IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2008, Anchorage, Alaska. ©IEEE.

Globally Optimal Shape-based Tracking in Real-time

Thomas Schoenemann and Daniel Cremers
Department of Computer Science
University of Bonn, Germany

Abstract

Most algorithms for real-time tracking of deformable shapes provide sub-optimal solutions for a suitable energy minimization task: The search space is typically considered too large to allow for globally optimal solutions.

In this paper we show that – under reasonable constraints on the object motion – one can guarantee global optimality while maintaining real-time requirements. The problem is cast as finding the optimal cycle in a graph spanned by the prior template and the image. The underlying combinatorial algorithm is implemented on state-of-the-art graphics hardware. Solutions on FPGAs are conceivable.

Experimental results demonstrate long-term tracking of cars in real-time, while coping with challenging weather conditions. In particular, we show that the proposed tracking algorithm is highly robust to illumination changes and that it outperforms local tracking methods such as the level set method.

1. Introduction

The trade-off between optimality of computed solutions and required computation time is of fundamental importance to many Computer Vision algorithms, in particular in the context of real-time applications such as the tracking of moving vehicles in traffic scenarios.

To guarantee real-time performance, researchers typically revert to a number of heuristics such as the selection of prominent feature points [9, 11, 15, 6]. In addition, optimization of cost functions is typically done in a local framework [19, 1], providing the best solution in the vicinity of the initialization. Drastic improvements in practical performance can be obtained by introducing sampling and particle filtering in order to avoid certain local minima [2, 7]. Yet, these approaches typically do not guarantee global optimality in reasonable time. While the quality of results increases with the number of particles, the run-time increases, too.

If real-time performance is not required, often variational methods with shape priors are used: They provide a consistent object model and take into account the full intensity information of the input image rather than tracking a sparse set of features independently. After pioneering works of Grenander et al. [10] and Cootes et al. [3], notable advances in the area of image segmentation with prior shape knowledge were made by Leventon et al. [14], Cremers et al. [5], Tsai et al. [20] and Rousson and Paragios [17]. All these methods can be applied to tracking. For comparison we selected the level set method of Rousson and Cremers [16], which models the distribution of shape and color of the object by means of kernel density estimates computed from a set of training shapes.

The above mentioned variational methods rely on continuous optimization and – being *local* optimization methods of typically non-convex functionals – depend on initialization. In contrast, several algorithms using discrete optimization have been proposed that do not depend on initialization. These works include the method of Coughlan et al. [4] to find open contours in images and the approach of Felzenszwalb [8]. While the latter produces globally optimal segmentations in polynomial time, due to its quadratic memory complexity pixel-accurate segmentation will not become practicable for many years to come.

In [18] we proposed an efficient solution to introduce shape knowledge in pixel-accurate globally optimal image segmentation. Yet, for tracking objects the algorithm suffers from three limitations: Firstly, their translation-invariant formulation treats objects independently. Secondly, the runtimes are in the order of 15 seconds per frame, which is far from real-time requirements. Lastly, the algorithm may give rise to incorrect multiple alignments, thereby failing to guarantee the desired global optimum. The solution we offered – a complete search over the initial correspondence – drastically increases run-times even further.

Contribution. We present the first shape-based method that is capable of tracking objects in real-time while providing globally optimal solutions. Real-time performance is achieved by exploiting prior knowledge about object motions as well as specific properties of the underlying combinatorial algorithm. In contrast to [18], the proposed al-

gorithm guarantees global optima without requiring a complete search over the initial correspondence.

On several long car sequences we demonstrate that the proposed algorithm can reliably track driving vehicles despite difficult weather and lighting conditions. In particular, we show that the proposed combinatorial technique outperforms respective local methods.

2. Ratios for Matching Shapes to Images

We start with a review of the approach of Schoenemann and Cremers [18]. By minimizing a ratio energy, they are able to match the contour of a prior shape to a given image.

2.1. Optimal Shapes as Minimum of a Ratio Energy

Given a contour S and an image I, the task is to find a contour C in the image that is located at image edges and similar to S. Both contours are parameterized by arc-length, i.e. $S : [0, l(S)] \to \mathbb{R}^2$ and $C : [0, l(C)] \to \mathbb{R}^2$ with $l(\cdot)$ denoting the length of a curve. The optimal solution is found by globally minimizing a ratio energy which normalizes the integral of a data term and a shape consistency measure by the length of the desired contour.

