Tracking the Invisible: Learning Where the Object Might be

Helmut Grabner[1] Jiri Matas[2] Luc Van Gool[1,3] Philippe Cattin[4]

[1]Computer Vision Laboratory
ETH Zurich
{grabner, vangool}@vision.ee.ethz.ch

[2]Center for Machine Perception
Czech Technical University, Prague
matas@cmp.felk.cvut.cz

[3]ESAT - PSI / IBBT
K.U. Leuven
luc.vangool@esat.kuleuven.be

[4]Medical Image Analysis Center
University of Basel
philippe.cattin@unibas.ch

Abstract

Objects are usually embedded into context. Visual context has been successfully used in object detection tasks, however, it is often ignored in object tracking. We propose a method to learn supporters *which are, be it only temporally, useful for determining the position of the object of interest. Our approach exploits the General Hough Transform strategy. It couples the supporters with the target and naturally distinguishes between strongly and weakly coupled motions. By this, the position of an object can be estimated even when it is not seen directly (*e.g.*, fully occluded or outside of the image region) or when it changes its appearance quickly and significantly. Experiments show substantial improvements in model-free tracking as well as in the tracking of "virtual" points,* e.g.*, in medical applications.*

1. Introduction

The fact that context helps in object detection is well-known. For instance, it has been repeatedly reported [7, 13] that one of the strongest predictors of vehicle presence and location in an image is the shadow it casts on the road. Similarly, many parts-whole relations have been exploited by detectors, *e.g.*, of a face *vs*. facial parts [4]. In detection, only stable, long-term and statistically significant object-context relationships are easily incorporated, *e.g.*, [12, 3].

In tracking, many temporary, but potentially very strong links exist between the tracked object and the rest of the image. Consider, for instance, the image in Fig. 1a of a skipping stone. The ripples on the surface not only strongly constrain the location of the object of interest – the stone – but also allow for a fairly good prediction of its future trajectory.

To exploit context for tracking, we propose to incrementally learn a model inspired by the Implicit Shape Model (ISM) [8], where local image features vote for the object positions. The core idea is depicted in Fig. 2. First, local image features from the whole image are extracted (yellow points). Given the position of the object of interest in the frame, these image features are usually divided into object points and points belonging to the background (see Fig. 2b) [9]. Object points lie on the object surface and thus always have a strong correlation to the object motion (green points). Background points, *e.g.*, points on other independently moving objects or in the static background, are considered to carry no information about the object position (blue points). Instead of this typical, binary distinction, we propose the concept of Supporters.

(a) What happens with the stone? (b) Where is the soccer ball?

Figure 1. *Supporters* come with different forms, durations of existence and predictive strengths. For example, the trajectory of the stone can be reconstructed from a single image (a).

Supporters are features which are useful to predicting the target object positions. They at least temporarily move in a way which is statistically related to the motion of the target (red points). A supporter can be very strong (comparable to an object feature), *e.g.*, a wristwatch on a hand holding the target; or quite weak when the coupling with the target motion is not that outspoken. The goal of our algorithm is, in other words, to find such local image features which help to predict the position of the target. A simple but highly effective method based on the Generalized Hough Transform

978-1-4244-6985-7/10/$26.00 ©2010 IEEE

(a) (b) (c)

Figure 2. (a) A frame with the target object marked. (b) *Supporters* are features that vote for the position of the object, since their motion appears correlated. They can belong to the object itself (green) or not (red). Uncorrelated features (blue) are discarded. (c) Even if the object cannot be tracked based on its appearance (*e.g.*, it gets occluded or changed its appearance), the supporters can help infer its position.

(GHT) is proposed for maintenance of the supporter set, *i.e.*, for associating and disassociating features with the tracked object.

The most closely related work is that of Cerman *et al*. [2], who improve object tracking by using a single "companion" region close to the target. The companion has a 2D affine coupling to the target. Our approach is inspired by the companion idea, but is more general in a number of important ways. We continuously estimate, update, and refine a model which predicts the position of the target. This model consists of the supporters, which carry information about the target position. In contrast to Cerman *et al*.'s companion, the motion coupling between supporters and target can be non-rigid. Multiple supporters can be activated at the same time. The supporters need not be confined to a region surrounding or next to the target. As a matter of fact, also a set of spatially dispersed, local patches with different types of motion couplings are allowed to drive the process. Moreover, our approach can re-detect the target and is more flexible in updating the model of its appearance. It can also deal with moving cameras, as it does not depend on a static background assumption as was the case in Cerman *et al*.

The remainder of the paper is organized as follows. In Sec. 2 we propose our supporter approach. Implementation details are presented in Sec 3. Sec. 4 discusses experimental results, including a medical application. Sec. 5 concludes the paper.

2. Learning the Motion Couplings

It takes very little for people to correctly infer the subsequent positions of moving objects. For example, let us consider Fig. 3. When trying to track a specific balloon, it is absolutely not necessary for it to be in view to still have a rather precise idea of where it is. Its relation to other objects, moving with it, suffices. In the example, as long as even only the basket below the balloon is visible, one still can make a good prediction. This coupled motion would also help to track the balloon even if its appearance changed rapidly. This type of inference from motion context is what we aim for in this paper.

