

RESEARCH ARTICLE

An Efficient Object Tracking Method on Quad-/Oc-Trees

Magda Przybylowski^{2*}, Pratim Ghosh³, Frederic Gibou^{1,2}

1 Department of Computer Science, University of California Santa Barbara, Santa Barbara, CA 93106-5110, United States of America, **2** Department of Mechanical Engineering, University of California Santa Barbara, Santa Barbara, CA 93106-5070, United States of America, **3** Department of Electrical and Computer Engineering, University of California Santa Barbara, Santa Barbara, CA 93106-5110, United States of America

* magda.przyby@gmail.com

Abstract

We introduce a fast error-free tracking method applicable to sequences of two and three dimensional images. The core idea is to use Quadtree (resp. Octree) data structures for representing the spatial discretization of an image in two (resp. three) spatial dimensions. This representation enables one to merge into large computational cells the regions that can be faithfully described with such a coarse representation, thus significantly reducing the total number of degrees of freedom that are processed, without compromising accuracy. This encoding is particularly effective in the case of algorithms based on moving fronts, since the adaptive refinement provides a natural means to focus the processing resources on information near the moving front. In this paper, we use an existing contour based tracker and reformulate it to the case of Quad-/Oc-tree data structures. Relevant mathematical assumptions and derivations are presented for this purpose. We then demonstrate that, on standard biomedical image sequences, a speed up of 5X is easily achieved in 2D and about 10X in 3D.



OPEN ACCESS

Citation: Przybylowski M, Ghosh P, Gibou F (2016) An Efficient Object Tracking Method on Quad-/Oc-Trees. PLoS ONE 11(3): e0150889. doi:10.1371/journal.pone.0150889

Editor: Philip Allen, University of Akron, UNITED STATES

Received: November 11, 2015

Accepted: February 19, 2016

Published: March 17, 2016

Copyright: © 2016 Przybylowski et al. This is an open access article distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Data Availability Statement: All relevant data are within the paper.

Funding: This research was supported by the National Science Foundation under agreement CHE 1027817.

Competing Interests: The authors have declared that no competing interests exist.

1 Introduction

Tracking objects in image sequences is a fundamental problem in computer vision. Error-free tracking is essential for various vision applications such as surveillance, security systems, and medical/biological image analysis. However, data uncertainty presents significant challenges for reliable object tracking. This includes, for example, illumination changes across consecutive frames, rigid/nonrigid deformation of the object of interest, partial occlusion, or data corruption.

Numerous frameworks have been proposed in recent years to address the above-mentioned problems. For example, a class of methods includes deformable shape models [1], active shape models (ASMs) [2] and active appearance models (AAMs) [3], which capture the statistics of shape and appearance variations using training examples. By combining ASM and AAM in a multiscale fashion, Mitchell *et al.* [4] achieved robustness against noise and clutter. Tsai *et al.* [5] applied principal component analysis (PCA) to model the variability in the training shapes represented by signed distance functions (SDFs). Zhu *et al.* [6] proposed a subject-specific

dynamic model designed for medical application using multilinear PCA. A potential drawback of these methods is that their performance depends on the training data, which may not encompass all possible scenarios.

Recently, graph-based algorithms have been proposed for tracking deformable objects. For example, Ishikawa and Jermyn [7] developed a polynomial time algorithm to extract similar objects from a set of time sequence images. Schoenemann and Cremers [8] proposed a real-time solution for tracking, implemented on GPUs, that is based on finding a minimum cost cyclic path in the product space spanned by the template shape and the given image. The cost function is derived from both the image data and the shape prior. They then extended their previous approach [9] by incorporating the edge information, which provides robustness against illumination changes. In another work [10], they also introduced a motion layer decomposition framework, which is solved using both discrete and continuous optimization. Although this method has been shown to be robust against occlusion, it is unclear how this algorithm may be adapted for tracking objects with unknown shape statistics. Discriminative methods [11, 12], which cast visual tracking as a two-class classification problem, dynamically update the classifiers in order to account for appearance changes and partial occlusion of the object-of-interest. Approaches such as [13] select the target as the one which minimizes the projection error in the space spanned by observed (tracked results from previous frames) and trivial templates (with one nonzero element).