For the data term, the positive edge detector function $g(\mathbf{x}) = 1/(1 + |\nabla I(\mathbf{x})|)$ is used. This function assigns low values to high image gradients. The shape consistency measure is a sum of two terms: First, the deviation of tangent angles is penalized. Second, a part of the prior curve S may be matched to a part of C of different length, but the induced length distortion is penalized.

Both points are based on the computation of a (monotone) matching function $m : [0, l(C)] \to [0, l(S)]$, expressing which point on C corresponds to which point on S. This matching function is optimized together with the image contour. The penalization of length distortion corresponds to penalizing the derivative m' of the matching m, where $m' = 1$ is favored. To compare tangent angles, the *cyclic* distance between the tangent angle $\alpha_C(s)$ at s and the corresponding tangent angle $\alpha_S(m(s))$ is computed.

The resulting optimization problem is reduced to finding cycles in a regular graph. Applying a combinatorial algorithm to the graph allows to find the optimal pair of image contour C and matching function m simultaneously.

2.2. The Optimal Contour as Cycle in a Graph

The mentioned graph has a torus-like, three-dimensional structure as visualized in Figure 1. Each pair of image contour and matching function is identified with a cycle in the graph. Its node set is spanned by the pixel set $\mathcal{P}$ of the image and K instances of each point on the prior contour, representing K possible image locations of one shape point:

$$\mathcal{V} = \mathcal{P} \times \{0, \dots, K \cdot |S|\} \tag{1}$$

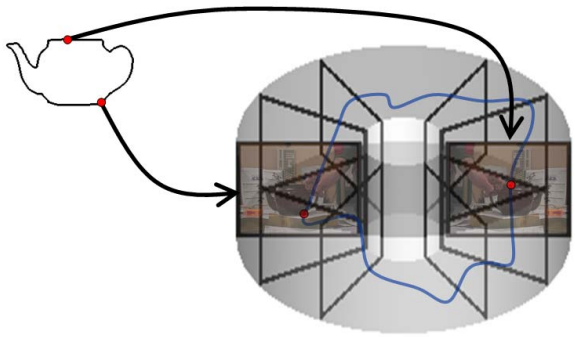


Figure 1. The structure of the graph: For any point on the prior contour there are K copies of the image in the graph. If a cycle in the graph passes through such a frame, this defines an assignment of a pixel in the image to a point on the contour of the prior shape.

For any point s on the prior contour, there are K collections of nodes, called *frames* in the following. A frame contains one node for each pixel in the image.

A cycle passing through the node $(\mathbf{x}, t)$ defines an image contour C passing through pixel $\mathbf{x}$. Moreover, this pixel is matched to the point $s = \lfloor t/K \rfloor$ on the prior contour. By following the cycle the complete object silhouette and matching function are recovered. To reflect the energy functional, each edge is assigned two weights for the respective parts of the numerator and the denominator integral.

3. Shape-based Tracking via Ratios

Given a video sequence and (the position of) an object in the first frame, we are interested in tracking the object over the entire sequence. For many applications this task needs to be solved in real-time. The method presented here fulfills this requirement.

Determining the object location at time t means to find its silhouette C_t in the corresponding image. The optimal silhouette is given as the minimum of a ratio energy which normalizes an integral of a data term, a shape consistency measure and a motion function by the length of C_t. The data term and shape consistency measure correspond to those in section 2.1, where for the length distortion the penalty function

$$\Psi(m') = \begin{cases} m' - 1 & \text{if } K \geq m' \geq 1 \\ \frac{1}{m'} - 1 & \text{if } \frac{1}{K} \leq m' < 1 \\ \infty & \text{otherwise} \end{cases} \tag{2}$$

is used. The constant K limiting the maximal length distortion is set by the user. For the deviation of tangent angles, the squared *cyclic* distance enters. For simplicity this is denoted $|\alpha_C - \alpha_S|^2$.

The motion function evaluates the motion of each point of the silhouette, i.e. the displacement between $C_t(s)$ and its corresponding point $C_{t-1}(m(s))$ in the previous image. It is denoted

$$\Theta : \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}$$

Our framework makes no restrictions on this function. We experimented with several functions and got best results when disallowing large motions, but not favoring any particular small motion:

$$\Theta(\mathbf{x}_t, \mathbf{x}_{t-1}) = \begin{cases} 0 & \text{if } ||\mathbf{x}_t - \mathbf{x}_{t-1}|| \leq D_{\max} \\ \infty & \text{otherwise} \end{cases} \quad (3)$$

with maximal distance $D_{\max}$ and where we use the Max-Norm for $||\cdot||$. The optimal contour C_t in image t is then determined by solving