Our approach combines the principles of (i) including context for tracking; (ii) establishing an object model on-the-fly; and (iii) exploiting the flexibility of local image features for object detection and tracking.

2.1. Problem Formulation

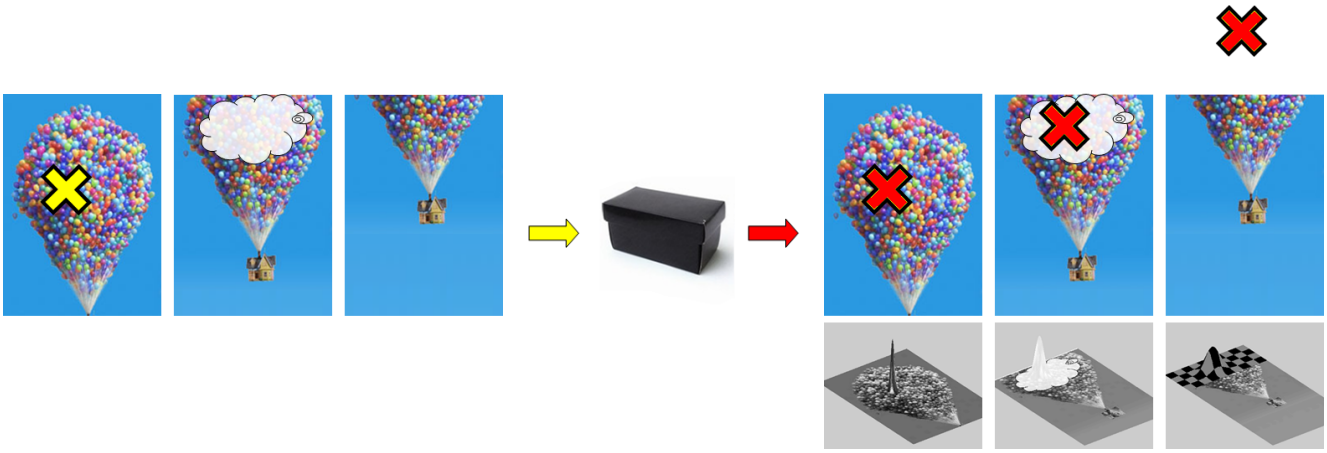
We want to learn a model of $P(\mathbf{x}|I)$, predicting the position $\mathbf{x}$ of an object[1] in the image I. The objective is to learn this model from a few, reliable measurements, *i.e.* from images where we feel confident about the target object positions. The model should then generalize enough to drive the tracker, also for images where the target position is less obvious. The supporter context will be part of the model, so that the object need not even be visible.

Implicit Shape Model for Object Detection. Local image features have been shown to be a powerful tool for specific object detection (*e.g.*, [10]). Furthermore, the Generalized Hough Transform (GHT) has been successfully combined with local features for object class detection. An example is [8], where an off-line stage learns a codebook of local features. Next, the Implicit Shape Model (ISM) of an object class is learned. After training from a large database of labeled images, the model can be used to detect objects from that object class in test images. First, features are extracted, which are matched to the codebook. Each feature subsequently casts probabilistic votes for possible object positions, where the hypothesis score is obtained as a sum over all votes. The score function S of the ISM is defined as a probability density over the object position $\mathbf{x} = (x, y)$ in the image I, *i.e.*

$$P(\mathbf{x}|I) \propto S = \sum_{\mathbf{f}\in\mathcal{F}} P(\mathbf{x}|\mathbf{f})P(\mathbf{f}|I), \qquad (1)$$

where $\mathcal{F}$ is a set of features. The indicator functions $P(\mathbf{f}|I)$ specifies if feature $\mathbf{f}$ is found in the image I, which then

[1] We currently only consider one target.



(a) input sequence (b) learning algorithm (c) output sequence

Figure 3. (a) Given an input sequence with some labeled data like the position a specific target balloon (yellow cross), (b) a model is learned in order to predict the object positions. (c) The model has to generalize such that the positions (red crosses) can be correctly predicted also in the other images, even when not visible (estimated positions are represented by an underlying probability distribution).

votes for the object position $\mathbf{x}$ by $P(\mathbf{x}|\mathbf{f})$. By analyzing this voting space, *i.e.* by finding its peaks, objects can be detected.

2.2. Learning the Supporter Model

Given the strengths of ISM and the fact that local features have been successfully used for object tracking as well (*e.g.* [9]), it stands to reason to combine these ideas. We will use local features, as in ISM, to predict the target object position. In contrast to regular ISM, we continuously update the pool of contributing features as well as their coupling to the target position.