Rich, dense flow information between consecutive frames in time sequence images can also be incorporated as a means to improve effectiveness. One way to include this information is to first compute a correspondence map (registration) between successive frames before computing the correlated segmentation of the video frames to solve the tracking problem. In recent works [14, 15], it has been shown that tracking performance can be further improved by solving the registration and the segmentation problems simultaneously (SRS for simultaneous registration and segmentation) thanks to a function that establishes the correspondence between the target and the reference level-set functions. The performance is improved in [16] with the help of a dynamical prior term (SRS+DP). The same authors recently generalized the SRS+DP framework (GSRS for generalized SRS), which can handle shading, reflections, and illumination changes. In [17], Ghosh *et al.* extend GSRS [18] for fast and efficient implementation. Modifications include the reconstruction of the sparse-error (due to partial occlusion, shading and reflections) between consecutive frames using a regularized variant of the L^1 norm, a new formulation of the dynamical shape prior term and the incorporation of a combination term, which modifies the update rule for the estimated functions. Through examples with natural and biological images sequences, the authors have illustrated the robustness of their formulation.

However, another fundamental problem in computer vision is the efficiency of the underlying strategy, in terms of CPU and memory requirement. A fast error-free tracking can be determining in some applications such as medical analysis or security systems. Medical analysis, with the ever increasing amount of data generated by modern acquisition systems, is in particular an area where large data sets need to be processed efficiently. The effectiveness of the approach of Ghosh *et al.* [17] is in large part due to the use of rich dense flow information between consecutive frames, a computationally expensive feature. However, dense flow information is mainly useful in the region near the evolving front so that the use of adaptive grid techniques that automatically coarsen away from the moving front have a distinct advantage. The key contribution of this paper is the use of Quad-/Oc-tree Cartesian grids for efficient processing of an algorithm that is robust against partial occlusion, drastic illumination changes, and data corruption.

The outline of this paper is as follows: we first briefly introduce the level-set formalism in section 2 and the SRS method with sparse error reconstruction in section 3. In section 4, we

develop the algorithm on Quad-/Oc-tree Cartesian grids. We finally compare the results obtained with the present method and that of [17] in section 5.

2 Level-Set Formalism

The level-set method, introduced by Osher and Sethian [19], is a general numerical technique for capturing the evolution of a moving front. Specifically, the front is represented as the zero level set of Lipschitz continuous function ϕ . We will denote the moving front as Γ and thus define $\Gamma = \{\mathbf{x} : \phi(\mathbf{x}) = 0\}$. The interior and exterior regions delimited by Γ are described by $\{\mathbf{x} : \phi(\mathbf{x}) > 0\}$ and $\{\mathbf{x} : \phi(\mathbf{x}) < 0\}$, respectively. The implicit level-set representation is particularly convenient in the case of topological changes. Given a velocity field \mathbf{u} , the motion of the front is given by the equation:

$$\phi_t + \mathbf{u} \cdot \nabla \phi = 0.$$

Finally, while any Lipschitz continuous function can define Γ , it is advantageous to use the signed distance function to the front. For a general front described by an arbitrary level-set function ϕ_{init} , the signed distance function is obtained by solving, in pseudo time τ , the equation:

$$\phi_\tau + \text{Sign}(\phi_{\text{init}})(|\nabla \phi| - 1) = 0,$$

where Sign refers to the signum function. This is the so-called reinitialization equation, introduced for uniform grids by Sussman *et al.* [20], improved by Russo and Smereka [21] and extended to unstructured grids in Min and Gibou [22].

3 Simultaneous Registration and Segmentation

Consider two consecutive images $I(t - 1) : \Omega(\subseteq \mathbb{R}^n) \rightarrow \mathbb{R}$, with $n = 2, 3$ and $I(t) : \Omega(\subseteq \mathbb{R}^n) \rightarrow \mathbb{R}$, in an image sequence at two consecutive time instances, $t - 1$ and t . Assuming that the initial contour of the object of interest is known *a priori*, we consider that the shape of the tracked object of interest at any arbitrary time t is embedded in the level-set function $\Phi^0(\mathbf{x}, t) : \Omega \rightarrow \mathbb{R}^n$.