$$\begin{aligned} \min_{C_t, m} \quad & \frac{\int_0^{l(C_t)} g(C_t(s))\, ds}{l(C_t)} + \lambda \frac{\int_0^{l(C_t)} \Psi(m'(s))\, ds}{l(C_t)} \\ & + \nu \frac{\int_0^{l(C_t)} |\alpha_{C_t}(s) - \alpha_{C_{t-1}}(m(s))|^2\, ds}{l(C_t)} \\ & + \frac{\int_0^{l(C_t)} \Theta(C_t(s), C_{t-1}(m(s)))\, ds}{l(C_t)} \end{aligned} \quad (4)$$

While this is a complex optimization problem it can be minimized in polynomial time. Analogously to the construction in Section 2.2 each possible solution is mapped to a cycle in a graph. An edge e in the graph represents a line segment of the image contour. It is assigned a numerator weight $n(e)$ and a denominator weight $d(e)$ representing the respective integral of the functional *along the line segment*. The tracking energy (4) of a possible solution is reflected by the ratio $\sum_{e\in\Gamma} n(e) / \sum_{e\in\Gamma} d(e)$. The optimal cycle in the graph can be found via the Minimum Ratio Cycle algorithm [13] introduced to Computer Vision in [12]. However, there are two problems with this approach:

- Some cycles in the graph do not correspond to a valid combination of alignment and image contour. One hence needs to optimize over the subset of *valid* cycles. How to do this without having a quadratic dependence of the run-time has so far been an open issue.
- The arising more complicated problem should preferably be solved in real-time.

4. Real-time Optimization

In this section we deal with the above mentioned drawbacks. We present a number of improvements which will result in real-time performance while guaranteeing that the global optimum in the space of *valid* solutions is found. To achieve this we apply the following steps:

- By assuming a maximal velocity of $D_{\max} = 15$ per frame – see eqn. (3) – the search space is significantly reduced.
- We rely on a parallel implementation of the Minimum Ratio Cycle algorithm, implemented on a graphics card (Sec. 4.2). Additionally the edge detector function $g(\cdot)$ is pre-computed in parallel.

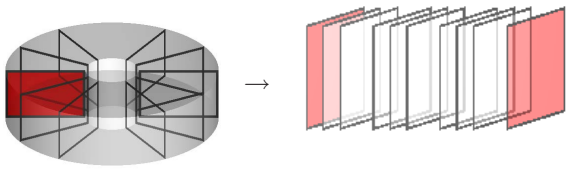


Figure 2. The graph is cut open at the red frame, the red frame is doubled. The arising graph is acyclic. This is the key for efficient optimization.

- The underlying combinatorial algorithm profits from smart initialization. By providing such an initialization (Sec. 4.1), we are able to reduce the run-time drastically: a sequence that needed 14 seconds now runs in 4 seconds.
- By applying a recursive splitting algorithm we guarantee that the globally optimal *valid* cycle is found. In practice this takes very little extra run-time (Sec. 4.3).

4.1. Ratio Optimization on Graphs

The optimum among *all* cycles is found via the Minimum Ratio Cycle algorithm [13]. It is based on iterated negative cycle detection w.r.t. *single* edge weights $w(e) = n(e) - \gamma\, d(e)$: Negative cycles in the graph correspond to cycles with ratio smaller than γ and vice versa.

The minimization algorithm starts with an upper bound γ on the optimal ratio. It then proceeds by performing negative cycle detection. If a negative cycle is found, γ is set to its ratio and the process repeated. Otherwise the last found cycle is returned as it must be an optimal one.

The better the initial bound, the less iterations are needed. We provide a tight upper bound by shifting the previous contour by up to 5 pixels in each direction, taking the minimum of all resulting energies.

Negative cycles are found by integrating cycle detection into the Moore-Bellman-Ford algorithm for distance calculations: From a given root node (in our case an extra node), the optimal path to each node in the graph is calculated. If the graph contains negative cycles, a cycle will form in the predecessor graph. Regularly checking the predecessor graph for cycles allows to find negative cycles.

4.2. Parallel Implementation

The structure of the torus-like graph allows highly parallel implementations of the optimization algorithm: By cutting the graph at frame 0 and duplicating frame 0, a directed acyclic graph is obtained. This is illustrated in Figure 2. In the resulting graph, the distance labels for frame t only depend on the labels of frames $t' < t$. They are determined by proceeding along t, computing the labels for frame t in parallel. GPU-based implementations currently allow speed-up factors of 300 and more. For details on the optimization procedure see [18].

