

Interventional Tool Tracking using Discrete Optimization

Hauke Heibel, Ben Glocker, Martin Groher, Marcus Pfister and Nassir Navab

Abstract—This work presents a novel scheme for tracking of motion and deformation of interventional tools such as guide-wires and catheters in fluoroscopic X-ray sequences. Being able to track and thus to estimate the correct positions of these tools is crucial in order to offer guidance enhancement during interventions. The task of estimating the apparent motion is particularly challenging due to the low signal to noise ratio (SNR) of fluoroscopic images and due to combined motion components originating from patient breathing and tool interactions performed by the physician.

The presented approach is based on modeling interventional tools with B-splines whose optimal configuration of control points is determined through efficient discrete optimization. Each control point corresponds to a discrete random variable in a Markov random field (MRF) formulation where a set of labels represents the deformation space. In this context, the optimal curve corresponds to the maximum a posteriori (MAP) estimate of the MRF energy. The main motivation for employing a discrete approach is the possibility to incorporate a multi-directional search space which is robust to local minima. This is of particular interest for curve tracking under large deformation.

This work analyzes feasibility of employing efficient first-order MRFs for tracking. In particular it shows how to achieve a good compromise between energy approximations and computational efficiency. Experimental results suggest to define both the external and internal energy in terms of pairwise potential functions. The method was successfully applied to the tracking of guide-wires in fluoroscopic X-ray sequences of several hundred frames which requires extremely robust techniques. Comparisons with state-of-the-art guide-wire tracking algorithms confirm the effectiveness of the proposed method.

Index Terms—X-ray fluoroscopy, Guide-wire tracking, Markov random fields, Discrete optimization

I. INTRODUCTION

Minimally invasive procedures, in particular endovascular interventions have become clinical routine. The common task in these procedures is to navigate a medical tool to the intervention situs. During this navigation task, physicians rely on their mental three-dimensional knowledge of anatomical structures as well as haptic feedback. The additional use of imaging modalities such as ultra-sound or in case of angiographic interventions X-ray images is crucial for a successful navigation. It offers instant visual feedback while the tools are maneuvered through the patient’s body towards the desired situs.

The goal of this work is to estimate spatio-temporal information of interventional tools which is necessary for the development of a navigation solution as the one described in Section

Hauke Heibel, Martin Groher and Nassir Navab are with Computer Aided Medical Procedures (CAMP), Technische Universität München, Germany (Email: heibel@cs.tum.edu), Ben Glocker is with Microsoft Research Cambridge, Marcus Pfister is with Siemens Healthcare, Forchheim, Germany.

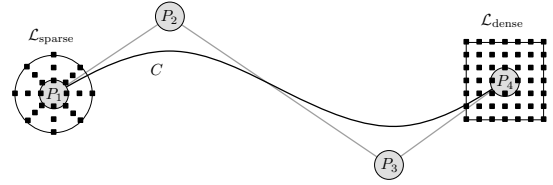


Fig. 1. MRF model of an open cubic B-spline curve C with control points P_i . A sparse and a dense version of the discrete set of labels \mathcal{L} capturing the multi-directional search space are illustrated (each black square corresponds to a displacement plus the zero-displacement at the control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

III. A possible approach to acquire this motion information is by means of tracking the object of interest between adjacent frames. The development of such an algorithm, however, is difficult for several reasons. As a consequence of the low energy X-rays used during fluoroscopic interventions, the images being used during the motion estimation usually exhibit a low SNR. Furthermore, since interventional tools are thin one-dimensional structures, they are oftentimes hard to distinguish from cluttered anatomical background such as bone and organ interfaces. Another difficulty which arises during tracking of guide-wires and catheters is the non-linearity of the motion. As opposed to rigid transformations, tools are deforming and changing their shape and size as a result of patient breathing, heart motion and tool interactions of the physician. Finally, since the images are generated by perspective projection of radio transparent structures the constant brightness assumption is violated which prohibits the use of well known optical flow techniques in order to perform the motion estimation.

This paper presents a novel framework for the tracking of curves as well as a method for the semi-automatic tracing of interventional tools which is required during the initialization. The derivation of a discrete scenario based on a MAP-MRF formulation allows to regard the tracking solution as the choice of an optimal labeling only comprising the control points of the curve (see Fig. 1). The discrete setting is appealing as it incorporates a *multi-directional search space* which is more robust to local minima. This is a major advantage compared to gradient-based approaches which follow only a single search direction, e.g. the one of steepest descent. In particular when the curve to be tracked undergoes large deformation between consecutive frames gradient descent may fail, while the discrete multi-directional search can still find the correct solution. The benefit of multi-directional search has also been recently shown in the context of mesh alignment by [Kainmueller et al. \(2010\)](#). The derivation of the discrete setting requires care to minimize model dependent approximation errors and several methods to diminish such errors are presented and

evaluated in this work. The tracking algorithm is evaluated on 17 clinical sequences and compared to a state-of-the-art guide-wire tracking algorithm confirming its effectiveness. The presented algorithm performs near real-time and proves to be both robust and accurate. The presented methods extend and improve the approach presented earlier in the work of Heibel et al. (2009).

The remainder of the paper is organized as follows. First, we will revisit previous work in the field of interventional tool tracking. In Section III we present the application context which requires image-based tool tracking. Afterwards, our method is described in detail in Section IV. A thorough evaluation is presented in Section V. It consists of three parts where we evaluate different components of our method. Finally, we conclude and discuss our approach in Section VI.

II. RELATED WORK

In this section we discuss existing approaches for tracking guide-wires during fluoroscopic interventions. Even though related, approaches tracking closed curves representing *contours* of objects (Yilmaz et al., 2006) are not discussed since guide-wires (as well as needles) can be adequately modeled only by open curves. To our best knowledge, so far none of the methods discussed in this section is yet fully suitable for everyday use in clinical practice.

The work of Palti-Wasserman et al. (1997) is the first to tackle the guide-wire tracking problem. It introduces the possibility to use guide-wire detection and tracking in order to extract functional information of the myocardium. The method detects a guide-wire by fitting a second-degree polynomial onto data produced by low-dose X-ray images filtered by a Laplacian or Marr-Hildreth kernel.

Baert et al. (2003a) propose a guide-wire tracking procedure based on energy minimization of a B-spline parameterization. The method is stratified, i.e. at first a rigid transformation of the guide-wire is estimated which is followed by computing the remaining deformable component through energy optimization via Powell’s direction set method. The external energy is based on feature images which are constructed by enhancing line-like structures of the initial spline position. For this work it is of interest to note that the authors conclude that they achieve a tracking precision “similar to the inter-observer variability” when using Hessian feature images. A second stratified approach similar to Baert et al. (2003a) is presented by Slabaugh et al. (2007). Here, however, the energy term is modeled such that smoothness is not only enforced but an additional soft constraint penalizes changes of the curve’s length.

A quite different approach has recently been presented by Wang et al. (2009). Here, the authors incorporate learning-based measurements for solving the tracking problem. Again, guide-wires are modeled by B-splines and the problem is solved in a stratified way. The objective functions are formulated in a Bayesian framework where a prior term is used to penalize deviations from the previously tracked curve. The likelihood term is a combination of the results of an AdaBoost classifier and an appearance-based measure in which intensity

differences to the previously detected curve are penalized. For the non-rigid tracking the authors add functions of the first- and second-order derivatives of the curves to regularize the optimization process.

Pauly et al. (2010) present an alternative tracking procedure also being based on learning. The work uses a likelihood term which utilizes mean orthogonal intensity profiles. The profiles are mapped through a learned support vector regression (SVR) to numerical values which can be interpreted as probabilistic measurements. Those measurements are computed for overlapping segments of the curve and their sum represents the curve’s likelihood term. The prior is ensuring length preservation and the optimization is carried out in a MAP-MRF framework as Heibel et al. (2009) propose.