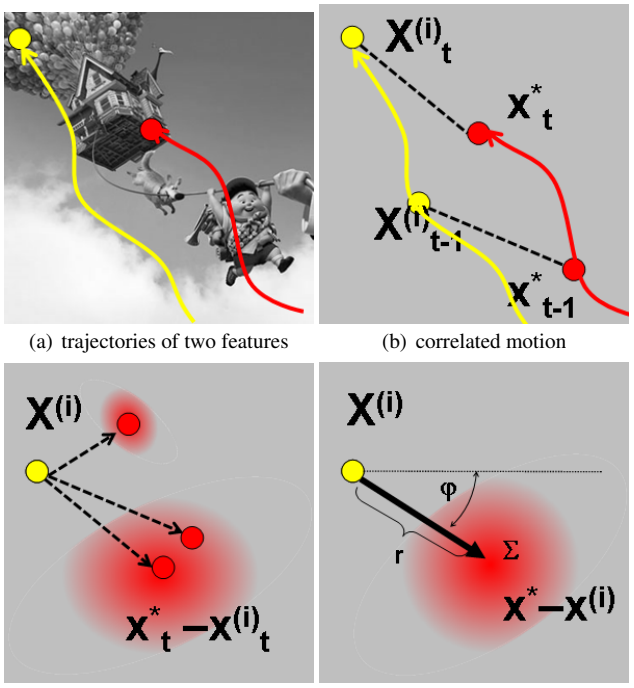
Suppose we have a feature point $\mathbf{x}^{(i)} = (x^{(i)}, y^{(i)})$ associated with a feature $\mathbf{f}^{(i)}$ and the position $\mathbf{x}^\star = (x^\star, y^\star)$ of the target. We want to find such features that can act as strong predictors for $\mathbf{x}^\star$. In general, the trajectory of such features must be coupled with the target by some motion model (see Fig. 4a-b). One might consider arbitrary complex motion models here but given the nature of the ISM, it is assumed that the relative position of feature and target is more or less fixed over short time intervals (see Fig. 4c).

Learning the model can now be made quite simple. For frames with good visibility of the target or with human localization in case of an interactive application, we estimate the indicator functions of the ISM model (Eq. (1)) for each detected feature $\mathbf{f}$ in the image. This includes estimating the relative position $\bar{\mathbf{x}} = \mathbf{x}^\star - \mathbf{x}^{(i)}$ of the target with respect to the feature position $\mathbf{x}^{(i)}$. This should be done in an on-line manner, since it might change over time. Hence we use the exponential forgetting principle, *i.e.*,

$$P_t(\mathbf{f}|I) \propto \alpha\, P_{t-1}(\mathbf{f}|I) + (1-\alpha)\, p(\mathbf{f}|I_t), \quad (2)$$

$$P_t(\bar{\mathbf{x}}|\mathbf{f}) \propto \alpha\, P_{t-1}(\bar{\mathbf{x}}|\mathbf{f}) + (1-\alpha)\, p(\mathbf{x}_t^\star - \mathbf{x}_t^{(i)}|\mathbf{f}), \quad (3)$$

where parameter $\alpha \in [0,1]$ determines the weighting of the past estimates. The current image I_t and the object position $\mathbf{x}_t^\star$ is included using $p(\mathbf{f}|I_t)$, the indicator of feature $\mathbf{f}$ in the current image; and $p(\mathbf{x}_t^\star - \mathbf{x}_t^{(i)}|\mathbf{f})$, the current relative object position.



(a) trajectories of two features (b) correlated motion

(c) GHT (d) simplification

Figure 4. (a,b) Estimating the coupling between the motions of two points, $\mathbf{x}^\star$ and $\mathbf{x}^{(i)}$ (c) using the GHT voting principle to describe the coupling, and (d) an approximation to the GHT distribution with a single Gaussian.

We focus on coupled motions, and discard situations where the supporter and target stand still. Hence, we only update the indicator functions when the object position $\mathbf{x}^\star$ or the corresponding local image feature position $\mathbf{x}^{(i)}$ has changed, *i.e.*,

$$||\mathbf{x}_t^{(i)} - \mathbf{x}_{t-1}^{(i)}||_2 > \theta_{mov} \;\vee\; ||\mathbf{x}_t^\star - \mathbf{x}_{t-1}^\star||_2 > \theta_{mov}, \quad (4)$$

where θ_{mov} corresponds to a small threshold.

Evaluation. The model built as described above is then used to determine the position of the object using Eq. (2) and Eq. (3) in Eq. (1) when the object can not be tracked directly, *i.e.*,

$$P(\mathbf{x}|I_t) \propto S_t = \sum_{\mathbf{f}\in\mathcal{F}} P_t(\mathbf{x}|\mathbf{f})\ P_t(\mathbf{f}|I_t). \tag{5}$$

where $\mathbf{x} = \bar{\mathbf{x}} + \mathbf{x}^{(i)}$ stands for the position in the image.

3. Practical Implementation

In the following we describe implementation aspects of our proposed approach. Our design decisions (approximations) were mainly motivated to speed up the tracking process while achieving good results. The overall algorithm is sketched in Alg. 1.

Image features. Although any image features could be used, our implementation uses Harris points [5] and describe them by a SIFT inspired descriptor [10]. In fact, we concatenate 8 bin orientation histograms, calculated on a 3×3 grid on an extracted 30×30 image patch around the found Harris point. Additionally, we estimate the main orientation φ_0 using the maximum orientation of a 16 bin orientation histogram calculated on the image patch.

