Discrete Tracking of Parametrized Curves

Tim Hauke Heibel[a]*, Ben Glocker[a,b], Martin Groher[a],
Nikos Paragios[b,c], Nikos Komodakis[d] and Nassir Navab[a]

[a]Computer Aided Medical Procedures (CAMP), Technische Universität München, Germany
[b]Laboratoire MAS, Ecole Centrale Paris, Chatenay-Malabry, France
[c]Equipe GALEN, INRIA Saclay - Ile-de-France, Orsay, France
[d]Computer Science Department, University of Crete, Greece

heibel@cs.tum.edu

Abstract

A novel scheme for deformable tracking of curvilinear structures in image sequences is presented. The approach is based on B-spline snakes defined by a set of control points whose optimal configuration is determined through efficient discrete optimization. Each control point is associated with a discrete random variable in a MAP-MRF formulation where a set of labels captures the deformation space. In such a context, generic terms are encoded within this MRF in the form of pairwise potentials. The use of pairwise potentials along with the B-spline representation offers nearly perfect approximation of the continuous domain. Efficient linear programming is considered to recover the approximate optimal solution. The method is successfully applied to the tracking of guide-wires in fluoroscopic X-ray sequences of several hundred frames which requires extremely robust techniques.

1. Introduction

Applications requiring spatio-temporal information about moving objects are various. A possible approach to acquire this information is by means of tracking a parametric curve representation of the object in time. Tracking of closed curves representing *contours* of objects has received a considerable amount of attention in the computer vision community [25]. In order to track curvilinear structures, however, a curve representation of their centerlines would be more appropriate than one of their contours, which inevitably involves the model of an open curve. Since the adaption of existing algorithms to the tracking of open curves is not straight-forward, only little work can be found, which tackles this problem.

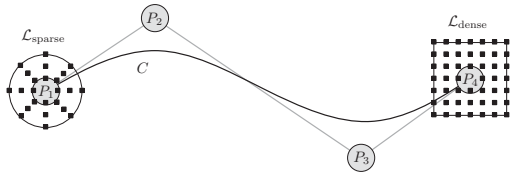


Figure 1: MRF model for an open cubic B-spline curve C with control points P_i. A sparse and a dense version of the discrete set of labels $\mathcal{L}$ capturing the deformation space are illustrated (each black square corresponds to a displacement plus the zero-displacement at the control point position).

We propose a unified framework for the tracking of open and closed parametrized curves, which model object characteristics like centerlines or boundaries. Based on a MAP-MRF formulation, we derive a discrete scenario in which the tracking solution can be regarded as the choice of an optimal labeling only comprising the control points of the curve, see Fig. 1. Our method performs in real-time, is shown to be both robust and accurate, and is generic in the choice of data and regularization terms. Moreover, the discrete framework can track multiple curves at the same time without altering or extending the model. In particular, we summarize the contributions as:

Bridging the gap between continuous formulation and discrete optimization: We propose a novel MAP-MRF model for parametrized curves based on B-spline snakes defined by a set of control points whose optimal configuration is determined through efficient discrete optimization. Each control point is associated with a discrete random variable where a set of labels captures the deformation space. The advantage of the MRF formulation is the ability of defining discrete local search spaces which can capture larger

*This research was funded by an academic grant from Siemens Medical Solutions Angiography/X-Ray division, Forchheim, Germany. The authors would like to thank in particular K. Klingenbeck-Regn and M. Pfister for their continuous support.

deformation, are less sensitive to initialization, and are not limited to the gradient direction of the cost function. Furthermore, our model enables us to use powerful recently proposed discrete optimization methods [15, 18].

Universality and accurate approximation of energies: In our model, generic energy terms are encoded within the MRF in the form of pairwise potential functions. We show that the use of pairwise potentials along with the B-spline representation offers nearly perfect approximation of continuous energy terms commonly used for curve tracking and evolution. Besides generic likelihood terms driving the motion of the curve, we consider local regularization terms, such as length preservation and/or diffusion. Additionally, our framework allows the integration of higher-order terms such as curvature without the need for introducing higher-order cliques. Within such a discrete framework, no differentiation of the energy terms is needed which allows for further extensions to other (application specific) terms without significant changes. For instance, we can easily employ locally learned shape priors.

Evaluation of discretization effects: We perform experiments on the discretization effects when modeling continuous energies in a discrete setting. We present promising tracking results with respect to different strategies for search space discretization, energy approximation, and MRF optimization. Two state-of-the-art methods, namely the TRW-S [15] and the FastPD [18] algorithm, are compared for the specific application of open curve tracking.

The remainder of the paper is organized as follows. First, we give an overview about the related work on curve tracking. In Section 3, we present the continuous formulation, followed by our discrete MRF model. Experiments on synthetic data and an evaluation of the mentioned discretization effects are presented in Section 4. Section 5 describes a specific application from the medical imaging domain, while the last Section concludes our paper.