The work of Petkovic and Lencaric (2010) proposes an enhancement method for guide-wires incorporating time by modeling intensity changes through the employment of Kalman filtering techniques. The drawback of the method is that it assumes a stationary patient in order to be able to establish a background estimate while at the same time requiring a moving guide-wire. The authors state that the approach does not perform well for parts of the guide-wire where it does not exhibit any motion.

Brost et al. (2010) use catheter tracking in a bi-plane C-arm scenario to reconstruct guide wires in 3D, while Ma et al. (2011) detect electrode features to improve catheter tracking. Both methods employ application-specific information additional to the actual guide-wire and are thus not discussed in more detail.

An overview of all relevant tracking methods including ours is given in Table I. The last entry of the table is intended to give the reader a rough idea of the magnitude of the individual algorithm’s run-times. Since the experiments were performed on separate systems with different processors and program environments, these values cannot be directly compared. Furthermore, the different methods offer different levels of potential regarding optimizations which follows that faster run-times may be achieved.

To summarize, all methods dealing with guide-wire tracking are using B-splines for modeling interventional tools (guide-wires, catheters and even needles since they may bend). However, all of the methods have drawbacks which hamper the every-day use in clinical practice: Petkovic and Lencaric (2010) cannot track guide-wires without motion, a situation likely to find in a clinical setting. Wang et al. (2009) and Pauly et al. (2010) require manually labeled training data to perform the tracking and cannot be applied right away to the multitude of different C-arms exhibiting a variety of parameters and thus output images. Baert et al. (2003a) and Slabaugh et al. (2007) use a stratified technique, which is implicitly slower than non-stratified methods. To the best of our knowledge there is still no fully robust and automatic technique available that can be used on the wide range of applications for guide-wire tracking.

III. APPLICATION CONTEXT

The application we envision for endovascular interventions is a navigation system similar to the ones proposed by

TABLE I
OVERVIEW OF DIFFERENT GUIDE-WIRE TRACKING METHODS.

method	parameter estimation	likelihood	prior	stratified	run-time (ms/frame) ^a
Baert et al.	continuous	different feature image	curvature	yes	5000 (0.2 fps)
Slabaugh et al.	continuous	feature image (phase congruency based)	curvature & length changes	yes	175 (5.7 fps)
Wang et al.	coarse-to-fine, exhaustive	AdaBoost classifier & intensity differences	displacements	yes	500 (2 fps)
Pauly et al.	discrete	SVR of orthogonal intensity profiles	length changes	no	660 (1.5 fps)
proposed method	discrete	feature image (Hessian & gradient based)	length changes	no	60 (16.7 fps)
Palti-Wasserman et al. ^b	n/a	feature image (mod. Laplacian & Marr-Hildreth)	n/a	n/a	n/a
Petkovic and Lencaric ^b	n/a	feature image (Kalman)	n/a	n/a	40 (25 fps)

^a The given run-times cannot directly be compared since they are assessed on different systems, but provide an idea of the magnitude of the algorithm run-times.

^b The last two methods are separated since they are posing the tracking problem slightly different and cannot directly be compared to the other methods.

van Walsum et al. (2005) or Bender et al. (2008). An optimal system should be leveraging pre-interventionally acquired image data being for instance generated by 3D rotational angiography (3DRA) which can be used to create a 3D model of the vessel system and in-turn to visualize the guide-wire position. The fused visualization of vessels and interventional tools could then be done directly in 3D or alternatively in a 2D overlay – a so called roadmap.

A typical recipe for such a navigation system can be formulated to contain the following core components:

- 1) Tool Detection
- 2) Motion Tracking
- 3) Data Annotation

In abdominal or cardiac interventions, the data annotation is hampered by patient motion, mainly induced by heart beat and respiration. Hence, tracking the tool in fluoroscopic images and a subsequent correction of the apparent motion is inevitable to provide a fully fledged navigation system. Such a system is expected to have advantages such as reduction of intervention time, improved success rates, fewer complications and lowered contrast dye application.

This work is primarily dealing with part 2 of the aforementioned components. Due to the self-similarity of guide-wires along the curve, it is difficult to develop a method which performs the motion estimation as well as the tool tip detection in a single step. Therefore, we focus on the tracking of a segment of the guide-wire. If the actual tool-tips were required, the provided motion data could be used to initialize an explicit end-point detection scheme such as the one proposed by Baert et al. (2003b).

IV. 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 the previous position of this object is known and its new position needs to be detected. In this work we will focus on the second phase though we will provide a semi-automatic strategy for handling the initialization. So called feature images are an important component of tracking algorithms since they usually comprise the external forces which drive the optimization process. Typically such images are acquired by enhancing edges, lines, or corners through image filtering techniques. For this work, we are using the well known feature images from Koller et al.

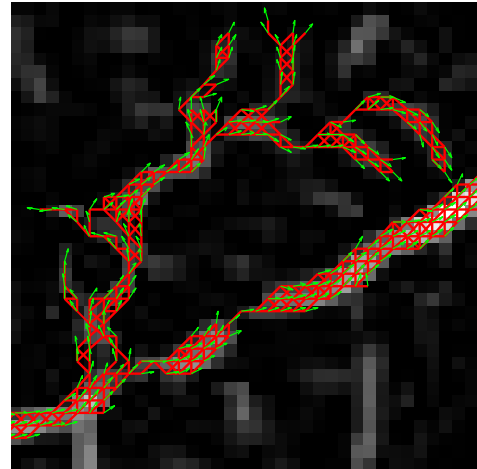


Fig. 2. Closeup of minimal initialization-graph with branching points. Red lines present graph edges and the green arrows the direction of principal curvature of graph nodes.

(1995). Alternatively, one could also make use of features as defined by Frangi et al. (1998). In the following, we provide an example for dealing with the initialization phase and then, we will focus on the tracking phase by introducing the curve model and our tracking algorithm.

A. Initialization

Currently, no method for the fully-automatic detection of guide-wires in fluoroscopy exists which produces results of sufficient accuracy (Mazouer et al., 2009). In this section we present an approach for the semi-automatic detection of guide-wires being based on shortest paths and structural information contained in the second-order derivatives. The approach is similar to the work recently presented by Schneider and Sundar (2010).

At first, our initialization procedure requires the user to select two points close to the extremities of the tool. The basic idea is that the algorithm tries to connect these two points by traversing pixels supposedly belonging to the interventional tool. Given the initial points, the system picks any of them and starts growing a graph by selectively adding neighbors of the current point. Once the neighbor-selection for a single point is done, the algorithm continues growing the graph on the yet unprocessed neighbors. The graph creation process is continued until the initially selected end-point has been added

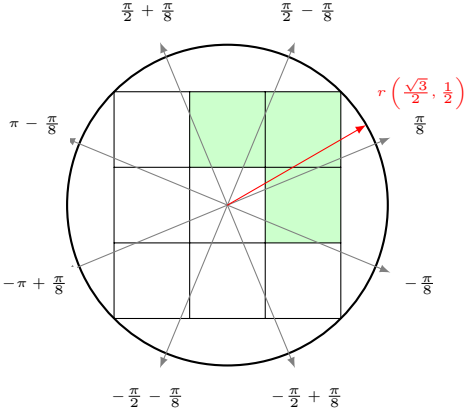


Fig. 3. Neighbor selection based on tangent direction \mathbf{t}_i (corresponding to the red vector). The green squares indicate the selected neighbors

to the graph or until no new neighbors can be added. If the graph does not already contain both initially selected points, the graph growing process is repeated from the initial point which is still missing. Again, the growing process continues until both points are present in the graph or until further growing fails. In the latter case the initialization would fail and require a retry. This would not be the case with methods which incorporate every single pixel in the graph (e.g., [Schneider and Sundar, 2010](#)) but the interactive nature of the proposed method outweighs the rare need of re-initializations.