Voting. Using the GHT as description is powerful, since it makes a natural distinction between strongly and weakly coupled motions, based on how peaked these are. However, given the limited data available, we approximate the voting. In fact, we approximated the voting of a feature $\mathbf{f}$ with a single Gaussian (see Fig. 4d), *i.e.*,

$$P(\mathbf{x}|\mathbf{f}) \propto \frac{1}{\sqrt{2\pi|\mathbf{\Sigma}|}}\ \exp\left(-0.5\ (\mathbf{x}-\boldsymbol{\mu})^{\mathsf{T}}\ \mathbf{\Sigma}^{-1}\ (\mathbf{x}-\boldsymbol{\mu})\right) \tag{6}$$

Where $\boldsymbol{\mu}$ is the mean, stored using polar coordinates (relative angle φ, with respect to the main orientation φ_0, and distance r)[2] and $\mathbf{\Sigma}$ is the covariance matrix.

Model. We store all detected feature points in a database $\mathcal{DB}$. The entries $(\mathbf{d}, r, \varphi, \mathbf{\Sigma})$ store the descriptor and the estimated voting vector, encoded by radius, angle and covariance matrix, respectively.

Feature matching. To estimate $P(\mathbf{f}|I)$, the feature $\mathbf{f}$ is matched with the database $\mathcal{DB}$ of already stored local features. If the best match is below a manually set threshold θ_{match}, it is considered to be a matching point, *i.e.*,

$$\max_{\mathbf{d}\in\mathcal{DB}}(\mathbf{f}^{\mathsf{T}} \cdot \mathbf{d}) > \theta_{match} \tag{7}$$

[2]Note, in our implementation the scale is fixed. However, this can be included in a straight forward manner.

Algorithm 1: Determination of Position

```
- init model, DB = {}
while run do
    - detect and track local image features
    - find best matches with current model (Eq. (7))
    if x* available then
        LEARNING THE MODEL
        if object has moved (Eq. (4)) then
            foreach matched feature (Eq. (7)) do
                - update voting vector, i.e. φ and r
                (Eq. (8) and Eq. (9))
                - update Σ (Eq. (11))
            end
            foreach non matched feature do
                - add key-point to DB
                - init voting vector, i.e. φ and r
                (Eq. (10))
                - init Σ = σ_0 I_2
            end
        end
    else
        APPLYING THE MODEL
        - get P(x|I) using all matches (Eq. (5))
        if P(x|I) is very confident then
            - build second level supporters (see text)
        end
    end
end
```

where we use the dot-product of the normalized feature descriptors (similar to [10]). If no match could be established, the point is added to the database and becomes a supporter.

Tracking. In addition to detecting local image features in each frame, we also use simple KLT tracking [1] for establishing feature matches from two successive image frames.

Updates. In the following, we present our simplified learning model, *i.e.*, Eq. (5). For a matched moving feature (see Eq. (4)) we adjust the voting vector, *i.e.*, the radius r and the relative angle φ (described above), using

$$r_t^{(i)} = \alpha\ r_{t-1}^{(i)} + (1-\alpha)\ r^{(i)}, \tag{8}$$

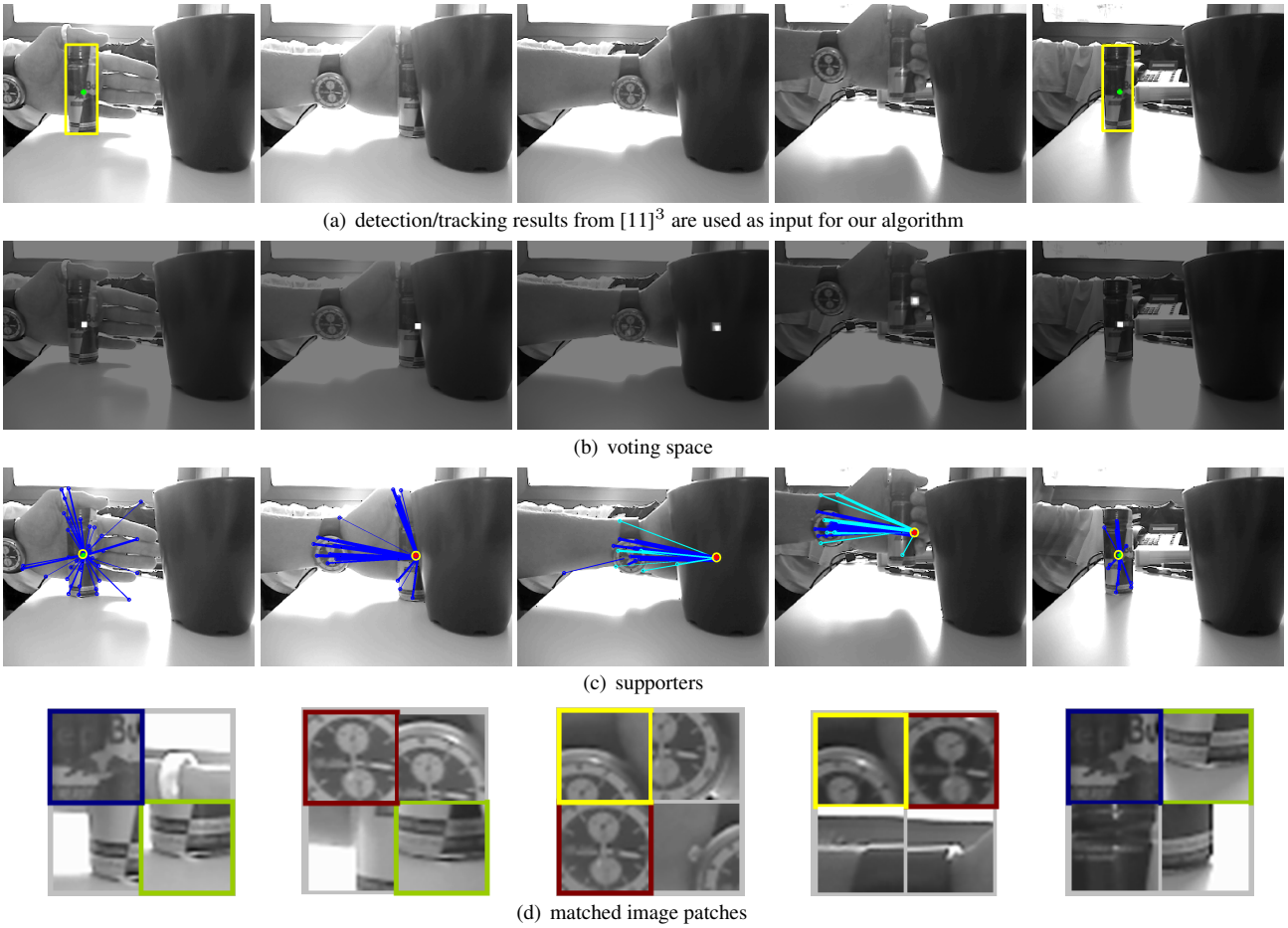
$$\varphi_t^{(i)} = \alpha\ \varphi_{t-1}^{(i)} + (1-\alpha)\ (\varphi^{(i)} - \varphi_0^{(i)}), \tag{9}$$