2. Related Work

A pioneering solution to the problem of detecting and tracking boundaries was proposed by Kass et al. [13] with their work on snakes. The idea behind snakes or active contours is matching a deformable model to image data by minimizing an energy function, which is composed of two components. One component which attracts the curve to object boundaries or its center in case of line-like structures and a second component used to restrict the motion the curves undergo. The first component models external energies driving the curve motion while the second component resembles internal forces of the curve that impose regularity and smoothness constraints on the problem.

In their work on the dynamic analysis of apparent contours, Cipolla and Blake [7] use B-splines instead of sets of pixels for modeling contours. They claim that the new representation allows them to completely drop the snake's internal energy, implying that their approach relies on an initialization being close to the global optimum. The parameters of the cost function are the components of the B-spline's control points and they are updated iteratively in steepest gradient direction. For a sufficient capture range a scale-space approach is required, which is also used by Kass et al. [13]. A difficulty that can arise in this context is oversmoothing. If the scale space parameter is chosen too big in order to be able to lock onto a given contour, neighboring edges and lines may merge and thus lead to a wrong optimum in the cost function.

An approach for curve tracking based on dynamic programming, which is closely related to our MRF formulation, has been subject of major interest in the 1990s [2, 10, 14]. Amini et al. [1] propose a method based on B-splines, in which the energy is separated into a sum of single energy terms, each term modeling the energy for a single B-spline span. Considering small search windows centered at the current locations of the control points, dynamic programming is used iteratively to compute the curve updates. On the one hand, this algorithm does not approximate the energy, but employs the local support of B-splines to compute an optimal solution for the continuous domain. On the other hand, if each control point is given a reasonable search space, the method cannot fulfill the hard real-time constraints due to the costly evaluation of all possible control point configurations.

For the application of snake-based segmentation, Caselles et al. [6] introduced a novel scheme for the detection of object boundaries. They reformulated the original snake cost function for B-splines into a geodesic formulation where a level-set function u is sought after. However, utilizing this approach for tracking, in particular open contours, is not straight forward if possible at all. The problem arises from the fact that level-set methods require the zero-level set to separate the image in at least two distinct regions, which for open curves would only be the case if the curve intersects the image border.

Another approach for tracking of curvilinear structures is proposed by Isard and Blake with the CONDENSATION algorithm [12]. Contours are again represented as B-splines but are restricted in their appearance to a shape-space [3, 4]. The authors formulate a propagation rule of shapes as an equivalent to Bayes' rule for inferring a posterior state density from data for time-varying cases given a learned prior.

Besides the presented approaches many more exist [25], however the discrete MAP-MRF formulation is a valuable learning-free alternative to existing tracking methods. To our knowledge, this is the first time that MRFs combined with B-splines are applied to the problem of curve tracking. In the following, we describe in detail the derivation of our framework.

3. Method

Most tracking algorithms consist of two phases – the initialization phase where the object to be tracked has to be identified, and the tracking phase where previous positions of this object are known. In this work we will focus on the second phase and within that phase especially on the curve model and the associated optimization strategy. Feature images driving the optimization process are an important component of tracking algorithms. Typically such images are acquired by enhancing edges, lines, or corners through image filtering techniques. We are not going into details on their choice, since we do not make any strict assumptions on their computation. In Section 5, an example for the a feature image is presented for the specific case of guidewire tracking in X-ray images. In the following, we will focus on the tracking phase by introducing the curve model and our tracking algorithm.

3.1. Curve Model

B-spline curves represent a convenient way in modeling curvilinear structures and object boundaries. The main advantages are a low-dimensional representation of a continuous curve, the implicit smoothness, and the local support of individual control points. A B-spline curve is defined as the linear combination of control points. Without loss of generality, we consider the particular definition of an open curve[1]

$$C(s) = \sum_{i=1}^{M} N_i(s) P_i \quad \text{where } s \in [0;1] \tag{1}$$

where N_i denote the basis functions and P_i the positions of M control points (see also Fig. 1). In order to track a curvilinear structure or object boundary, one seeks the optimal configuration of the control points such that the modeled curve fits the object being visible in an image.

3.2. Curve Tracking in the Continuous Domain

In the following we review the general continuous formulation of the curve tracking problem. Given an initial curve C, we want to estimate the curve model parameters which provide the best fit of the curve to the corresponding structures visible in an image. A common approach of formulating such a problem is through a *maximum a posteriori* (MAP) estimate. Given an observation $\mathcal{I}$ (in our case a feature image), the MAP estimate is defined as

$$C^* = \arg\max_{C \in \mathbb{F}} P(\mathcal{I} \,|\, C) P(C) \tag{2}$$

where C^* is the optimal curve, $P(\mathcal{I} \,|\, C)$ is the *likelihood* of the estimate and $P(C)$ encodes the *prior* information on the set of feasible solutions $\mathbb{F}$. Assuming Gibbs' distribution for the prior and Gaussian for the likelihood, we can reformulate the MAP estimate as an energy minimization problem