As soon as the graph generation is finished, a connection between the two user selected nodes is computed via Dijkstra’s shortest path algorithm. A partial input graph can be found in Figure 2. This figure also depicts the fact that only a fraction of the pixels in the image is selected and that these pixels are furthermore located on structures looking similar to interventional tools.

We will now define how neighbors are chosen and how the edge weights for the shortest path estimation are computed. From differential geometry we know that the eigenvector of the Hessian of image intensities, corresponding to the eigenvalue having smallest absolute value is pointing in the direction of least (principal) curvature. In our case this means that the eigenvector is aligned tangential to the interventional tool. Such eigenvectors are represented by the arrows in Figure 2. Given a unit length tangential vector \mathbf{t}_i we choose the three neighbors (see green boxes in Figure 3) from the 8-neighborhood of point \mathbf{p}_i by investigating the tangent direction $\lambda = \text{atan2}(\mathbf{t}_i^y, \mathbf{t}_i^x)$. This angle will fall into one of eight sectors each spanning 45° and each defining a single triplet of neighbors. A neighbor \mathbf{p}_{i+1} is added to the graph if its corresponding *vesselness* \mathcal{I} is non-zero, i.e. if

$$\mathcal{I}(\mathbf{p}_{i+1}) > 0 \quad (1)$$

and if the angle γ between the current point’s tangent \mathbf{t}_i is pointing in a similar direction as the potential neighbor’s tangent \mathbf{t}_{i+1} , which is expressed as

$$\gamma = \arccos\left(\|\langle \mathbf{t}_i, \mathbf{t}_{i+1} \rangle\|\right) \leq 45^\circ. \quad (2)$$

The first condition from Equation (1) is based on the vesselness measure of [Koller et al. \(1995\)](#). This measure, referred to as $\mathcal{I} \in [0; 1]$, is non-zero if and only if the largest eigenvalue of the Hessian at point \mathbf{p}_{i+1} is large and positive which ensures that only dark ¹ and line-like structures are being traced. Equation (2) ensures that tangential vectors located at \mathbf{p}_i and \mathbf{p}_{i+1} are pointing in similar directions which is motivated by the assumption that the object shape we want to segment is smooth. Finally, weights for the edges connecting the points \mathbf{p}_i and \mathbf{p}_{i+1} need to be assigned. Here, we chose

$$w = (1 - \mathcal{I}(\mathbf{p}_{i+1})) \gamma^2$$

which implies high values for points \mathbf{p}_{i+1} having a small vesselness (i.e. a low edge-cost for those belonging to an interventional tool) and it also means that it is cheaper to connect points if the angle γ between their tangents is small. After running Dijkstra’s shortest path algorithm we obtain an ordered set of points

$$\mathbf{Q} = \{\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_L\}$$

which are representing the desired curve. This simple but effective method cannot deal well with objects having self-intersections. Then again, the method does not require any learning, is easy to control and straight forward to implement as opposed to the more complex approaches represented by [Mazouer et al. \(2009\)](#) and [Honnorat et al. \(2010\)](#).

In the next section we describe how the point set \mathbf{Q} can be represented in a compact way which enables more efficient optimization strategies and already imposes geometric constraints like certain smoothness on the curve.

B. Curve Model

B-spline curves represent a convenient way of modeling interventional tools. 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 curve²

$$C(s) = \sum_{i=1}^M N_i(s) P_i \quad \text{where } s \in [0, 1] \quad (3)$$

where N_i denote the basis functions and P_i the positions of M control points.

In order to track an interventional tool or object boundary, we seek an optimal fit to the data \mathbf{Q} by finding a compromise between *closeness* and *smoothness*. An implementation of such a method called `parcur` is publicly available in the library `FITPACK` [Dierckx \(1987\)](#). The method computes the optimal control points P_i as well as their number. In this work we are using B-splines of order 4 (cubic splines) for modeling guide-wires. The upper bound for the number of control points is set to 12.

¹They appear as bright features in the image \mathcal{I} after the enhancement as can be seen in Figure 2.

²Closed curves can be constructed by merging certain tuples of control points.

C. 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 MAP estimate. Given an observation \mathcal{I} (in our case a feature image), the MAP estimate is defined as

$$C^* = \arg \max_{C \in \mathbb{F}} P(\mathcal{I} | C) P(C) \quad (4)$$

where C^* is the optimal curve, $P(\mathcal{I} | C)$ is the *likelihood* probability of the estimate and $P(C)$ encodes the *prior* probability information on the set of feasible solutions \mathbb{F} . Assuming the probability distributions follow a Gibbs distribution with an underlying energy function, we can equivalently find the MAP via an energy minimization

$$C^* = \arg \min_{C \in \mathbb{F}} E(\mathcal{I} | C) + E(C) \quad (5)$$

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 term on the parameter space. In our scenario the likelihood term

$$E_{\text{ext}}(\mathcal{I} | C) = \frac{1}{l} \int_0^1 \psi(\mathcal{I}(C(s))) ds, \quad (6)$$

which is also referred to as the *external energy*, is driving the curve to its actual position. The variable l represents the curve length and is computed as

$$l = \int_0^1 \|C'(s)\| ds. \quad (7)$$

In Equation (6), ψ is used to make the feature image's values suitable for minimization. In our case, we simply chose $\psi(x) = 1 - x$ because we ensure that image responses are normalized to the interval $[0, 1]$. The prior term, also referred to as *internal energy*, constrains the motion of the curve. We use the same method as in our previous work (Heibel et al., 2009) where we penalize changes of the curve length compared to the initial curve. These changes are penalized via the squared relative error

$$\rho(v, v_{\text{init}}) = \left(1 - \frac{v}{v_{\text{init}}}\right)^2, \quad (8)$$

where v is the currently observed value and v_{init} the initial value we want to retain throughout the tracking. The actual length preserving regularization is then carried out by computing

$$E_{\text{int}}^{\text{len}}(C) = \frac{1}{l} \int_0^1 \rho(\|C'(s)\|, \|C'_{\text{init}}(s)\|) ds. \quad (9)$$

This kind of regularization prevents the curve from collapsing into a singular point. In the equations above C' denotes the derivative of the parametric curve C with respect to s .

The total energy of the curve tracking problem can be formulated as

$$E_{\text{total}} = (1 - \lambda) E_{\text{ext}} + \lambda E_{\text{int}}^{\text{len}} \quad (10)$$

where λ weighs the external against the internal energy.

In continuous optimization, minimizing the above energy is commonly done via gradient descent. 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 on the energy can be achieved, meaning the method converges to a local minimum. Even if sometimes convenient to use, such an approach has limitations. A particular issue is the initialization. It is assumed that the initial contour is sufficiently close to the correct solution, otherwise gradient descent is likely to converge to a bad solution. Assuming that previous curve positions yield good initializations limits the applicability to scenarios with small deformations. However, in practice many sequences exhibit large motions of up to 50 pixels (see Table IV). In our discrete approach, we incorporate a multi-directional search which allows to find solutions which are not constrained to the direction of the gradient, and thus, provides a robust tracking algorithm. The details are given in the following section.

D. Discrete Curve Tracking with MRFs

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 different possibilities for the discretization (a sparse and a dense one) of the deformation space are illustrated in Figure 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. A common approach for modeling the labeling task in terms of energy minimization is the usage of first-order MRFs (Li, 2001)

$$E_{\text{mrf}} = \sum_{i \in \mathcal{V}} \theta^i(x_i) + \sum_{(i,j) \in \mathcal{E}} \theta^{ij}(x_i, x_j) \quad (11)$$

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 and *independently* from all other nodes, they evaluate the cost for an assignment of a certain label x_i . In such a scenario the pairwise interaction terms are used to model regularization between neighboring nodes.