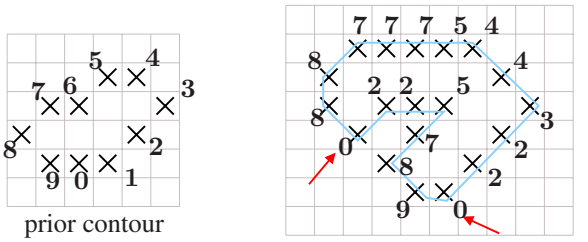
Proposed method, shown for frames #1, 30, 60 and 90: Despite the bad weather, the car is reliably tracked.

competitive algorithm [18]: the trees are favored.

without exclusion of multiple alignments

proposed method (frame # 50)

Figure 4. Tracking of a car in rainy weather: Thanks to spatial coherence and exclusion of multiple alignments, this works reliably.



prior contour

unwanted multiple alignment

Figure 3. A prior contour and an unwanted multiple alignment (assuming $K \geq 3$): As indicated by the red arrows, there are two (unconnected) pixels that correspond to the prior point 0.

4.3. Excluding Incorrect Solutions

The above described algorithm finds the optimum of *all* cycles in the graph. However, some cycles do not correspond to a valid combination of matching and image contour. Such cycles contain several nodes in frame 0, i.e. nodes of form $(\mathbf{x}, 0)$. Intuitively speaking they wrap around multiple times in the interior of the graph. We call this *multiple alignments*. An example is given in Figure 3. In our experience the optimal cycle corresponds to a multiple alignment in about 2% of all cases.

However, optimizing (4) means to find the optimal cycle corresponding to a *single* alignment. We provide an efficient solution by applying a recursive algorithm: If the globally optimal cycle is a multiply aligned one, all nodes in the cycle that belong to frame 0 are collected. Frame 0 is then partitioned into as many components as nodes were found, such that no two of them are in the same component. For each component, the algorithm is called recursively, where the initial ratio is set to the ratio of the last found singly aligned cycle (or the initial upper bound if none was found). In the recursive calls, all nodes of frame 0 that are not in the indicated component are removed (in practice, their distance labels are fixed at ∞).

While none of the recursive calls will return the previously found multiple alignment, some may again find multiply aligned cycles. In this case the partitioning process is continued and more recursive calls are made. In the end the globally optimal single alignment is found. While the worst case complexity of this algorithm is quadratic in the number of pixels, in practice we observe linear run-times.

5. Experiments

When tracking objects in real-world traffic scenarios one has to cope with several challenges: The camera varies the shutter time and one faces difficult weather conditions (rain, shadows and sunlight falling directly into the camera). In addition, both the camera and the objects to be tracked are in motion. Yet, tracking needs to be done in real-time.

On several real-world traffic sequences, all recorded under difficult weather conditions, we demonstrate that the proposed method can handle these challenges. The data were recorded at 25 fps, leaving 40 msec per frame for real-time processing. Since the silhouettes of (rigidly) moving cars usually deform only mildly, we set K to 2. Unless otherwise stated the regularity weights $\lambda = \nu = 0.5$ were used together with a maximal distance $D_{\max}$ of 15. Our GPU-implementation is based on the CUDA framework and was run on a Geforce 8800 GTX.

5.1. Real-time Tracking in Traffi Scenarios

With the optimizations introduced in this paper, shape-based tracking becomes real-time capable: Despite the rainy weather, the sequence in Figure 4 is processed with 25 fps.

Figure 5 shows another challenging sequence[1] containing shadow effects as well as sunlight falling directly into the camera. It is processed with 17 fps. When reducing the resolution by 25% in each dimension, real-time perfor-

[1] http://www.citr.auckland.ac.nz/6D/datasets.htm

Figure 5. Real-time tracking of a car over 250 frames. Despite sunlight falling directly into the camera and shadow effects, the car is not lost.

mance is achieved – with slightly less accurate segmentations.

5.2. Comparison to the State-of-the-art

In Figure 6 we compare the proposed method with two Level Set approaches, differing by the data term only. The first is the model of Cremers and Rousson [16]: Despite the sophisticated, histogram-based data term the object is lost almost immediately. The second method uses intensity comparison as data term. This is a bit more stable, but after less than 20 frames the object is lost, too.