with the current observations

$$r^{(i)} = ||\mathbf{x}_t^{(i)} - \mathbf{x}_t^{\star}||_2, \quad \text{and} \quad \varphi^{(i)} = \angle(\mathbf{x}_t^{(i)}, \mathbf{x}_t^{\star}). \tag{10}$$

Further, for determining the quality of the feature we update the covariance matrix

$$\mathbf{\Sigma}_t^{(i)} = \alpha\ \mathbf{\Sigma}_{t-1}^{(i)} + (1-\alpha)\ \mathbf{\Sigma}^{(i)}, \tag{11}$$



(a) detection/tracking results from [11][3] are used as input for our algorithm

(b) voting space

(c) supporters

(d) matched image patches

Figure 5. *ETH-Cup* sequence: (a) By using a state-of-the-art approach combining detection and tracking, the object positions can be successfully tracked and also re-detected once it gets lost. (b) Our approach estimates a score (encoded in the brightness), which is related to the probability distribution of the object center position, exploiting both object appearance as well as its motion context. (c) The supporters are learned directly (position provided by the tracker; green points; blue lines) and 2^{nd} level supporters (learned by the estimated position of the 1^{st} level supporters; red points; cyan lines). The predicted position is the significant maximum in the voting space, which is indicated by the yellow circle. (d) Image patches corresponding to matched supporters are shown in more detail. The same color identifies the use of the same supporter feature. Please note that a pure interpolation between the tracked points from (a) would not work here.

with the current estimate

$$\mathbf{\Sigma}^{(i)} = (\mathbf{x}_t^\star - \boldsymbol{\mu}^{(i)})\ (\mathbf{x}_t^\star - \boldsymbol{\mu}^{(i)})^\mathsf{T}. \tag{12}$$

For strong correlated features the variance thus decreases and the voting gets more peaky.

"Reliable information". The reliable information, when available, is the input for our learning. In this work we mainly explore two cues, (i) a human supervisor (no errors) and (ii) a very conservative object detector/tracker (*i.e.*, very high precision but potentially low recall). Although our approach is able to cope with some (uncorrelated) errors, experiments showed that having few but accurate information sources leads to better results.

Second level supporter. As described above, we refrain from updating the model for supporters when the object is directly and reliably observed – the "first level supporters" discussed so far. In order to still benefit from surrounding image features even when no reliable information is present, we introduce so called "second level supporters". If the voting is very confident, *i.e.* the maximum of the voting space is above a user defined threshold, we use it for training additional supporters. Since self-prediction is used, we encode the uncertainty by multiplying the covariance matrices by a factor of two. Furthermore, in order to avoid drifting as a result from the self-learning, we avoid a feedback loop and use these second level supporters only to calculate $P(\mathbf{x}|I)$, *i.e.* for the determination of the target location, together with the first level supporters. They are, however, not used to assess the confidence of the voting and thus do not help initiate the creation of further supporters. This remains the sole privilege of the first level supporters.

4. Experimental Results

In the following we demonstrate the benefits of the presented approach. First, we give a detailed analysis focusing on improving model-free tracking. Second, we use only information from the first frame in order to incrementally include supporters into the model. Finally, a practical application from medical imaging is shown.

4.1. Improving Model-Free Tracking

Recently published approaches for model-free tracking combine tracking and detection in a unified framework (*e.g.*, [11]). They are, to some extent, able to cope with appearance changes of the object (*e.g.*, out of plane rotations) and partial occlusions. Nevertheless, at least some parts of the object have to be visible.