$$C^* = \arg\min_{C \in \mathbb{F}} E(\mathcal{I} \,|\, C) + E(C) \tag{3}$$

where the likelihood energy $E(\mathcal{I} \,|\, C)$ acts as a cost function measuring the quality of a certain model configuration, and the prior energy $E(C)$ acts as a regularization (or smoothness) term on the parameter space. In our scenario the likelihood term (4) is also referred to as the *external energy*, driving the curve to its actual position,

$$E_{\text{ext}}(\mathcal{I} \,|\, C) = \int_0^1 \psi(\mathcal{I}\,(C(s)))\, ds \tag{4}$$

where $\psi : \mathbb{R} \mapsto [0, \infty)$ is a strictly decreasing function. The prior is called the *internal energy*, used to constrain the motion of the curve. Changes in the length with respect to the initialization can be penalized through the first derivative by

$$E_{\text{int}}^{\text{length}}(C) = \int_0^1 \left(1 - \frac{\|C'(s)\|}{\|C'_{\text{init}}(s)\|}\right)^2 ds. \tag{5}$$

In our application, we penalize length changes with respect to the curve C_{init} detected during initialization. Length preservation is an important constraint in case of open curves because standard penalty terms on the first or higher-order derivatives are favoring a shrinkage of the curve. In case of closed curves, usually regularization terms such as diffusion and curvature are considered

$$E_{\text{int}}^{\text{diff}}(C) = \int_0^1 \|C'(s)\|^2\, ds \tag{6}$$

$$E_{\text{int}}^{\text{curv}}(C) = \int_0^1 \|C''(s)\|^2\, ds \tag{7}$$

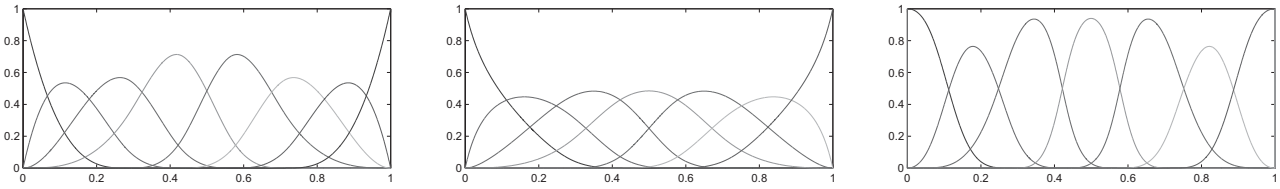
Often, different internal energies are combined by setting weighting factors to the single terms. In case of B-splines, the inherent smoothness is often sufficient and higher-order terms such as curvature can be discarded [7]. The total energy of the curve tracking problem is the sum of the external and internal energies

$$E_{\text{total}} = E_{\text{ext}} + \lambda\, E_{\text{int}} \tag{8}$$

where λ acts as weighting controlling the influence of the regularization term.

In continuous optimization, minimizing the above energies is commonly done in a gradient descent approach. The initial contour is updated iteratively by computing the derivative of the energy function with respect to the model parameters. The algorithm stops if no further improvement

[1] Note that closed curves can be constructed by merging certain tuples of control points.



(a) Unary – $N_i(s)$ (b) Pairwise Sum – $\hat{N}^{+}_{ij}(s)$ (c) Pairwise Product – $\hat{N}^{*}_{ij}(s)$

Figure 2: Influence functions originating from an open cubic B-spline with eight control points.

on the energy can be achieved, meaning the method converges to a local minimum. Even if sometimes convenient to use, such an approach has two major drawbacks. First of all, the algorithm requires the derivation of the energy term, which is oftentimes complex to calculate analytically and has to be done specifically for every function. Second, a convergence to a good solution relies on the fact that the initial contour is sufficiently close. If the gradient descent starts far away from the structure to be tracked, chances are high for obtaining bad solutions. This may easily happen in sequences with larger motions. Multi-resolution approaches (e.g. Gaussian scale space) might help in certain scenarios but do not overcome this general limitation (see also Section 2).

3.3. Discrete Curve Tracking with MRFs

In order to overcome the limitations of continuous optimization, we render our tracking problem in a discrete formulation. Let us consider a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ consisting of a set of discrete variables or nodes $\mathcal{V}$ and a set of edges $\mathcal{E}$ connecting neighboring pairs of variables in order to represent their dependencies. Additionally, we introduce a discrete set of labels $\mathcal{L}$ capturing the search space of the model parameters. Each label $x_i \in \mathcal{L}$ is associated with a two-dimensional displacement vector $\mathbf{d}^{x_i}$ from the deformation space $\mathcal{D} \subset \mathbb{R}^2$. Two possibilities for the discretization of the deformation space are illustrated in Fig. 1. If we associate each control point of our B-spline model with one node of our graph, the task is to assign an optimal label to each of the nodes or a displacement to each of the control points, respectively. Note that our graph is a chain with $|\mathcal{V}| = M$ which is either open ($|\mathcal{E}| = M - 1$) or closed ($|\mathcal{E}| = M$). A common approach for modeling the labeling task in terms of energy minimization is the usage of first-order MRFs [19]

$$E_{\text{mrf}} = \sum_{i \in \mathcal{V}} \theta^i(x_i) + \sum_{(i,j) \in \mathcal{E}} \theta^{ij}(x_i, x_j) \tag{9}$$

where θ^i are the *unary potentials* and θ^{ij} are the *pairwise potentials*.