As demonstrated in Figure 4, the approach of Schoenemann and Cremers [18] is not suitable for this kind of data, either: As it does not prefer a shape to be (spatially) close to the previous shape, the optimal shape is placed in the trees. Moreover, when including the Θ-function, for $\lambda = \nu = 0.25$ multiple alignments arise. The proposed method produces a valid solution with only a few milli-seconds of extra time.

5.3. Tracking a Passing Car

Lastly we demonstrate in Figure 7 that our method is able to handle significant scale changes and silhouette deformations: In the beginning, the car is viewed partially from the side and close to the camera. At the end of the sequence it is far away and viewed from behind. These results also demonstrate robustness to lighting changes: Due to the shadows cast by the trees, the luminance of the car changes several times from light to dark.

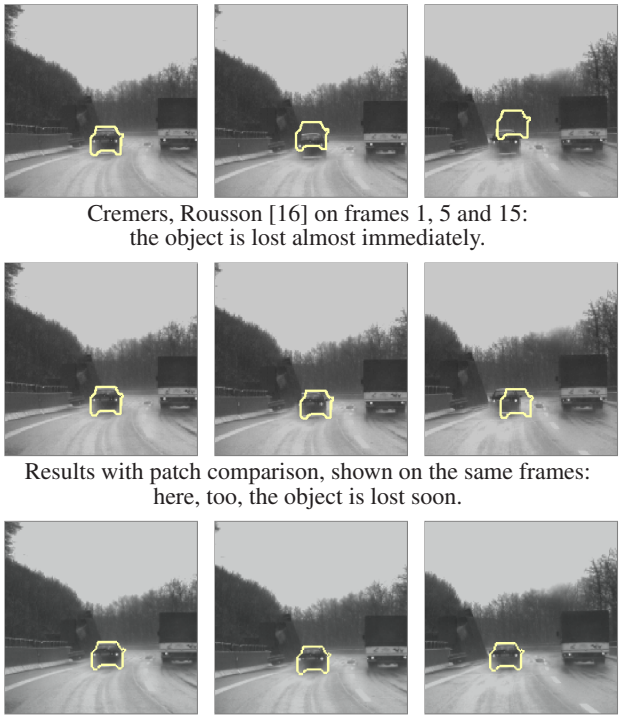


Figure 6. While both simple and sophisticated methods fail after a few frames, the proposed method tracks the object over the entire one hundred frames.

Conclusion

We propose the first shape-based real-time tracking algorithm which guarantees globally optimal solutions. Finding a shape template in an image is reduced to finding cycles in a product graph spanned by image and template. The underlying combinatorial algorithm is parallelizable and profits from smart initialization.

In real-world sequences under harsh weather and illumination conditions, we demonstrate reliable tracking results over hundreds of frames.

Acknowledgements This research was supported by the German Research Foundation (DFG), grants #CR-250/1-1 and #CR-250/2-1. We thank Thomas Pock for helpful comments on efficient GPU-programming and Daimler Research for sharing their data with us.

References

[1] S. Avidan. Support vector tracking. In *IEEE Int. Conf. on Comp. Vision and Patt. Recog.*, Hawaii, Dec. 2001.