The assumption that the likelihood of a labeling can be computed from a sum of independent unary terms, however, is not valid in our scenario. Considering B-splines with higher-order basis functions, the effect of a single control point on the deformation of the curve cannot be modeled independently from its neighbors because neighboring basis function overlap. In order to overcome this problem, we propose a novel MRF model for the case of curve tracking using B-splines.

An important component of this model is the integration of the basis functions as weighting coefficients within the energy terms. Thus, curve points close to a certain control point will

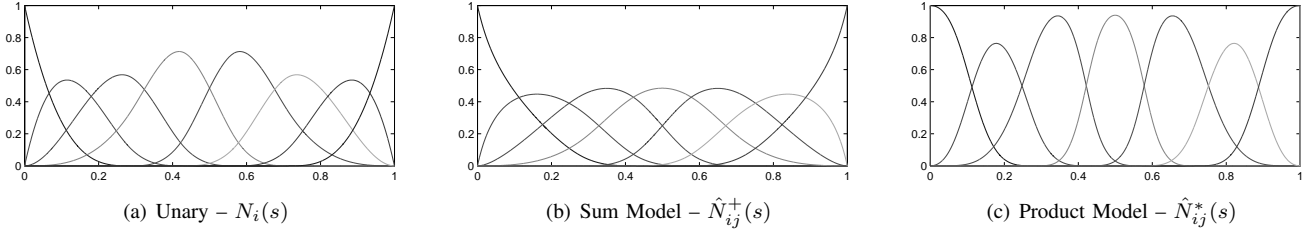


Fig. 4. Influence functions originating from an open cubic B-spline with eight control points.

have more influence on its energy than points farther away. A similar approach is used by [Glocker et al. \(2008\)](#) 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 and 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. In this work we define the MRF energy as

$$E_{\text{mrf}} = \frac{1}{l} \sum_{(i,j) \in \mathcal{E}} \left((1 - \lambda) \theta_{\text{ext}}^{ij}(x_i, x_j) + \lambda \theta_{\text{len}}^{ij}(x_i, x_j) \right). \quad (12)$$

The discrete version of the energy from Equation (6) is then composed of the pairwise terms

$$\theta_{\text{ext}}^{ij}(x_i, x_j) = \sum_{s \in \Omega_{ij}} \hat{N}_{ij}(s) \psi(\mathcal{I}(C_{ij}(x_i, x_j, s))). \quad (13)$$

Similarly we reformulate Equation (9) as

$$\theta_{\text{len}}^{ij}(x_i, x_j) = \sum_{s \in \Omega_{ij}} \hat{N}_{ij}(s) \rho(\|C_{ij}(x_i, x_j, s)\|, \|C_{\text{init}}(s)\|). \quad (14)$$

In the equations above, the weighting coefficients $\hat{N}_{ij}(s)$ evaluate the influence of a curve point s on the energy of a control point pair (i, j) . The integrals from Equations (6) and (9) are turned into discrete sums over s from the support $\Omega_{ij} = \text{supp}(\hat{N}_{ij}(s))$.

Intermediate deformations of the curve C are denoted by C_{ij} . In our earlier work ([Heibel et al., 2009](#)) these deformations were computed by applying the corresponding displacements \mathbf{d}^{x_i} and \mathbf{d}^{x_j} to the control points P_i and P_j only, which we revised in this paper. The initial modeling was problematic in the presence of large deformations because it was assuming that neighbors of P_i and P_j were not moving at all when the energies $\theta^{ij}(x_i, x_j)$ were evaluated. This led to an increase of the approximation errors with increasing deformations. The new approach assumes that neighboring control points exhibit similar motions which brought us to the

idea of applying the potential displacement \mathbf{d}^{x_i} to all control points P_1, \dots, P_i and likewise the displacement \mathbf{d}^{x_j} to the remaining control points P_j, \dots, P_M . This novel deformation model can be expressed as

$$C_{ij}(x_i, x_j, s) = C(s) + \sum_{l=1}^i N_l(s) \mathbf{d}^{x_i} + \sum_{l=j}^M N_l(s) \mathbf{d}^{x_j}. \quad (15)$$

Due to the locality of B-splines, Equation (15) contains only $d + 1$ basis functions $N_l(s)$ which are non-zero when the underlying spline has degree d . This means that only $d + 1$ control points and their associated displacements contribute to the deformation of $C_{ij}(x_i, x_j, s)$.

Regarding the influence functions $\hat{N}_{ij}(s)$, we are considering two different versions in this work – either through the addition of basis functions which we call the *sum model* (see Figure 4(b))

$$\hat{N}_{ij}^+(s) = \frac{N_i(s) + N_j(s)}{\sum_{l=1}^M N_l(s) + N_{l+1}(s)} \quad (16)$$

or through multiplication which we will refer to as the *product model* (see Figure 4(c))

$$\hat{N}_{ij}^*(s) = \frac{N_i(s)N_j(s)}{\sum_{l=1}^M N_l(s)N_{l+1}(s)}. \quad (17)$$

The normalization is required in order to ensure that the discretized version of the energy term from Equation (12) is correctly approximating the continuous version from Equation (10) which is true if all non-zero influence functions at a site s sum up to 1.

The performance of the two weighting functions is evaluated in our experiments in Section V-B and compared to the standard approach where external energies are modeled through unary potentials (see Figure 4(a)), i.e.

$$\theta_{\text{ext}}^i(x_i) = \sum_{s \in \Omega_{ij}} N_i(s) \psi(\mathcal{I}(C_i(x_i, s))) ds. \quad (18)$$

It turns out that the product model performs best of all methods due to its limited support and its related faster decay of the influence function. We show furthermore that our model represents a good compromise between model accuracy and complexity. The approximation error could be reduced (or even completely removed) if more complex MRF models are used. However, the consideration of higher-order cliques ([Ramalingam et al., 2008](#)) is computationally too time-consuming in real-time environments even despite recent developments ([Ishikawa, 2009](#)).

E. 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, typically based on iterative graph-cuts or efficient message passing. Regarding our specific model, two properties were considered when using one of the existing techniques. First, our graph is a chain (see Fig. 1) allowing the exact computation of the global optimal labeling when using max-product algorithms (e.g. Belief Propagation, [Pearl, 1988](#); [Weiss and Freeman, 2001](#) or TRW-S, [Kolmogorov, 2006](#)). Second, our energy is nonsubmodular which is a (theoretical) problem for some methods using graph-cuts ([Kolmogorov and Zabih, 2004](#)). However, certain truncation techniques ([Rother et al., 2005](#)) on the energy terms make it still possible to use graph-cut based techniques (e.g. Expansion Move, [Boykov et al., 2001](#)). There are also methods based on iterative graph-cuts which can handle a wider class of MRF energies (e.g. Swap Move, [Boykov et al., 2001](#) and FastPD, [Komodakis et al., 2007](#)). Yet another alternative for our kind of energy function is iterative quadratic pseudo-boolean optimization (QPBO, [Kolmogorov and Rother, 2007](#)) which is also able to minimize nonsubmodular functions. The particular choice of one optimizer over another is discussed in more detail in the sections of the individual experiments.

max. def.	model		
	unary	sum model	product model
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 II
SYNTHETIC EXPERIMENT FOR ASSESSING THE ENERGY APPROXIMATION ERROR UNDER DIFFERENT AMOUNTS OF DEFORMATION AND THREE DIFFERENT INFLUENCE MODELS. REPORTED ARE THE AVERAGE CURVE DISTANCES AND STANDARD DEVIATIONS IN PIXELS OVER ONE HUNDRED FRAMES PER SEQUENCE.