We applied the detection/tracking approach of Stalder *et al.* [11], where the object is manually initialized in the first frame. In the *ETH-Cup* sequence (see Fig. 5a) a small bottle is moved by hand behind a cup (which fully occludes it), lifted up and finally gets visible again. The tracker provides high precision, *i.e.* hardly any incorrect positions are reported. However, the object is lost once it gets (partially) occluded. The result (output) of the tracker is passed as input to the proposed approach. The resulting voting space is depicted in Fig. 5b, which this time clearly shows a strong peak even when the object is fully occluded.

Learned Supporters. Fig. 5c shows the learned supporters which predict the estimated position of the object through the voting space. The line points to the mean and the width encodes the confidence (proportional to $|\mathbf{\Sigma}|^{-1}$) and is only plotted for significant features. Second level supporters (cyan) further stabilize the prediction as the number of first level supporters (blue) might decrease over time (second to fourth image).

Additionally, Fig. 5d shows examples of active supporters at the corresponding time instances. As can be seen, the system manages to efficiently exchange such supporters for others that are better suited at any time. At the beginning, supporters lie mostly on the object. Once the object gets occluded, the position is determined by the on-line learned context (watch). As soon as the object re-appears, so do features lying on it in the supporter set.

Comparison. For comparison we generated a gold standard of where the object is in the scene. Ten humans were asked to mark the most probable center position of the object in every frame. We consider the reported position as correct (true positive), if it is within 15 pixels of the average human estimate (average plus or minus 2 times the standard deviation). As can be seen from Fig. 6 and Tab. 1, our approach essentially increases the recall (true positive divided by the length of the sequence), while just loosing slightly in precision (true positives divided by the sum when tracking results are reported). This is due to the fact that we are able to predict the position even when the object is not visible.

[3] `www.vision.ee.ethz.ch/boostingTrackers`, 2009/09/23 (standard parameter)

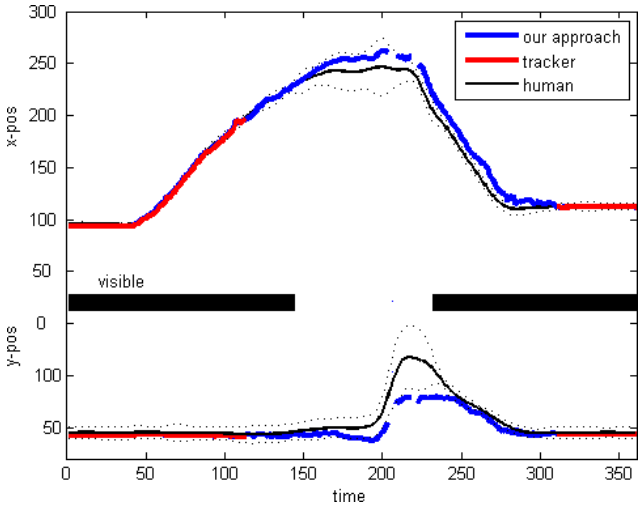


Figure 6. Tracking the object was improved substantially over the original tracker [11] (red), especially when the object is partly or fully occluded. Our algorithm (blue) shows qualitatively the same behavior as human annotators (black; mean $\pm$ standard deviation from 10 humans).

Method	Recall	Precision
tracker/detector [11]	45%	100%
+ proposed approach	89%	97%

Table 1. Improving model-free tracking for *ETH-Cup*.

4.2. One-Shot-Learning

As a special case, let us assume that reliable information is only provided at the first frame. In particular, we manually delineate the target object in that frame and only trust that information, no further detections later on. All interest points on the object are taken as the only supporters. The indicator distribution for each supporter is a Gaussian, with as its mean the relative position to the target object in the first frame, and with a preliminary covariance matrix $\mathbf{\Sigma} = \sigma^2\mathbf{I}_2$ (where $\mathbf{I}_2$ stands for a 2×2 identity matrix and where we used $\sigma^2 = 10$). In order to improve robustness, supporters that also match one of the background features are removed from the model. Since after the first frame no further reliable information is available, only second level supporters are learned.

Fig. 7 shows two example sequences captured with a hand held camera. In the first sequence an image on a piece of paper is tracked. After the static initialization of the first level supporters from the first image, the second level supporters are continuously learned and adapted. They allow