3.1 Registration Model

The aim of registration is to find a displacement vector field $\mathbf{u} : \Omega \rightarrow \mathbb{R}^n$ that establishes the dense correspondence between two consecutive image frames. We denote the components of \mathbf{u} as $(u(\mathbf{x}), v(\mathbf{x}))$ and $(u(\mathbf{x}), v(\mathbf{x}), w(\mathbf{x}))$ in two and three spatial dimensions, respectively. This vector field is traditionally obtained by maximizing the posterior distribution $P(\mathbf{u}|I(t), I(t - 1))$ (see for example [1, 23] and references therein):

$$\begin{aligned} \mathbf{u}^* &= \underset{\mathbf{u}}{\text{argmax}} P(\mathbf{u}|I(t), I(t - 1)) \\ &\sim \underset{\mathbf{u}}{\text{argmax}} P(I(t), I(t - 1)|\mathbf{u}) P(\mathbf{u}). \end{aligned} \tag{1}$$

In the present work, based on [17], we simultaneously estimate both misalignment (dense correspondence map) and corruption in the image sequence by assuming that the corruption due to partial occlusion, shading, reflections, etc., is sparse in nature. Consider that the mapping $G : \Omega \rightarrow \mathbb{R}$ models the sparse corruptions between successive time frames. We can

rewrite Eq (1) by incorporating G as follows:

$$\begin{aligned}
 \langle \mathbf{u}^*, G^* \rangle &= \operatorname{argmax}_{\mathbf{u}, G} P(\mathbf{u}, G | I(t), I(t-1)) \\
 &\sim \operatorname{argmax}_{\mathbf{u}, G} P(I(t), I(t-1) | \mathbf{u}, G) P(\mathbf{u}, G) \\
 &= \operatorname{argmax}_{\mathbf{u}, G} P(I(t), I(t-1) | \mathbf{u}, G) P(\mathbf{u}) P(G),
 \end{aligned}
 \tag{2}$$

where we assumed in the last step that \mathbf{u} and G are independent of each other. Maximizing the expression in Eq (2) is equivalent to minimizing the following energy functional under certain simplifying assumptions:

$$\begin{aligned}
 \min_{\mathbf{u}, G} E_{RG}(\mathbf{u}, G; I(t-1), I(t)) &= \int_{\Omega} \rho_u (|I(\mathbf{T}(\mathbf{x}), t) - I(\mathbf{x}, t-1) - G|^2) d\mathbf{x} \\
 &+ \alpha_u \int_{\Omega} \rho_u (|\nabla u|^2 + |\nabla v|^2) d\mathbf{x} \\
 &+ \alpha_{g_1} \int_{\Omega} \rho_u (|G|^2) d\mathbf{x} + \alpha_{g_2} \int_{\Omega} \rho_u (|\nabla G|^2) d\mathbf{x},
 \end{aligned}
 \tag{3}$$

where, $\mathbf{T}(\mathbf{x}) = \mathbf{x} + \mathbf{u}(\mathbf{x})$, α_u controls the smoothness of the derived vector field, while α_{g_1} and α_{g_2} control the sparsity and the smoothness in the estimated corruption. It has been demonstrated in [3] that L^1 -minimization is more suitable than the quadratic penalizer, $\rho(x^2) = x^2$; hence we use $\rho_u = \sqrt{x^2 + \epsilon^2}$, the most robust convex penalizer known in the optical flow community. Due to the small positive constant ϵ , ρ_u is still convex, which offers advantages for the minimization process. The constant ϵ is chosen in our experiments to be 0.001.

Since the energy E_{RG} is highly nonlinear, the minimization process is not trivial. A minimizer for \mathbf{u} must satisfy the Euler-Lagrange equations:

$$\begin{aligned}
 C_u &= \rho'_u(I_d^2) I_d I_x - \alpha_u \nabla \cdot (\rho'_u (|\nabla u|^2 + |\nabla v|^2) \nabla u) = 0, \\
 C_v &= \rho'_u(I_d^2) I_d I_x - \alpha_v \nabla \cdot (\rho'_u (|\nabla u|^2 + |\nabla v|^2) \nabla v) = 0,
 \end{aligned}
 \tag{4}$$

where

$$\begin{aligned}
 I_d &= I(\mathbf{T}(\mathbf{x}), t) - I(\mathbf{x}, t-1) - G, \\
 I_x &= \partial_x I(\mathbf{T}(\mathbf{x}), t), \\
 I_y &= \partial_y I(\mathbf{T}(\mathbf{x}), t).
 \end{aligned}$$