In most applications, the unary terms play the role of the likelihood energy. Independently from all other nodes, the cost for an assignment of a certain label x_i is evaluated. Then, the pairwise interaction terms play the role of regularization between neighboring nodes. However, the assumption that the likelihood of a labeling can be computed from a sum of independent unary terms is actually not valid in our scenario. Considering B-splines with higher-order basis functions, the effect of a single control point onto the deformation of the curve cannot be modeled independently from its neighbors because the basis functions overlap (see also Fig. 2(a)). Therefore, we propose a novel MRF model for the case of curve tracking using B-splines. First, we will use the basis functions as weighting coefficients within the energy terms. Thus, curve points close to a certain control point will have more influence on its energy than points far away. A similar approach is used in [11] where the authors are using MRFs for non-rigid image registration based on cubic B-splines. Such a weighting allows a suitable approximation of the energy terms with respect to the control points. For improving this approximation, we propose to reformulate the external energy from the continuous domain also as pairwise interaction terms. Modeling the external energy as pairwise terms has big advantages. The non-vanishing interval of basis functions along the curve domain for control point tuples is bigger than the interval corresponding to a single control point (see also Fig. 2(b) and 2(c)). Compared to unary potentials, the energy computation for the simultaneous movement of a pair of control points yields a more accurate approximation of the continuous energy[2]. Following these observations, we define the MRF energy as

$$E_{\text{mrf}} = \sum_{(i,j) \in \mathcal{E}} \left(\theta^{ij}_{\text{ext}}(x_i, x_j) + \lambda\, \theta^{ij}_{\text{int}}(x_i, x_j) \right) \tag{10}$$

where our discrete version of Eq. (4) is defined as

$$\theta^{ij}_{\text{ext}}(x_i, x_j) = \int_0^1 \hat{N}_{ij}(s)\, \psi(\mathcal{I}\left(C_{ij}(x_i, x_j, s)\right))\, ds \tag{11}$$

[2]Mind that for exact energy computation one would need to define cliques of size $d + 1$ where d is the degree of the B-spline basis functions.

and Eq. (5) is reformulated in the discrete domain as

$$\theta^{ij}_{\text{int}}(x_i, x_j) = \int_0^1 \hat{N}_{ij}(s) \left(1 - \frac{\|C'_{ij}(x_i, x_j, s)\|}{\|C'_{\text{init}}(s)\|}\right)^2 ds. \tag{12}$$

Here, the weighting coefficient $\hat{N}_{ij}(s)$ evaluates the pairwise influence of a curve point s to the energy of pair (i, j) and the curve function $C_{ij}(x_i, x_j, s)$ describes the *potential deformation* of a curve when two control points i and j are displaced simultaneously by $\mathbf{d}^{x_i}$ and $\mathbf{d}^{x_j}$, respectively. The potential deformation can be computed very efficiently since only certain parts of the curve affected by the deformation have to be recomputed

$$C_{ij}(x_i, x_j, s) = C(s) + N_i(s)\,\mathbf{d}^{x_i} + N_j(s)\,\mathbf{d}^{x_j}. \tag{13}$$

Similar to (12), other energy terms such as diffusion (6) and curvature (7) can be formulated in the discrete domain.

We consider two different versions for defining the influence functions, either through the addition of basis functions which we will call the *sum model* (see Fig. 2(b))

$$\hat{N}^{+}_{ij}(s) = \frac{N_i(s) + N_j(s)}{N_1(s) + 2\sum_{k=2}^{M-1} N_k(s) + N_M(s)}, \tag{14}$$

or through multiplication which we will call the *product model* (see Fig. 2(c))

$$\hat{N}^{*}_{ij}(s) = \frac{N_i(s) N_j(s)}{\sum_{k=1}^{M-1} N_k(s) N_{k+1}(s)}. \tag{15}$$

In both cases, the normalization is needed because the overall integral of the basis functions should be equal to one in order to preserve the energy. The performance of the two versions will be evaluated in our experiment section and compared to the naïve approach of modeling the external energy through unary potentials, i.e.

$$\theta^{i}_{\text{ext}}(x_i) = \int_0^1 N_i(s)\, \psi(\mathcal{I}\,(C_i(x_i, s)))\, ds. \tag{16}$$

We believe that our model is a good compromise between model accuracy and complexity. One could claim that the approximation error could be reduced (or even completely removed) if more complex models are used. However, the consideration of higher-order cliques [21] or high-dimensional label spaces is currently computationally intractable in real-time environments such as tracking. Another advantage of our MRF model is the capability of tracking multiple objects simultaneously by combining the single MRF energies, one per object, into one global labeling problem.