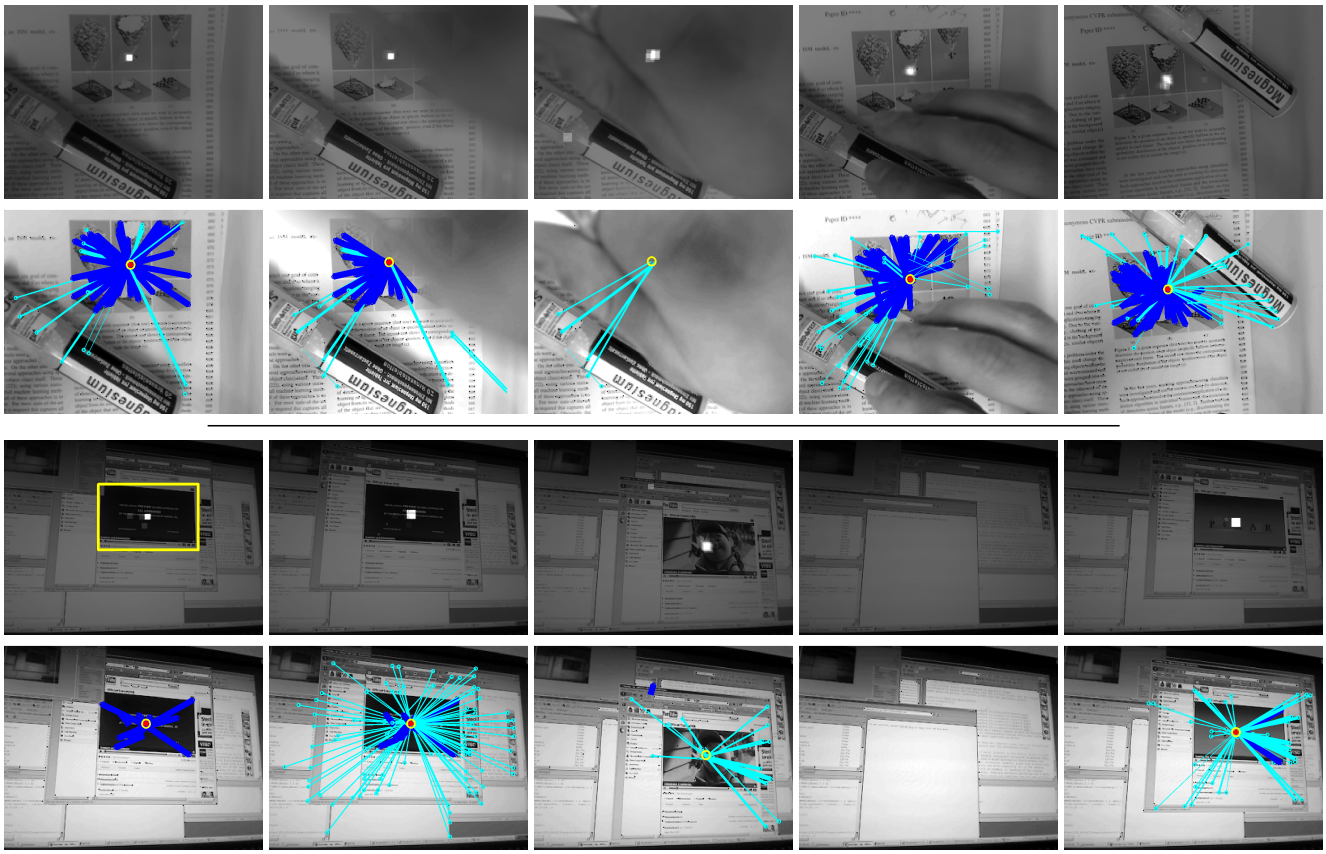


Figure 7. Voting space (first and third row) and corresponding supporters (second and fourth row) for two sequences where the target object is only marked in the first frame only. The target object (a picture on the paper and a video) are tracked successfully.

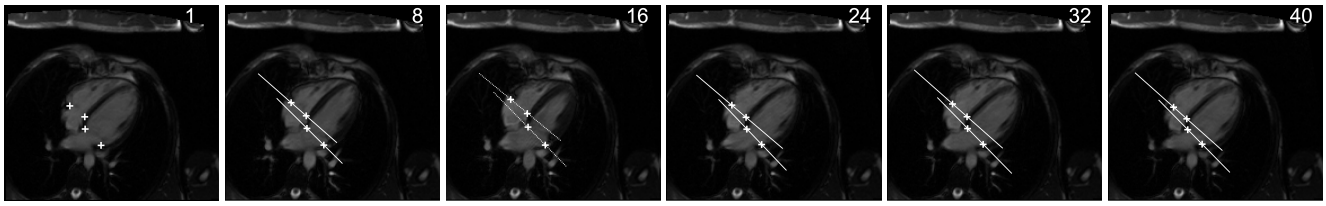
to correctly predict the correct target position even when the picture is completely occluded as in the middle image. They are, furthermore, continuously adapted when the scene changes. In the second sequence a window showing the trailer from the movie "Up" is tracked. Although the object itself changes rapidly it can be robustly tracked thanks to the learned second level supporters. Hence, tracking is almost totally based on motion context here.

4.3. Medical Application

Object tracking is also a topic of major interest in the medical field. In fact, the medical community developed a largely independent ecosystem of methods (the registration methods [6]) that, similarly to optical flow, try to find a dense deformation field between two images often captured by different imaging modalities. Occlusion plays only a minor role in the medical field, but the objects can move completely out of the imaging plane and thus become invisible.

An example is tracking of the cardiac valves in two-dimensional Magnetic Resonance (MR) sequences. For reliable diagnosis, the medical doctors need two-dimensional image sequences with the cardiac valves always in the imaging plane. As the optimal imaging planes for the two valves are initially unknown, an image sequence perpendicular to the valve plane is acquired, see Fig. 8. The cardiac valves are then labeled in this sequence to find the optimal imaging plane for the subsequent scan. This scan then shows the cardiac valves always in the imaging plane.