Similarly, the Euler-Lagrange equation that needs to be satisfied by G is:

$$-\rho'_u(I_d^2) I_d I_x + \alpha_{g_1} \rho'_u (|G|^2) G - \alpha_{g_2} \nabla \cdot (\rho'_u (|\nabla G|^2) \nabla G) = 0.$$

To solve these equations, we adapted the approach from [24, 25]: \mathbf{u} and G are both solved iteratively in multiple scales using fixed point schemes. We implement a fully implicit scheme for the smoothing terms (e.g., the term associated with α_u and α_{g_2}) and a semi-implicit scheme for the data terms (e.g., the first term in Eq (3) and the term associated with α_{g_1}). The term corresponding to warping, $I(\mathbf{T}(\mathbf{x}), t)$, is obtained using linear interpolation.

3.2 Segmentation Model

It is standard to segment the t -frame using the dynamic prior term, given the observed contour $\phi^0(t)$. However, there are a few drawbacks with this approach. Examples include the absence of an energy minimizer or the dependence on the choice of the domain. These can be avoided with the formulation of Ghosh *et al.* [17], which is based on the dual formulation of the total variation (TV) norm, by including the shape prior term to define a new dynamic shape prior term, $d_p^2 : \Omega \rightarrow \mathbf{R}^+$:

$$d_p^2(\phi^o(\mathbf{x}); \hat{\phi}^-(\mathbf{x})) = \int_{\Omega} \rho_p(|\hat{\phi}^-(\mathbf{x})|^2) |\nabla H(\phi^o)| d\mathbf{x}, \tag{5}$$

where $\rho_p(\cdot)$ is a robust estimator; here we assume $\rho_p(y^2) = y^2$. One then considers the following constrained minimization problem:

$$\min_{0 \leq \phi^0 \leq 1} E_{DS}(\phi^0(\mathbf{x}); I(t), d_p^2) = \underbrace{\int_{\Omega} d_p^2 |\nabla \phi^0(\mathbf{x})| d\mathbf{x}}_{TV_{d_p^2}(\phi^0)} + \alpha_{s_1} \underbrace{\int_{\Omega} \ln \left(\frac{p(I|\theta_2)}{p(I|\theta_1)} \right)}_{\eta(\mathbf{x}, \theta_1, \theta_2)} \phi^0(\mathbf{x}) d\mathbf{x}. \tag{6}$$

The solution is constrained to lie between 0 and 1 since the unconstrained problem does not have a stationary solution. The first term in Eq (6) measures the weighted total variation norm of the function ϕ^0 . It can be rigorously demonstrated (see [26, 27] and references therein) that Eq (6) is equivalent to solving the following unconstrained problem:

$$\min_{0 \leq \phi^0 \leq 1} E_{DS}(\phi^0(\mathbf{x}); I(t), d_p^2) = TV_{d_p^2}(\phi^0) + \int_{\Omega} [\alpha_{s_1} \eta(\mathbf{x}, \theta_1, \theta_2) \phi^0 + \hat{\alpha} \zeta(\phi^0)] d\mathbf{x}, \tag{7}$$

where $\zeta(y) := \max \{0, 2|y - \frac{1}{2}| - 1\}$ and $\hat{\alpha} > \frac{\alpha_{s_1}}{2} \|\eta(\mathbf{x})\|_{L^\infty(\Omega)}$. We can also rewrite $\zeta(y)$ as:

$$\zeta(y) = \begin{cases} -2y & \text{if } y < 0, \\ 2(y - 1) & \text{if } y > 1, \\ 0 & \text{otherwise.} \end{cases}$$

Since the two energies in Eqs (6) and (7) agree for $\{\phi^0 \in L^\infty(\Omega) : 0 \leq \phi^0(\mathbf{x}) \leq 1, \forall \mathbf{x}\}$ it is sufficient to prove that any minimizer of Eq (7) satisfies the constraint $0 \leq \phi^0(\mathbf{x}) \leq 1$. The variational formulation (7) can now be solved using a convex regularization [28] term:

$$\begin{aligned} \min_{\phi^o, \varphi} E_{DS}(\phi^o, \varphi; I(t), d_p^2) &= TV_{d_p^2}(\phi^o) + \frac{1}{2\beta} \|\phi^o - \varphi\|^2 \\ &+ \int_{\Omega} [\alpha_{s_1} \eta(\mathbf{x}, \theta_1, \theta_2) \varphi + \alpha \zeta(\varphi)] d\mathbf{x}, \end{aligned} \tag{8}$$

where $\beta > 0$ is a small constant, so that φ is a close approximation of ϕ^o .