3.4. Optimization

Once our problem is formulated in a discrete setting, we need to choose an MRF optimization strategy. Fortunately, recent advances in discrete optimization brought a couple of very powerful techniques, mainly either based on iterative graph-cuts or efficient message passing. Regarding our specific model, there are two properties which should be considered when using one of the existing techniques. First, for the special case of open curves our graph is a tree (see Fig. 1) allowing the exact computation of the global optimal labeling when using max-product algorithms [20, 24] (e.g. Belief Propagation, TRW-S [15]). Second, our energy is nonsubmodular which is a (theoretical) problem for some methods using graph-cuts [17]. Using certain truncation techniques [22] on the energy terms make it still possible to use graph-cut based techniques (e.g. Expansion Move [5]). Another possibility for minimizing nonsubmodular functions is described in [16] but this technique might result in unlabeled nodes which is not appropriate in our setting. There are also methods based on iterative graph-cuts which can handle a wider class of MRF energies (e.g. Swap Move [5], FastPD [18]). Especially the FastPD algorithm is interesting in our case since a good performance with strong optimality properties is reported. A detailed review of the existing optimization methods is out of the scope of this paper. We refer the reader to the given references. In our experiment section we compare the performance of the TRW-S and the FastPD algorithm as representatives for the message-passing and graph-cut approaches. As we shall see, as usual there is a compromise between speed and accuracy.

4. Synthetic Experiments

In this section we present the performance analysis of our approach on synthetic data. Basically, we perform two different experiments to investigate particular model properties. The first experiment is dedicated to determine the influence of the approximation error in the likelihood potential functions. The second experiment determines the influence of different versions for discretization of the search space.

4.1. Approximation Error

In order to determine the influence of the approximation error we perform several tests on synthetic data. An initial open B-spline curve with six control points is deformed by assigning random labelings. After deformation, an image frame is generated by careful rasterization which we then use for the tracking algorithm. In this experiment we use the TRW-S as the optimization method since it can recover the global optimal solution for our energy function [20]. By knowing that the exact ground truth spline is within our dis-

Max. Def.	Model: Unary	Pairwise Sum	Pairwise Product
6	1.00 (± 0.43)	0.34 (± 0.19)	0.28 (± 0.17)
8	1.15 (± 0.52)	0.36 (± 0.23)	0.30 (± 0.19)
10	1.33 (± 0.60)	0.42 (± 0.25)	0.36 (± 0.22)
12	1.41 (± 0.70)	0.43 (± 0.27)	0.36 (± 0.23)
14	1.74 (± 0.82)	0.48 (± 0.30)	0.43 (± 0.28)
16	1.79 (± 0.93)	0.49 (± 0.32)	0.43 (± 0.29)
18	2.00 (± 1.04)	0.52 (± 0.34)	0.47 (± 0.32)
20	2.19 (± 1.15)	0.57 (± 0.40)	0.52 (± 0.34)

Table 1: Synthetic experiment for assessing the energy approximation error under different amounts of deformation and three different likelihood models. Reported are the average curve distances and standard deviations in pixels over one hundred frames per sequence.

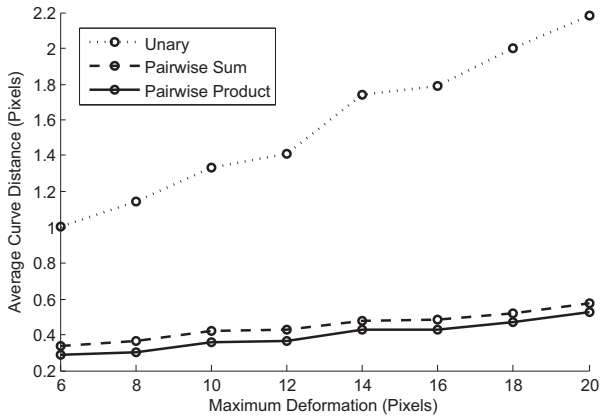


Figure 3: Plot of the average curve distances from Table 1.

crete label space, we can estimate the error induced by the energy approximation only. We generate eight sequences each consisting of one hundred frames. Different amounts of maximum deformation is applied in each sequence, from 6 to 20 pixels control point displacements. We evaluate both proposed pairwise versions, the sum model and the product model, as well as the naïve approach with unary potentials for the external energy. For a quantitative evaluation of the synthetic results, we compute the average curve distance (ACD) in pixels between the resulting curve C and the ground truth C_{gt} as

$$ACD(C, C_{\mathrm{gt}}) = \int_0^1 \mathrm{d}_{min}(C(s), C_{\mathrm{gt}})\, ds \qquad (17)$$

$$\mathrm{d}_{min}(P, C_{\mathrm{gt}}) = \arg\min_u |P - C_{\mathrm{gt}}(u)| \qquad (18)$$

The solution to equation (18) is found by minimizing

$$\frac{(P - C_{\mathrm{gt}}(u))^\top C'_{\mathrm{gt}}(u)}{|P - C_{\mathrm{gt}}(u)|\,|C'_{\mathrm{gt}}(u)|} \qquad (19)$$

w.r.t. u via Newton iterations.

