detect objects, which are not visible in the initial frame. The second problem is, that there is no adaption of the template of an object to the image data; so, if the perspective or illumination changes too much, the tracker would loose its objects. The solution of these problems is to expand the objects to store a couple of views as templates for the tracking, where each view will have a certain trust. So the tracking algorithm will sample from the set of motions and indepently from the set of views. This allows the tracker to use either a newer view which might be occluded or an unoccluded but older view. A reinitialization of objects through a segmentation of new images can be integrated into this extension which will lead to an adaption of the view of objects to the current image.

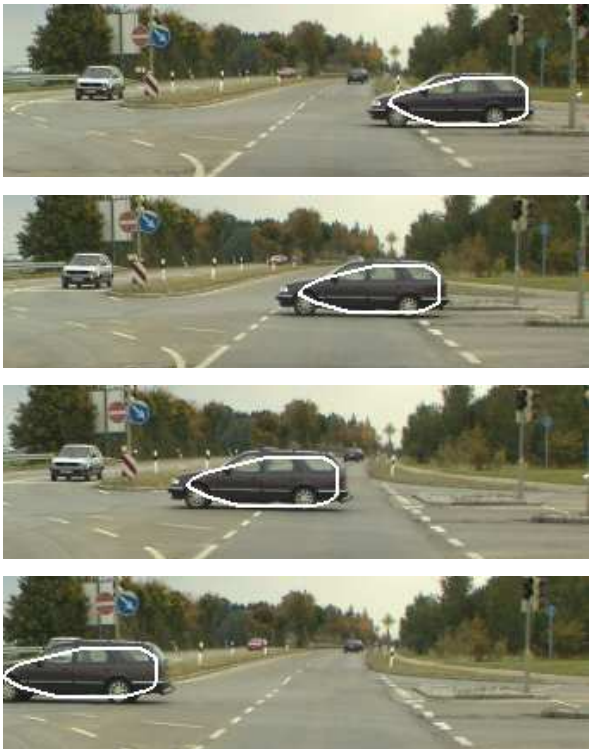
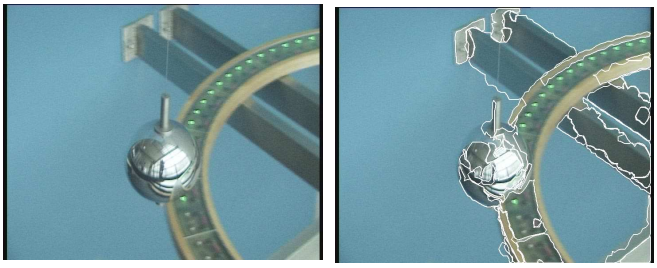
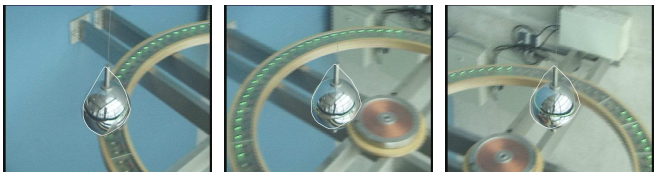


Fig. 3. Natural scene taken with a moving camera



(a) First Frame, original image

(b) First Frame, segments of medium size



(c) Frame 5

(d) Frame 25

(e) Frame 45

(f) Frame 65

(g) Frame 85

(h) Frame 105

Fig. 4. Example of successful tracking: Foucault Pendulum at university of Koblenz, captured with a moving camera (a) first frame, (b) first frame with segment borders in white, (c-h) tracked sphere, the convex hull is outlined white.

5. REFERENCES

[1] Volker Rehrmann, "Object oriented motion estimation in color image sequences," in *ECCV 1998, Vol. I, LNCS 1406*, 1998, pp. 704–719.

[2] Bernt Schiele, "Model-free tracking of cars and people based on color regions," in *Proceedings 1st IEEE Int. Workshop on PETS, Grenoble, France*, Mar. 2000.

[3] Chris Stauffer and W.E.L Grimson, "Adaptive background mixture models for real-time tracking," in *CVPR*, 1998.

[4] Michael Isard and Andrew Blake, "CONDENSATION – conditional density propagation for visual tracking," *Int. Journal on Computer Vision*, vol. 1, no. 29, pp. 5–28, 1998.

[5] Michael Isard and Andrew Blake, "ICONDENSATION: Unifying low-level and high-level tracking in a stochastic framework," *Lecture Notes in Computer Science*, vol. 1406, pp. 893–908, 1998.

[6] Katja Nummiaro, Esther Koller-Meier, and Luc Van Gool, "A Color-Based Particle Filter," in *1st Int. Workshop on Generative-Model-Based Vision GMBV'02, in conjuction with ECCV'02*, 2002, pp. 53–60.

[7] David Tweed and Andrew Calway, "Tracking Many Objects Using Subordinated Condensation," in *The 13th British Machine Vision Conference (BMVC 2002)*, 2002.

[8] P. Perez, C. Hue, J. Vermaak, and M. Gangnet, "Color-based probabilistic tracking," in *ECCV 2002, LNCS 235, Springer*, 2002, vol. LNCS 2350, p. 661.675.

[9] Zia Khan, Tucker Balch, and Frank Dellaert, "An MCMC-based Particle Filter for Tracking Multiple Interacting Targets," in *Eur. Conference on Computer Vision 2004 (ECCV 2004)*, 2004.

[10] Volker Rehrmann and Lutz Priese, "Fast and robust segmentation of natural color scenes," in *Proc. of 3rd Asian Conf. on Computer Vision, Special Session on Advances in Color Vision*. 1998, vol. I, pp. 704–719, Springer Verlag.

[11] M. Nagao and T. Matsuyama, "Edge preserving smoothing," in *Computer Graphics and Image Processing*, 1979, vol. 9, pp. 394–407.

[12] Andrew Blake, Michael Isard, and David Reynard, "Learning to track the visual motion of contours," *Artificial Intelligence*, vol. 78, no. 1-2, pp. 179–212, 1995.