V. EVALUATION AND RESULTS

In this section we present the performance analysis of our approach. The evaluation consists of three parts. First, we show with a synthetic example that the optimal influence function is the *product model* from Equation (17). Next, we evaluate sparse versus dense label sets as well as how optimization approaches computing the optimal labeling solution compare to approximative algorithms. For the second part of the evaluation we choose TRW-S, FastPD as well as QPBO as representatives of the two possible optimization classes where TRW-S represents the class of algorithms computing optimal labelings, while the other two allow real-time performance. The last part presents evaluations of the tracking algorithm

on clinical data labeled by an expert and comparisons to the guide-wire tracking approach presented by [Wang et al. \(2009\)](#).

A. Error Assessment

Before continuing with the experiments, the assessed error measures need to be defined. As opposed to the commonly used Hausdorff distance a discretized and asymmetric error measure defined as

$$d(C, C_G, s) = \min_t (\|C(s) - C_G(t)\|_2) \quad (19)$$

is used in this work. In the Equation (19), C is the currently observed curve and C_G the manually annotated ground-truth curve. We compute the minimal distance between between the tracked and annotated curve. Evaluating Equation (19) at different sites s allows us to compute the mean tracking error, its standard deviation as well as the median. This error function is advantageous since it does not penalize single outliers which is important because only curve segments are tracked between adjacent frames. Furthermore, Equation (19) is corresponding to the error assessed in other works as e.g. [Wang et al. \(2009\)](#) and thus providing a better basis for comparing such methods.

Additional statistics, namely the percentage of *missed* annotated pixels and the percentage of *falsely* tracked pixels are computed for the comparison with the work of [Wang et al. \(2009\)](#). The *missed* pixels are pixels of the annotation having a distance to the tracked curve being higher than a specific threshold. Similarly, the *false* pixels are those on the tracked curve being farther away from the annotation as defined by the same threshold. Following with previous works, we choose this threshold to be equal to 3 pixels.

B. Approximation Error

Our first experiment is evaluating different ways of diminishing our inherent approximation errors. In order to measure 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 the deformation, an image frame is generated by careful rasterization. This frame is in turn used for the tracking algorithm. This procedure is repeated for a single sequence based on the same initial curve until 100 frames are generated. Overall, eight sequences are generated where each sequence is created with a different upper bound on the maximum deformation assigned by the random labelings.

In this experiment we use the TRW-S ([Kolmogorov, 2006](#)) as the optimization method since it can recover the global optimal solution for tree-like MRF topologies ([Pearl, 1988](#)). By knowing that the exact ground truth spline is within our discrete label space, we can estimate the error induced by the energy approximation only. Since each frame from every sequence is generated from the exact same initial curve and since in this experiment the system is reinitialized with that specific curve in every frame, we can discard the regularization term (i.e. set $\lambda = 0$) and thus remove any regularization dependent bias.

As mentioned before, different amounts of maximum deformations are used in each sequence, ranging from 6 to 20

max. def.	steps	sparse			dense		
		TRW-S	FastPD	QPBO	TRW-S	FastPD	QPBO
6	5	0.32(± 0.11)	0.44(± 0.14)	0.35(± 0.11)	0.30(± 0.10)	0.41(± 0.14)	0.33(± 0.12)
	10	0.32(± 0.11)	0.43(± 0.13)	0.33(± 0.11)	0.29(± 0.10)	0.40(± 0.12)	0.33(± 0.12)
	20	0.31(± 0.10)	0.43(± 0.13)	0.33(± 0.12)	0.29(± 0.10)	0.39(± 0.13)	0.32(± 0.11)
8	5	0.42(± 0.13)	0.59(± 0.28)	0.48(± 0.20)	0.37(± 0.12)	0.53(± 0.18)	0.43(± 0.16)
	10	0.41(± 0.14)	0.56(± 0.22)	0.45(± 0.18)	0.35(± 0.12)	0.48(± 0.18)	0.40(± 0.15)
	20	0.40(± 0.14)	0.55(± 0.22)	0.44(± 0.17)	0.35(± 0.12)	0.48(± 0.17)	0.40(± 0.14)
10	5	0.52(± 0.16)	0.76(± 0.29)	0.61(± 0.23)	0.48(± 0.12)	0.69(± 0.25)	0.57(± 0.24)
	10	0.51(± 0.15)	0.74(± 0.28)	0.59(± 0.23)	0.43(± 0.12)	0.66(± 0.25)	0.54(± 0.20)
	20	0.50(± 0.15)	0.73(± 0.29)	0.57(± 0.18)	0.44(± 0.12)	0.64(± 0.26)	0.54(± 0.22)
12	5	0.62(± 0.24)	0.87(± 0.39)	0.72(± 0.34)	0.53(± 0.20)	0.75(± 0.33)	0.65(± 0.31)
	10	0.60(± 0.23)	0.84(± 0.33)	0.69(± 0.29)	0.50(± 0.17)	0.70(± 0.26)	0.64(± 0.38)
	20	0.59(± 0.22)	0.82(± 0.33)	0.70(± 0.33)	0.48(± 0.16)	0.73(± 0.38)	0.61(± 0.36)
14	5	0.75(± 0.27)	0.98(± 0.39)	0.85(± 0.34)	0.59(± 0.22)	0.86(± 0.39)	0.75(± 0.36)
	10	0.71(± 0.29)	0.93(± 0.42)	0.84(± 0.39)	0.56(± 0.20)	0.82(± 0.42)	0.71(± 0.41)
	20	0.71(± 0.28)	0.93(± 0.44)	0.82(± 0.38)	0.56(± 0.21)	0.79(± 0.46)	0.71(± 0.39)
16	5	0.91(± 0.49)	1.30(± 0.75)	1.13(± 0.75)	0.69(± 0.33)	1.17(± 0.89)	0.95(± 0.70)
	10	0.88(± 0.47)	1.34(± 0.96)	1.04(± 0.65)	0.67(± 0.28)	1.24(± 0.97)	0.98(± 0.79)
	20	0.87(± 0.48)	1.33(± 0.89)	1.09(± 0.68)	0.66(± 0.29)	1.14(± 0.89)	0.94(± 0.77)
18	5	1.11(± 0.56)	1.47(± 0.90)	1.26(± 0.76)	0.89(± 0.54)	1.34(± 0.96)	1.22(± 0.87)
	10	1.07(± 0.56)	1.42(± 0.84)	1.20(± 0.70)	0.84(± 0.49)	1.29(± 0.95)	1.18(± 0.94)
	20	1.07(± 0.59)	1.48(± 0.90)	1.25(± 0.77)	0.84(± 0.51)	1.28(± 0.93)	1.13(± 0.84)
20	5	1.16(± 0.64)	1.54(± 0.81)	1.38(± 0.76)	0.96(± 0.46)	1.37(± 0.86)	1.20(± 0.74)
	10	1.09(± 0.68)	1.52(± 0.87)	1.27(± 0.70)	0.85(± 0.45)	1.40(± 0.98)	1.16(± 0.78)
	20	1.08(± 0.63)	1.51(± 0.84)	1.31(± 0.69)	0.83(± 0.38)	1.33(± 0.95)	1.14(± 0.69)
run-time (ms)	5	119.49	25.80	38.15	356.00	30.55	33.61
	10	493.13	39.53	51.42	$> 4.5 \cdot 10^4$	71.22	70.07
	20	1573.80	68.36	81.49	$> 1.1 \cdot 10^5$	303.56	284.03

TABLE III

SYNTHETIC EXPERIMENT FOR COMPARING THE SPARSE AND DENSE DEFORMATION SPACE DISCRETIZATION. RUN-TIMES ARE ASSESSED ON A 1.6 GHZ INTEL CORE I7. ALL ERRORS ARE REPORTED IN PIXELS. THROUGHOUT THE EXPERIMENT FASTPD WAS CONFIGURED WITH A MAXIMUM OF 100 ITERATIONS AND QPBO WITH 2 OUTER SWEEPS.

pixels control point displacements. We evaluate both proposed pairwise versions, the *sum model* and the *product model*, as well as the standard approach with unary potentials for the external energy. For a quantitative evaluation of the synthetic results, we evaluate the mean of the distances (and their standard deviations) from Equation (19) at different, uniformly sampled arc-length sites s .