Max. Def.	Steps	*SPARSE* TRW-S	*SPARSE* FastPD	*DENSE* TRW-S	*DENSE* FastPD
6	5	0.31 (± 0.20)	0.44 (± 0.30)	0.31 (± 0.18)	0.40 (± 0.27)
	10	0.32 (± 0.21)	0.44 (± 0.30)	0.31 (± 0.19)	0.41 (± 0.26)
	20	0.32 (± 0.21)	0.44 (± 0.30)	0.32 (± 0.19)	0.41 (± 0.26)
8	5	0.37 (± 0.26)	0.51 (± 0.40)	0.35 (± 0.22)	0.51 (± 0.36)
	10	0.37 (± 0.26)	0.52 (± 0.40)	0.35 (± 0.23)	0.48 (± 0.34)
	20	0.37 (± 0.26)	0.51 (± 0.38)	0.36 (± 0.23)	0.47 (± 0.32)
10	5	0.45 (± 0.33)	0.64 (± 0.49)	0.44 (± 0.27)	0.60 (± 0.42)
	10	0.44 (± 0.32)	0.60 (± 0.46)	0.43 (± 0.27)	0.57 (± 0.40)
	20	0.44 (± 0.31)	0.60 (± 0.46)	0.44 (± 0.27)	0.56 (± 0.37)
12	5	0.50 (± 0.39)	0.71 (± 0.57)	0.47 (± 0.31)	0.65 (± 0.47)
	10	0.49 (± 0.37)	0.69 (± 0.54)	0.47 (± 0.30)	0.64 (± 0.44)
	20	0.49 (± 0.37)	0.68 (± 0.52)	0.47 (± 0.30)	0.63 (± 0.43)
14	5	0.59 (± 0.45)	0.80 (± 0.63)	0.54 (± 0.36)	0.75 (± 0.54)
	10	0.57 (± 0.44)	0.77 (± 0.60)	0.54 (± 0.35)	0.71 (± 0.52)
	20	0.57 (± 0.43)	0.75 (± 0.59)	0.54 (± 0.34)	0.70 (± 0.53)
16	5	0.63 (± 0.52)	0.95 (± 0.86)	0.61 (± 0.41)	0.87 (± 0.73)
	10	0.62 (± 0.48)	0.88 (± 0.74)	0.59 (± 0.39)	0.82 (± 0.65)
	20	0.62 (± 0.48)	0.85 (± 0.70)	0.58 (± 0.39)	0.81 (± 0.64)
18	5	0.75 (± 0.66)	1.06 (± 0.99)	0.71 (± 0.49)	1.01 (± 0.86)
	10	0.74 (± 0.63)	1.05 (± 0.96)	0.70 (± 0.45)	0.90 (± 0.73)
	20	0.74 (± 0.60)	1.03 (± 0.94)	0.70 (± 0.45)	0.90 (± 0.71)
20	5	0.84 (± 0.73)	1.18 (± 1.11)	0.74 (± 0.55)	1.09 (± 0.97)
	10	0.81 (± 0.70)	1.10 (± 0.98)	0.74 (± 0.50)	0.99 (± 0.83)
	20	0.80 (± 0.67)	1.09 (± 0.96)	0.73 (± 0.49)	1.01 (± 0.89)
Time per Frame (ms)	5	146.61	16.52	1238.87	48.93
	10	566.74	31.30	$> 1.6 \cdot 10^4$	198.34
	20	2219.61	61.53	$> 23 \cdot 10^4$	904.95

Table 2: Synthetic experiment for comparing the sparse and dense deformation space discretization. Runtimes are assessed on a 2.16 GHz Intel Mobile CPU. ACDs are given in pixels.

The results are summarized in Table 1. The product model performs best on all sequences. Especially, when considering larger deformations the approximation error has a strong influence in case of unary potentials while both pairwise models still yield very good results of always less than one pixel. The error characteristics are also depicted in Fig. 3. Throughout this experiment we set $\lambda = 0$ since we do not want to penalize for length changes in case of synthetic deformation where the length preserving constraint does not hold.

4.2. Deformation Space Discretization

Our second experiment aims at the evaluation of different discretization strategies. Since the number of labels is an important parameter for the runtime of MRF optimization techniques, we want to determine a reasonable compromise between speed and tracking accuracy. We propose two different strategies for discretization, a *sparse* one and a *dense* one, see Fig. 1. Both versions are parametrized by two values, the number of sampling steps along a certain displacement direction and the range which defines the allowed maximum displacement. In case of sparse discretization, the deformation space is sampled along eight directions, namely horizontal, vertical and diagonal each in

positive and negative direction. In case of dense sets, we sample the complete square space at a control point. Given the number of steps S, we get $|\mathcal{L}_{\text{sparse}}| = 8S + 1$ including the zero-displacement. For the dense version we get $|\mathcal{L}_{\text{dense}}| = (S + 1)^2$. Similar to the first experiment, we generate eight synthetic sequences by assigning uniformly distributed random displacements on the six control points. Thus, this time the ground truth is not covered by our label space. The range of the label space is set to the maximum random deformation and we test different values for the number of sampling steps, namely 5, 10, and 20.