3.3 Simultaneous Registration and Segmentation

A commonly used objective function in this regard is the one which establishes the correspondence between the target ($\phi^o(t)$) and the reference ($\phi^o(t - 1)$) level-set functions using a mapping function $\mathbf{u}(\mathbf{x}, t - 1, t)$. The functional combination term seeks to register the isocontours of the source and target shapes within a narrow band of the zero level sets. The width of the

narrow band is chosen by a parameter ϵ' . The choice of ϵ' is crucial since it determines the scale at which objects are registered (see [29] for a detailed discussion). In contrast, Ghosh *et al.* [17] proposed a functional that does not require the scale information *a priori*. In this case, the combination term is (similar to Eq (5)) defined as:

$$\min_{\mathbf{u}, \phi^o(t)} E_{NC}(\mathbf{u}, \phi^o(t); \phi^o(t-1)) = d_c^2 = \int_{\Omega} \rho_c(|\phi^o(\mathbf{T}(\mathbf{x}), t-1)|^2) |\nabla H(\phi^o(\mathbf{x}, t))| d\mathbf{x},$$

where $\rho_c(y^2) = y^2$. Finally, the method simultaneously estimates \mathbf{u} , G , and $\phi^o(t)$ for the current frame at time t . This combination term modifies the update rule for the estimated functions. Recalling Eq (4), the new Euler-Lagrange equations that need to be satisfied for \mathbf{u} can be written as:

$$\rho'_c(\phi_\omega^2) \phi_\omega \phi_x |\nabla H(\phi^o(t))| + C_u = 0,$$

$$\rho'_c(\phi_\omega^2) \phi_\omega \phi_y |\nabla H(\phi^o(t))| + C_w = 0,$$

where

$$\phi_\omega = \phi^o(\mathbf{T}(\mathbf{x}), t-1), \quad \phi_x = \partial_x \phi_\omega \quad \text{and} \quad \phi_y = \partial_y \phi_\omega.$$

The update rule for the sparse error, $G(x, y)$, remains the same since it is independent of the combination term E_{NC} . For the segmentation problem, we use:

$$\min_{\phi^0} \{TV_{d_p^2+d_c^2}(\phi^0) + \frac{1}{2\beta} \|\phi^0 - \varphi\|^2\}. \tag{9}$$

This formulation has been shown to be robust against partial occlusion, drastic illumination changes, and data corruption. A drawback of the existing approach is the computational cost associated with solving the partial differential equations of the model on a grid with as many degree of freedom as the number of pixels/voxels. However, the region of interest for processing the data at every step is located in only a small band around the moving front. It is therefore desirable to adopt a strategy that can leverage the locality of information. Next, we describe an algorithm based on Quad-/Oc-tree data structure, which enables to significantly decrease the number of degrees of freedom, while conserving the accuracy of the method of [17].

4 Algorithm on Quad/Oc-tree Data Structures

In this section, we describe how the level-set method on Quad-/Oc-tree introduced by Min and Gibou [22] can be combined with the methodology of [17] to produce an efficient algorithm for object tracking and segmentation.

4.1 Quad/Oc-tree Data Structures

Quadtree (resp. Octree) data structures can be used to represent the spatial discretization of a geometrical domain in two (resp. three) spatial dimensions as depicted in Fig 1: initially the root of the tree is associated with the entire domain cell. Further refinement is obtained by recursively splitting each cell into four (resp. eight) children until the desired level of detail is achieved. We call ‘level of the tree’ its depth. We refer the reader to the books of Samet [30, 31] for more details on the definition of these data structures and the standard algorithms associated with them.

In this work, the maximum level of the tree is that corresponding to the pixel size, i.e. a 1024×1024 image will be associated with a quadtree with a maximum level of 10 (as in 2^{10}).