The results are summarized in Table II. The *product model* performs best on all sequences because it exhibits a limited support. Moreover, it is in this way reducing approximation errors introduced by the incomplete modeling of the displacement inter-dependency. Especially, if larger deformations are considered the approximation error has a strong influence in case of unary potentials while both pairwise models still yield good results of an average error of less than one pixel.

C. Deformation Space Discretization

The second experiment is intended to evaluate different discretization strategies as well as the effect of approximating optimizers like FastPD or QPBO. 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 Figure 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. This construction yields a multi-

directional search space which is an important property for curve tracking applications. 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. The main difference is that 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.

Another difference is that the initial curve used during the tracking is not parameterized in the same way as the curve used to generate the synthetic data. Therefore, the experiment does not only reflect errors resulting from deformations which are not covered by the label space but also those errors resulting from a sub-optimal curve parameterization.

Again, we use the mean distance (and its standard deviation) as a measure of tracking quality. TRW-S optimizer is chosen since it represents a state-of-the-art algorithm computing the optimal labeling for our energy term (recall that our MRF is a chain). On the approximation side, we chose FastPD and QPBO because of their speed and good optimality properties.

Table III shows that the use of FastPD combined with

sequence name	frames	spacing	deformation		curve length	
			mean	max	mean	max
cardiac 01	55	0.184	17	28	413	438
cardiac 02	72	0.184	13	25	440	457
cardiac 03	30	0.184	16	43	505	543
cardiac 04	10	0.278	20	51	432	437
cardiac 05	11	0.278	13	49	645	665
abdominal 01	143	0.432	8	20	170	175
abdominal 02	228	0.308	8	19	78	83

TABLE IV

SEQUENCE PROPERTIES. THE PIXEL SPACING IS GIVEN IN *mm* AND THE DEFORMATION AND LENGTH PARAMETERS ARE DENOTED IN *pixels*.

category	parameter	
	name	value
spline modeling	degree	3
	max. control points	12
discretization	search steps	40 (321 labels)
	search range	15 (mm)
regularization	λ	0.7

TABLE V
TRACKING PARAMETERS.

sparse label sets is extremely efficient and reasonable tracking accuracy can be achieved. Furthermore, it shows that QPBO achieves tracking results which are even closer to the optimal ones while introducing a minor runtime penalty. The difference between sparse and dense label sets is quite small regarding the tracking error while in case of sparse sets we achieve near real-time performance in all experiments. As expected, TRW-S gives the better results in terms of accuracy but it is not suitable for real-time applications. In our previous work (Heibel et al., 2009) these experiments lead us to choosing FastPD as our optimizer which we revise in this work because of the improved stability of QPBO.

D. Experiments on clinical data

In the final experiment we evaluate our tracking algorithm on 17 hand-labeled clinical sequences out of which 5 were acquired during cardiac (diagnostic catheterizations) and 2 during abdominal (transarterial chemoembolization; TACE) interventions with an AXIOM-Artis³ system at a cine rate of 15 frames per second. The remaining 10 sequences were acquired during a cardiac angioplasty. The challenging sequences contain an overall of more than 2500 frames and deformations of up to 50 pixels. More detailed properties of the individual sequences can be found in Table IV. The first 5 cardiac data sets were made available to us by the authors Wang et al. (2009) and each of the data sets did not only contain image data but also ground-truth annotations as well as the tracking results of the author’s algorithm. The abdominal sequences were provided by our clinical partners; they were acquired during TACE interventions and as the cardiac sequences, hand-labeled by an expert. The remaining cardiac sequences were provided by the authors of Honnorat et al. (2010).

³Siemens Healthcare, Forchheim, Germany

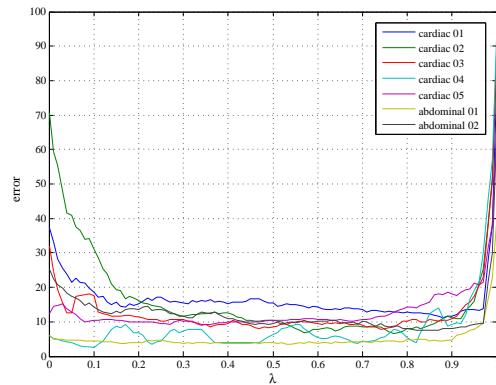


Fig. 5. The plot shows the evaluation of the regularization parameter λ and its dependency on the mean of false and missed tracking rates. The plot indicates that a wide variety of parameter choices is possible without dramatically hampering the tracking quality.

We use a single set of parameters (see Table V) throughout all experiments. The regularization weight is set to $\lambda = 0.7$ though Figure 5 indicates that other choices are possible without severely decreasing the algorithm’s performance. It should be noted here, that Figure 5 shows a mixture of false and missed tracking rates because it is desired to minimize both values at the same time instead of a single performance measure. The figure shows the results of an experiment in which we evaluated the tracking error (the mean of false and missed rates) for 100 different parameters λ and for each medical sequence.

The labeling is configured with a search range of 15 mm and 40 search steps in each direction resulting in an overall of 321 labels per node. The sparse labeling is chosen accordingly to Section V-C and following the results of Section V-B the *product model* is used as our influence function.

Regarding the algorithm run-time, it is useful to split the overall processing into two steps. The first step, before the optimization is carried out, is the computation of the feature image \mathcal{I} and the second step is the actual optimization process. We have measured a mean processing time of about 33.52 ms ($\sigma = 13.06$ ms) for the computation of the feature image \mathcal{I} and a mean optimization time of 26.22 ms ($\sigma = 9.60$ ms). This results in an overall of about 60 ms processing time. With these processing rates this is to the best of our knowledge the first algorithm which is capable of processing fluoroscopic images at the same rate as they are acquired by the imaging system.

A summary of the comparison between the two methods can be found in Table VI. The experiments show that our method is performing on average equally well as compared to the robust guide-wire tracking (RGWT) by Wang et al. (2009) while it is not relying on any learning. Visual results of the tracking performance are depicted in Figure 6. The results show that the tracking may suffer from the introduction of local loops and foldings in areas of high curvature but it also shows corresponding images from the same sequences in which the algorithm recovered from such intermediate errors. The first row of Figure 6 shows in particular the robustness to poor features where the algorithm is capable of maintaining correct curve shapes despite the lack of image information.

sequence name	mean (px)		standard deviation (px)		median (px)		false (%)		missed (%)	
	MRF-T	RGWT	MRF-T	RGWT	MRF-T	RGWT	MRF-T	RGWT	MRF-T	RGWT
cardiac 01	1.51	1.67	1.87	1.64	0.98	1.17	10.78	12.21	15.62	13.51
cardiac 02	1.52	1.57	2.72	1.81	0.71	0.89	7.15	11.37	7.65	9.94
cardiac 03	1.49	1.29	2.41	1.51	0.69	0.73	8.86	9.19	9.88	11.16
cardiac 04	0.85	0.82	0.92	0.73	0.61	0.69	3.11	1.44	2.00	0.61
cardiac 05	1.14	1.33	1.32	1.49	0.69	0.82	9.10	10.30	8.80	11.85
abdominal 01	0.81	n/a	0.63	n/a	0.68	n/a	1.87	n/a	4.13	n/a
abdominal 02	1.19	n/a	1.37	n/a	0.78	n/a	6.15	n/a	18.75	n/a
sequence 01	3.16	n/a	6.07	n/a	1.17	n/a	7.00	n/a	8.40	n/a
sequence 02	3.90	n/a	9.23	n/a	1.22	n/a	5.60	n/a	12.40	n/a
sequence 03	1.40	n/a	2.90	n/a	0.78	n/a	0.67	n/a	0.60	n/a
sequence 04	1.19	n/a	2.32	n/a	0.89	n/a	1.80	n/a	0.00	n/a
sequence 05	1.34	n/a	0.40	n/a	1.31	n/a	0.00	n/a	0.80	n/a
sequence 06	1.37	n/a	1.69	n/a	1.03	n/a	1.00	n/a	1.00	n/a
sequence 07	1.27	n/a	0.28	n/a	1.33	n/a	0.20	n/a	0.20	n/a
sequence 08	1.19	n/a	0.37	n/a	1.20	n/a	0.00	n/a	0.20	n/a
sequence 09	1.46	n/a	2.74	n/a	1.01	n/a	0.60	n/a	2.60	n/a
sequence 10	1.48	n/a	2.80	n/a	0.84	n/a	7.00	n/a	6.80	n/a