Again, we use the ACD as a measure of tracking quality. In this experiment we also evaluate the performance of two different optimization techniques, in particular TRW-S and FastPD. The results are summarized in Table 2. Again, we set $\lambda = 0$ and this time only the pairwise product model is used. The results show that the use of FastPD combined with sparse label sets is extremely efficient and reasonable tracking accuracy can be achieved. In fact, the difference between sparse and dense label sets is quite small regarding the tracking error while in case of sparse sets we achieve real-time performance in all experiments. Expectedly, the TRW-S gives the better results in terms of accuracy but it is not suitable for real-time applications. This conclusion encourages the use of FastPD and sparse discretization in our further application on real data.

5. Application

In the following, we show the successful application of our algorithm to the tracking of guide-wires in fluoroscopic image sequences (see Fig. 4(a)). Fluoroscopy is a modality in which low-dose X-ray videos are generated using intraoperative C-arms. The tracking information acquired from such sequences can be used to improve the navigation task of the physician by enhanced visualization techniques. Consequently, a reduction of X-ray dose is achieved and the treatment becomes less hazardous for patient and clinicians. Since fluoroscopic images are very noisy, a robust yet accurate method must be applied, which runs at 7-15 Hz.

5.1. Initialization

As a preprocessing step, we initialize our tracking algorithm with curves that are automatically extracted from the first frame of the sequence. To this end, we employ a line detector [23], which yields an ordered set of points for each curvilinear structure it detects. Afterwards, a B-spline is fitted to each of the point sets. For our application we use an algorithm, which minimizes discontinuity jumps in the k-th derivative when the desired curve has degree k [8]. In our case we choose a cubic B-spline model, i.e. $k = 3$.

5.2. Feature Image

Since our algorithm is tracking tubular structures, we use a feature image comprising the ridgeness measure proposed by Frangi et al. [9]. Their measure is a function of the eigenvalues λ_1 and λ_2 of the Hessian $\mathcal{H}$ and is computed as

$$\mathcal{I} = \begin{cases} 0 & \text{if } \lambda_2 > 0, \\ \exp(-\frac{R_B^2}{2\beta^2}) \left(1 - \exp(-\frac{S^2}{2\gamma^2})\right) & \end{cases} \tag{20}$$

where $|\lambda_1| \leq |\lambda_2|$. $R_B = \lambda_1/\lambda_2$ is a blobness measure corresponding to the eccentricity of the second order ellipse and $S^2 = |\mathcal{H}|_{\text{F}} = \lambda_1^2 + \lambda_2^2$ is a measure penalizing homogeneous regions in images. In all of our tests $\beta = 0.5$ and $\gamma = 5.0$. An example image is shown in Fig. 4(b).

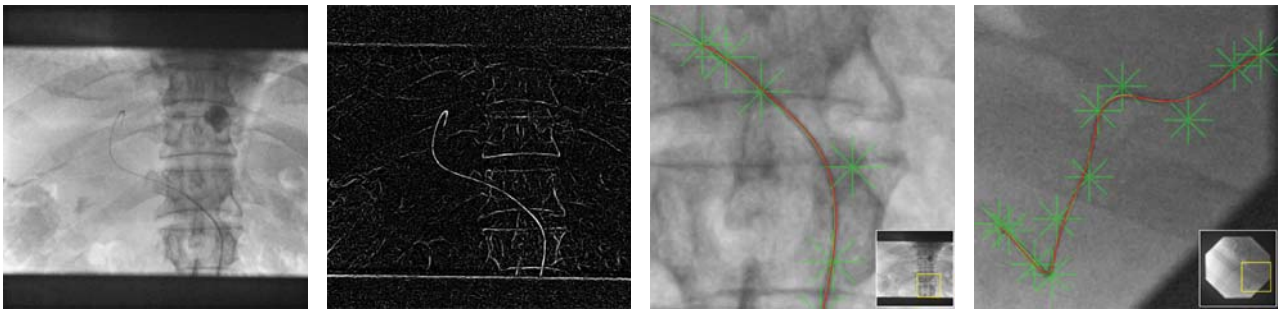
5.3. Performance Evaluation

We perform tests on two clinical sequences of 142 and 228 frames with a resolution of 512×512 pixels. In order to evaluate the tracking results, we manually segment the guide-wires in each frame. Throughout the experiment the length preserving term (12) is used where the regularization parameter is set to $\lambda = 0.9$. In the first sequence, the line detection results in a single initial curve whereas in the second sequence two curves are detected which are tracked simultaneously (see Fig. 4(c) and 4(d)). Throughout the experiments, we use the product model and the FastPD optimization on a sparse label space with 15 steps and a range of 15 pixels. The range is confirmed by clinical experts and corresponds to the expected maximum deformation regarding patient's breathing and physician's guidewire manipulation. Similar to our synthetic experiments, the ACD measure (17) is assessed for the accuracy analysis. The mean ACD over all frames is determined to be $0.35\ (\pm 0.29)$ pixels for the first sequence. The tracking error of the two curves in the second sequence is $0.22\ (\pm 0.20)$ and $0.31\ (\pm 0.27)$ pixels, respectively. Our method achieves real-time performance of more than 9 frames per second on a 2.16 GHz Intel Mobile CPU which includes the computation of the feature image.