Common tracking approaches do not work for these images as the valves perpetually change their shape and sometimes are not even visible within the imaging plane. In this scenario a first sequence of 40 images, over an entire cardiac cycle, is captured. The medical doctor defines, in the first frame, four points that mark the two planes through both valves, see Fig. 8a. This accurate human input data is then used to train the model. In contrast to the previously described one-shot-learning, no object boundary but only four target points are labeled and therefore all feature points are considered as supporters. As quasi-static background features such as the MR table or thoracic wall would negatively influence the tracking accuracy we try to suppress them. As they are far away from the target points, their influence is limited using the squared Euclidean distance in the calculation of the covariance matrix for each feature i in the first frame, thus $\mathbf{\Sigma}^{(i)} \propto ||\mathbf{x}^\star - \mathbf{x}^{(i)}||^2 \cdot \mathbf{I}_2$. The model learned from the first image is then applied to the remaining images



(a) (b) (c) (d) (e) (f)

Figure 8. (a) Manually labeled valve positions, (b-f) the found valve positions in 5 cardiac phases. As the confidence of the voting space was too low, in cardiac phase 16, the valve positions were interpolated (indicated by the dotted lines) from neighboring images.

in the sequence, see Fig. 8b-f. If the maximum confidence in the voting space falls below a fixed threshold, the target position for all four points is interpolated from neighboring images. One can also think of requesting the user for further manual input to avoid interpolation.

The found valve positions that define the optimal imaging plane are then feed back to the MR scanner. A new sequence with the cardiac valves always in the imaging plane is acquired. Although the user could have manually labeled all the positions of the valves in every frame, user intervention should be kept at a minimum. The reason being that prolonged labeling bears the danger that the heart slightly drifted away from its previous position during the first scan and thus limiting the usefulness of the second MR acquisition.

5. Conclusion

In this work we explore context in visual tracking. *Supporters* are learned on-line in order to determine the most probable position of the object in the scene. These supporters are determined from statistical dependences, *e.g.*, caused by coupled motions. They might come from target object points, shadows, other objects interacting with the target, or any regions that provide information about the object positions. This coupling can be permanent or temporary and is exploited using the Generalized Hough Transform principle. The proposed approach can easily be combined with tracking approaches and significantly improves tracking performance, especially when the object is occluded or if it changes its appearance heavily. Additionally, we showed results from a medical application that robustly tracks (virtual) points in an MR cardiac image sequence.

Obviously there exist situations where our assumptions are violated. If the coupling with the supporters changes abruptly, then our method will have no time to adapt and relying on such supporters becomes a liability instead of an asset. One might think of a magician as an extreme case, who explicitly exploits that and thereby even misleads the human visual system. This type of situations is beyond the scope of the paper and calls for the inclusion of higher level scene understanding and reasoning.

Acknowledgments. This research was supported by the European Community's Seventh Framework Programme under grant agreement no FP7-ICT-216465 SCOVIS and FP7-ICT-247022 MASH. We further thank Henk-Joost Crooijmans for inspiring discussions.

Supplementary Material. Datasets, annotations and result videos are available on the authors' web-page.

References

[1] S. Baker and I. Matthews. Lucas-kanade 20 years on: A unifying framework. *IJCV*, 56(3):221–255, 2004.

[2] L. Cerman, J. Matas, and V. Hlavac. Sputnik tracker: Looking for a companion improves robustness of the tracker. In *Proc. Scandinavian Conf. on Image Analysis*, 2009.

[3] C. Desai, D. Ramanan, and C. Fowlkes. Discriminative models for multi-class object layout. In *Proc. ICCV*, 2009.

[4] M. Hamouz, J. Kittler, J.-K. Kamarainen, P. Paalanen, H. Kälviäinen, and J. Matas. Feature-based affine-invariant localization of faces. *PAMI*, 27(9):1490–1495, 2005.

[5] C. Harris and M. Stephens. A combined corner and edge detection. In *Proc. Alvey Vision Conference*, pages 147–151, 1988.

[6] D. L. G. Hill, P. G. Batchelor, M. Holden, and D. J. Hawkes. Medical image registration. *Phys. Med. Biol.*, 46:R1–R44, 2001.

[7] H. Kruppa. *Object Detection using Scale-specific Boosted Parts*. PhD thesis, ETH-Zurich, 2004.

[8] B. Leibe, A. Leonardis, and B. Schiele. Robust object detection with interleaved categorization and segmentation. *IJCV*, 2007.

[9] V. Lepetit, P. Lagger, and P. Fua. Randomized trees for real-time keypoint recognition. In *Proc. CVPR*, volume 2, pages 775–781, 2005.

[10] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[11] S. Stalder, H. Grabner, and L. V. Gool. Beyond semi-supervised tracking: Tracking should be as simple as detection, but not simpler than recognition. In *Proc. IEEE WS on On-line Learning for Computer Vision*, 2009.

[12] A. Torralba. Contextual priming for object detection. *IJCV*, 53(2):169–191, 2003.

[13] C. Tzomakas and W. Seelen. Vehicle detection in traffic sceens using shadows. Technical report, Institut fur Neuroinformatik, Ruhr University, June 1998.