TABLE VI

EXPERIMENTAL VALIDATION OF THE PROPOSED METHOD (MRF-T) ON CLINICAL DATA. WHERE THE DATA WAS AVAILABLE, THE METHOD HAS BEEN COMPARED TO THE ROBUST GUIDE-WIRE TRACKING (RGWT) METHOD OF [WANG ET AL. \(2009\)](#). TRACKING ERRORS ARE REPRESENTED IN PIXELS (PX) AND THE FALSE AND MISSED RATES IN PERCENT OF THE CURVE LENGTH. THE THRESHOLD FOR COMPUTING FALSE AND MISSED RATES WAS SET TO 3 PIXELS.

VI. DISCUSSION

In catheterization procedures, the tip of the guide-wire is the actual object of interest for navigating the catheter to the intervention situs. Its detection is therefore an important task in order to provide computer aided navigation. The guide-wire tip is usually detected by explicitly searching for end-points of a curvilinear object starting from only a segment of the curve ([Baert et al., 2003b](#)). In this paper we propose a novel method which allows to track curve segments which is the prerequisite for the actual tip detection.

The proposed approach makes use of efficient discrete, multi-directional search which is robust to large deformations. No laborious off-line training stage is required opposed to alternative methods based on learning. Our method achieves highly accurate results in a single step and is computationally more efficient allowing near real-time performance compared to stratified, multi-stage methods.

The evaluation of our experimental results has shown that the underlying feature images are some time causing a performance degradation of the tracking. The problem is that in the absence of the actual guide-wire the tracked curves are being attracted by the surrounding edges (organ interfaces, bony structures). This is usually caused by poor visibility of the guide-wire in the X-Ray images and typically spans multiple successive frames of angiographic sequences. Overcoming this issue is difficult and a possibility to further improve the tracking algorithm is the incorporation of heuristics which allow to detect such cases. One approach might be to try to retain the shape of the guide-wire as observed the last time when it was completely visible. This issue is hampering all published methods which deal with guide-wire tracking during these sequences of poor visibility and thus requires further investigation in the future.

With the currently used regularization approach it may still happen that loops (see Figure 6(k)) appear in a few

intermediate frames. Another issue is the rare occurrence of curve folding (see Figure 6(h)) which is not averted by our regularization strategy. Problem specific regularization priors could be introduced to prevent these issues. Similarly to the proposal for dealing with poor visibility the penalization of shape changes between frames could help to improve these cases. In contrast to the penalization of length changes which is carried out in our work against the initial curve C_{init} it is not immediately clear against which curve shape changes should be penalized. Using the curve observed in the previous frame could lead to an error accumulation and is in particular problematic once the curve actually ends up in a wrong configuration. The alternative of using the initial curve is again problematic when it comes to shape penalizations because the guide-wire might actually be changing its shape during the intervention. Again, this problem offers research opportunities for future work.

VII. ACKNOWLEDGMENT

This research was funded by an academic grant from Siemens Healthcare Angiography/X-Ray division, Forchheim, Germany. The authors would like to thank in particular Klaus Klingenberg for his continuous support. Further thanks go to Peng Wang, Terrence Chen and Dorin Comaniciu from Siemens Corporate Research for granting access to their research results without which the comparative experiments from Section V could not have been performed. Last but not least, the effort of Nikos Komodakis, Nikos Paragios and Hiroshi Ishikawa of making their discrete optimization software publicly available is highly appreciated and was of great value during the development of this work.

REFERENCES

D. Kainmueller, H. Lamecker, H. Seim, S. Zachow, and H.-C. Hege, "Improving deformable surface meshes through

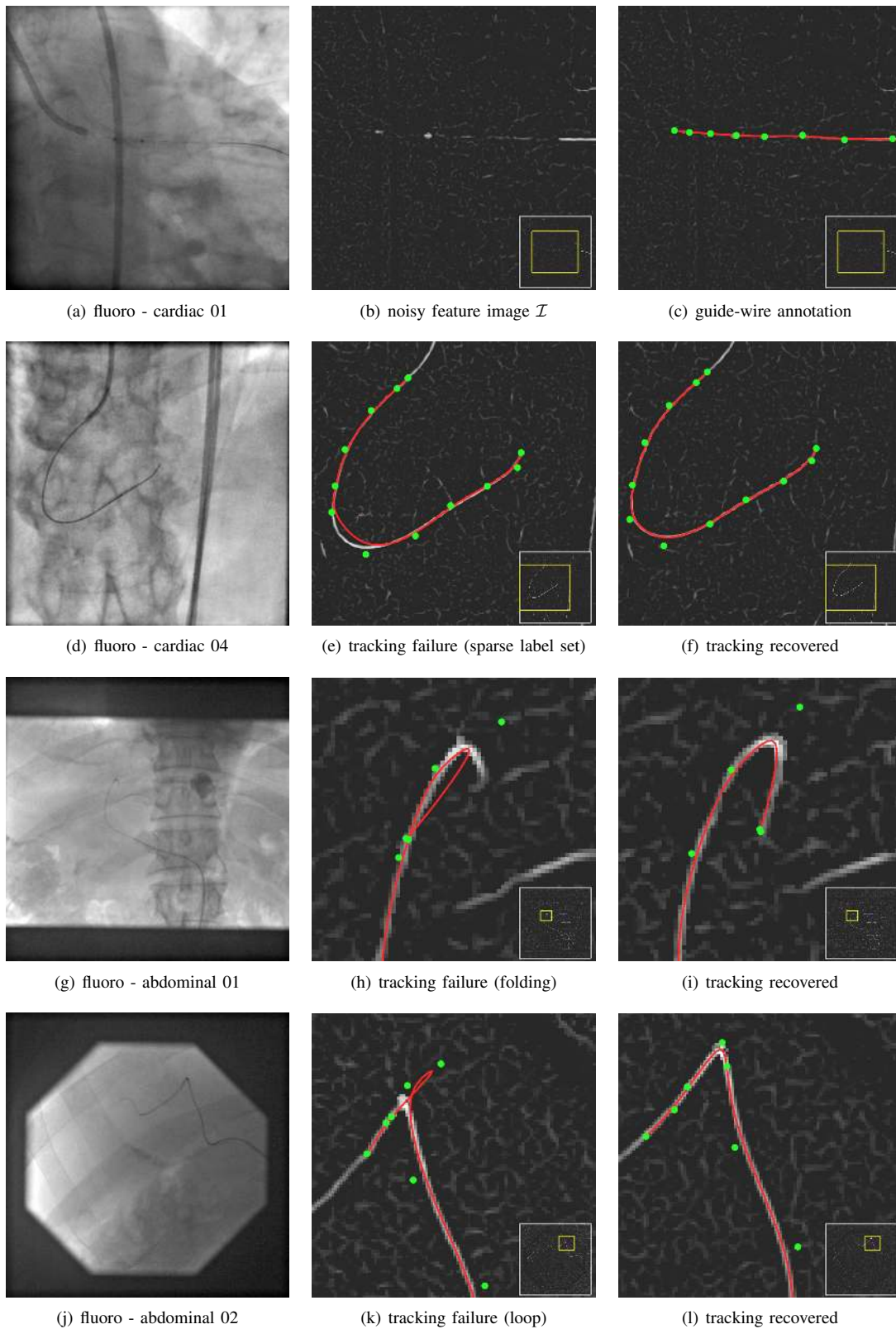


Fig. 6. Tracking results from selected clinical sequences. The images were slightly modified in their brightness and color levels in order to improve the display. The first row shows that the algorithm is able to deal with noisy feature images \mathcal{I} and the remaining three rows depict the three sources of intermediate tracking errors – a too sparse label set, foldings and the introduction of loops. In all three cases, the algorithm was capable of recovering from these errors.