6. Conclusion

In this paper, we introduce a novel framework for the fast tracking of parametrized curves. Based on an MRF formulation and efficient discrete optimization, our algorithm is shown to be generic, robust to poor features and high deformations of the object to be tracked, while an accuracy on a sub-pixel level can be continuously maintained. Moreover, hard real-time constraints can be met even if multiple curves are tracked at the same time.

The application of this paper is the tracking of open curves representing guide-wires in noisy fluoroscopic image sequences. However, the method can be easily extended



(a) Fluoro Image (b) Frangi Features (c) Sequence 1 (d) Sequence 2

Figure 4: Application of guide-wire tracking. (a) and (b) show a fluoroscopic frame and its respective feature image. (c) and (d) show exemplary tracking results for the two sequences (zoomed in). Green lines show the manual segmentation, red lines show the tracked curves. The green stars represent the sparse search space.

to closed curves and used for object contour detection and tracking. To this end, only the B-spline curve model has to be slightly altered while the core tracking can be left unchanged as it only relies on the control points of the spline.

Furthermore, we could benefit from learned shape priors estimated from a training set by defining an additional pairwise potential as

$$\theta^{ij}_{\text{prior}}(x_i, x_j) = -\log(p(d(\bar{C}, C_{ij}))) \qquad (21)$$

where p denotes the probability of a distance $d(\cdot,\cdot)$ between a mean shape $\bar{C}$ and C_{ij}.

References

[1] A. A. Amini, R. W. Curwen, and J. C. Gore. Snakes and splines for tracking non-rigid heart motion. In *ECCV*, London, UK, 1996. Springer-Verlag. 2

[2] A. A. Amini, T. E. Weymouth, and R. Jain. Using dynamic programming for solving variational problems in vision. *PAMI*, 12, 1990. 2

[3] A. Blake, R. Curwen, and A. Zisserman. A framework for spatiotemporal control in the tracking of visual contours. *IJCV*, 11, 1993. 2

[4] A. Blake and M. Isard. *Active Contours: The Application of Techniques from Graphics, Vision, Control Theory and Statistics to Visual Tracking of Shapes in Motion*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1998. 2

[5] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *PAMI*, 23(11), 2001. 5

[6] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. *IJCV*, 22, 1997. 2

[7] R. Cipolla and A. Blake. The dynamic analysis of apparent contours. In *ICCV*, 1990. 2, 3

[8] P. Dierckx. *Curve and Surface Fitting with Splines*. Oxford University Press, Inc., New York, NY, USA, 1993. 7

[9] A. F. Frangi, W. J. Niessen, K. L. Vincken, and M. A. Viergever. Multiscale vessel enhancement filtering. In *MICCAI*, 1998. 7

[10] D. Geiger, A. Gupta, L. A. Costa, and J. Vlontzos. Dynamic programming for detection, tracking, and matching deformable contours. *PAMI*, 17, 1995. 2

[11] B. Glocker, N. Komodakis, G. Tziritas, N. Navab, and N. Paragios. Dense image registration through mrfs and efficient linear programming. *Medical Image Analysis*, 12(6), 2008. 4

[12] M. Isard and A. Blake. Condensation conditional density propagation for visual tracking. *IJCV*, 29, 1998. 2

[13] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *IJCV*, V1(4), 1988. 2

[14] D. Kim. B-Spline representation of active contours. In *Symp. on Signal Processing and its Applications*, 1999. 2

[15] V. Kolmogorov. Convergent tree-reweighted message passing for energy minimization. *PAMI*, 28(10), 2006. 2, 5

[16] V. Kolmogorov and C. Rother. Minimizing nonsubmodular functions with graph cuts-a review. *PAMI*, 29(7), 2007. 5

[17] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? *PAMI*, 26(2), 2004. 5

[18] N. Komodakis, G. Tziritas, and N. Paragios. Fast, approximately optimal solutions for single and dynamic mrfs. In *CVPR*, 2007. 2, 5

[19] S. Z. Li. *Markov random field modeling in image analysis*. Springer-Verlag New York, Inc., 2001. 4

[20] J. Pearl. *Probabilistic Reasoning*. San Francisco, CA: Morgan Kaufmann, 1988. 5

[21] S. Ramalingam, P. Kohli, K. Alahari, and P. Torr. Exact inference in multi-label crfs with higher order cliques. In *CVPR*, 2008. 5

[22] C. Rother, S. Kumar, V. Kolmogorov, and A. Blake. Digital tapestry [automatic image synthesis]. In *CVPR*, 2005. 5

[23] C. Steger. An unbiased detector of curvilinear structures. *PAMI*, 20(2), 1998. 7

[24] Y. Weiss and W. Freeman. On the optimality of solutions of the max-product belief-propagation algorithm in arbitrary graphs. *IEEE Trans. On Information Theory*, 47(2), 2001. 5

[25] A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *ACM Comput. Surv.*, 38(4), 2006. 1, 2