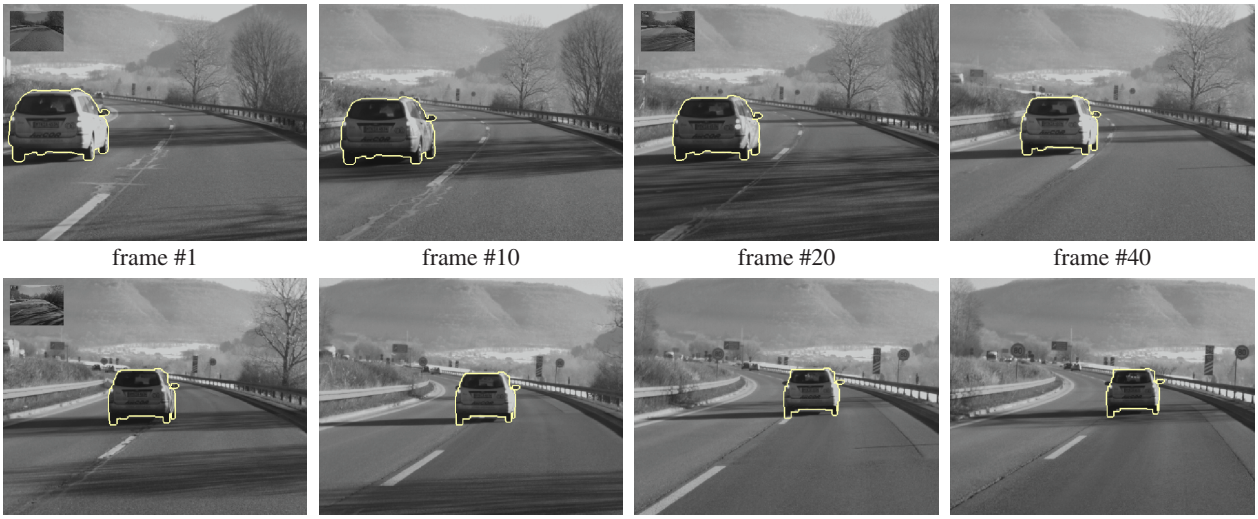


Figure 7. Stable tracking of deforming silhouettes. The changing lighting conditions do not distract the approach.

[2] A. Blake and M. Isard. *Active Contours*. Springer, London, 1998.

[3] T. F. Cootes, C. J. Taylor, D. M. Cooper, and J. Graham. Active shape models – their training and application. *Comp. Vis. Image Underst.*, 61(1):38–59, 1995.

[4] J. Coughlan, A. Yuille, C. English, and D. Snow. Efficient deformable template detection and localization without user initialization. *Comp. Vis. Image Underst.*, 78(3):303–319, 2000.

[5] D. Cremers, F. Tischhäuser, J. Weickert, and C. Schnörr. Diffusion snakes: Introducing statistical shape knowledge into the Mumford–Shah functional. *Int. J. of Comp. Vision*, 50(3):295–313, 2002.

[6] J. Denzler and H. Niemann. Active rays: Polar-transformed active contours for real-time contour tracking. *Real-Time Imaging*, 5:203 – 213, 1999.

[7] A. Doucet, N. de Freitas, and N. Gordon. *Sequential Monte Carlo Methods in Practice (Statistics for Engineering and Information Science)*. Springer, New York, 2001.

[8] P. F. Felzenszwalb. *Representation and Detection of Shapes in Images*. PhD thesis, Massachusetts Institute of Technology, Sept. 2003.

[9] W. Förstner and E. Gülch. A fast operator for detection and precise localization of distinct points, corners and circular features. In *Proc. Intercommission Conf. on Fast Processing of Photogrammetric Data*, pp. 281–305, Interlaken, Switzerland, 1987.

[10] U. Grenander, Y. Chow, and D. M. Keenan. *Hands: A Pattern Theoretic Study of Biological Shapes*. Springer, New York, 1991.

[11] C. Harris and M. Stephens. A combined corner and edge detector. In *Proc. of The Fourth Alvey Vision Conference*, pp. 147–151, Manchester, 1988.

[12] I. H. Jermyn and H. Ishikawa. Globally optimal regions and boundaries as minimum ratio weight cycles. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 23(10):1075–1088, 2001.

[13] E. L. Lawler. Optimal cycles in doubly weighted linear graphs. In *Theory of Graphs: International Symposium*, pp. 209–213, New York, USA, 1966. Gordon and Break.

[14] M. Leventon, W. Grimson, and O. Faugeras. Statistical shape influence in geodesic active contours. In *IEEE Int. Conf. on Comp. Vision and Patt. Recog.*, volume 1, pp. 316–323, Hilton Head Island, SC, 2000.

[15] D. Lowe. Object recognition from local scale-invariant features. In *IEEE Int. Conf. on Comp. Vision*, Corfu, Greece, Sept. 1999.

[16] M. Rousson and D. Cremers. Efficient kernel density estimation of shape and intensity priors for level set segmentation. In *Medical Image Computing and Computer Assisted Intervention*, volume 1, pp. 757–764, 2005.

[17] M. Rousson and N. Paragios. Shape priors for level set representations. In A. Heyden et al., editors, *Europ. Conf. on Comp. Vision*, pp. 78–92. Springer, 2002.

[18] T. Schoenemann and D. Cremers. Globally optimal image segmentation with an elastic shape prior. In *IEEE Int. Conf. on Comp. Vision*, Rio de Janeiro, Brazil, October 2007.

[19] J. Shi and C. Tomasi. Good features to track. In *IEEE Int. Conf. on Comp. Vision and Patt. Recog.*, Seattle, June 1994.

[20] A. Tsai, A. Yezzi, W. Wells, C. Tempany, D. Tucker, A. Fan, E. Grimson, and A. Willsky. Model–based curve evolution technique for image segmentation. In *IEEE Int. Conf. on Comp. Vision and Patt. Recog.*, pp. 463–468, Kauai, Hawaii, 2001.