- omni-directional displacements and MRFs,” in *Proc. Int’l Conf. Medical Image Computing and Computer Assisted Intervention (MICCAI)*, Beijing, China, September 2010.
- T. H. Heibel, B. Glocker, M. Groher, N. Paragios, N. Komodakis, and N. Navab, “Discrete tracking of parametrized curves,” in *Proc. IEEE Computer Vision and Pattern Recognition (CVPR)*, 2009.
- S. Baert, M. Viergever, and W. Niessen, “Guide-wire tracking during endovascular interventions,” *IEEE Trans. Med. Imag. (TMI)*, vol. 22, no. 8, pp. 965–972, Aug. 2003.
- G. G. Slabaugh, K. Kong, G. B. Unal, and T. Fang, “Variational guidewire tracking using phase congruency,” in *Proc. Int’l Conf. Medical Image Computing and Computer Assisted Intervention (MICCAI)*, 2007.
- P. Wang, T. Chen, Y. Zhu, W. Zhang, S. K. Zhou, and D. Comaniciu, “Robust guidewire tracking in fluoroscopy,” in *Proc. IEEE Computer Vision and Pattern Recognition (CVPR)*, 2009.
- O. Pauly, H. Heibel, and N. Navab, “A machine learning approach for deformable guide-wire tracking in fluoroscopic sequences,” in *Proc. Int’l Conf. Medical Image Computing and Computer Assisted Intervention (MICCAI)*, 2010.
- D. Palti-Wasserman, A. Brukstein, and R. Beyar, “Identifying and tracking a guide wire in the coronary arteries during angioplasty from X-ray images,” *IEEE Trans. Biomed. Eng.*, vol. 44, no. 2, pp. 152–164, 1997.
- T. Petkovic and S. Lencaric, “Guidewire tracking with projected thickness estimation,” in *IEEE Int’l Symp. on Biomedical Imaging: Macro to Nano (ISBI)*, 2010.
- A. Yilmaz, O. Javed, and M. Shah, “Object tracking: A survey,” *ACM Comput. Surv.*, vol. 38, no. 4, p. 13, 2006.
- A. Brost, R. Liao, N. Strobel, and J. Hornegger, “Respiratory motion compensation by model-based catheter tracking during EP procedures,” *Medical Image Analysis*, vol. 14, no. 5, pp. 695–706, 2010.
- Y. Ma, G. Gao, G. Gijssbers, C. A. Rinaldi, J. S. Gill, R. Razavi, and K. S. Rhode, “Image-based automatic ablation point tagging system with motion correction for cardiac ablation procedures,” in *IPCAI*, 2011, pp. 145–155.
- T. van Walsum, S. A. M. Baert, and W. J. Niessen, “Guide Wire Reconstruction and Visualization in 3DRA Using Monoplane Fluoroscopic Imaging,” *IEEE Transactions on Medical Imaging*, vol. 24, no. 5, pp. 612–623, May 2005.
- F. Bender, M. Groher, A. Khamene, W. Wein, T. H. Heibel, and N. Navab, “3D dynamic roadmapping for abdominal catheterizations,” in *Proc. Int’l Conf. Medical Image Computing and Computer Assisted Intervention (MICCAI)*, New York, USA, September 2008.
- S. Baert, T. van Walsum, and W. J. Niessen, “Endpoint localization in guide wire tracking during endovascular interventions,” *Academic Radiology*, vol. 10, pp. 1424–1432, 2003.
- T. Koller, G. Gerig, G. Szekely, and D. Dettwiler, “Multiscale detection of curvilinear structures in 2D and 3D image data,” in *Proc. Int’l Conf. of Computer Vision (ICCV)*, 1995, pp. 864–869.
- A. F. Frangi, W. J. Niessen, K. L. Vincken, and M. A. Viergever, “Multiscale vessel enhancement filtering,” *Lecture Notes in Comput. Sci.*, vol. 1496, pp. 130–137, 1998.
- P. Mazouer, T. Chen, Y. Zhu, P. Wang, P. Durlak, J.-P. Thiran, and D. Comaniciu, “User-constrained guidewire localization in fluoroscopy,” in *Proc. SPIE Medical Imaging (SPIE)*, 2009.
- M. Schneider and H. Sundar, “Automatic global vessel segmentation and catheter removal using local geometry information and vector field integration,” in *IEEE Int’l Symp. on Biomedical Imaging: Macro to Nano (ISBI)*, 2010.
- N. Honnorat, R. Vaillant, and N. Paragios, “Guide-wire extraction through perceptual organization of local segments in fluoroscopic images,” in *MICCAI*, 2010, pp. 440–448.
- P. Dierckx, “FITPACK user guide part 1: curve fitting routines,” Department of Computer Science, Katholieke Universiteit Leuven, Belgium, Tech. Rep. 89, 1987.
- S. Z. Li, *Markov random field modeling in image analysis*. Springer-Verlag New York, Inc., 2001.
- B. Glocker, N. Komodakis, G. Tziritas, N. Navab, and N. Paragios, “Dense image registration through MRFs and efficient linear programming,” *Medical Image Analysis*, vol. 12, no. 6, pp. 731–741, 2008.
- S. Ramalingam, P. Kohli, K. Alahari, and P. Torr, “Exact inference in multi-label CRFs with higher order cliques,” in *Proc. IEEE Computer Vision and Pattern Recognition (CVPR)*, 2008.
- H. Ishikawa, “Higher-order clique reduction in binary graph cut,” in *Proc. IEEE Computer Vision and Pattern Recognition (CVPR)*, 2009.
- J. Pearl, *Probabilistic Reasoning*. San Francisco, CA: Morgan Kaufmann, 1988.
- 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*, vol. 47, no. 2, pp. 736 – 744, 2001.
- V. Kolmogorov, “Convergent tree-reweighted message passing for energy minimization,” *IEEE Trans. Pattern Anal. Mach. Intell. (PAMI)*, vol. 28, no. 10, pp. 1568 – 1583, 2006.
- V. Kolmogorov and R. Zabih, “What energy functions can be minimized via graph cuts?” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 26, no. 2, pp. 147–159, 2004.
- C. Rother, S. Kumar, V. Kolmogorov, and A. Blake, “Digital tapestry,” in *Proc. IEEE Computer Vision and Pattern Recognition (CVPR)*, 2005.
- Y. Boykov, O. Veksler, and R. Zabih, “Fast approximate energy minimization via graph cuts,” *IEEE Trans. Pattern Anal. Mach. Intell. (PAMI)*, vol. 23, no. 11, pp. 1222–1239, 2001.
- N. Komodakis, G. Tziritas, and N. Paragios, “Fast, approximately optimal solutions for single and dynamic MRFs,” in *Proc. IEEE Computer Vision and Pattern Recognition (CVPR)*, 2007.
- V. Kolmogorov and C. Rother, “Minimizing nonsubmodular functions with graph cuts - a review,” *IEEE Trans. Pattern Anal. Mach. Intell. (PAMI)*, vol. 29, no. 7, pp. 1274 – 1279, 2007.