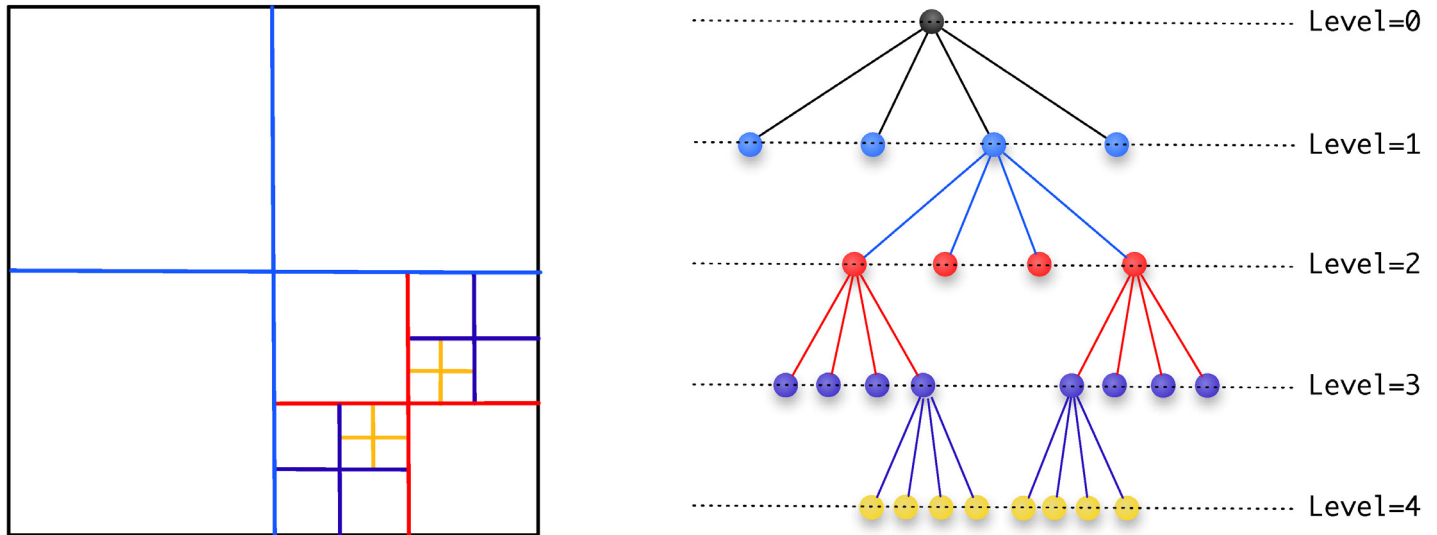


Fig 1. Discretization of a two-dimensional domain (left) and its quadtree representation (right). The entire domain corresponds to the root of the tree (level 0). Each cell can then be recursively subdivided further into four children. In this example, the tree is non-graded, since the difference of level between some adjacent cells exceeds one. (Color online.)

doi:10.1371/journal.pone.0150889.g001

Also, in the case where a tree cell is associated to a subset of an image, the cell’s value is defined as the average pixels’ value.

4.2 Constructing Quad-/Oc-tree

Since the accuracy of the method depends on the size of the cells adjacent to the moving front, we impose that the finest cells lie on the interface. This can be achieved by using the signed distance function to the interface along with the Whitney decomposition, as first proposed by Strain in [32]. For a general function $\phi : \mathbb{R}^n \rightarrow \mathbb{R}$ with Lipschitz constant $Lip(\phi)$, the Whitney decomposition is extended in Min [22] as: starting from a root cell, split any cell C if

$$\max_{v \in Nodes(C)} |\phi(v)| \leq \frac{1}{2} Lip(\phi) \times \text{diag-size}(C),$$

where $\text{diag-size}(C)$ refers to the length of the diagonal of the current cell C and $Nodes(C)$ refers to the set of four (resp. eight) nodes of the current cell. This condition enforces the splitting of any cell whose edge length exceeds its distance to the interface, hence producing a computational grid where the smallest cells capture the interface location. Larger and larger cells are located as the distance from the interface increases (see [22] for details).

Fig 2 depicts an example of an image (left) and its quadtree representation (right). Note that far from the simulated moving front (depicted by a red interface), the image is blurred due to averaging of pixels’ value, while the original image resolution is kept near the interface. In the work of [16], the image was filtered by a gaussian filter to avoid spurious noise. In the case of Quad-/Oc-tree representation, the coarsening away from the front acts as a filter so we do not apply a Gaussian filter.

4.3 Finite Difference Discretizations

In the case of nonregular Cartesian grids, the main difficulty is to derive discretizations at T-junction nodes, i.e. nodes for which there is a missing neighboring node in one of the Cartesian

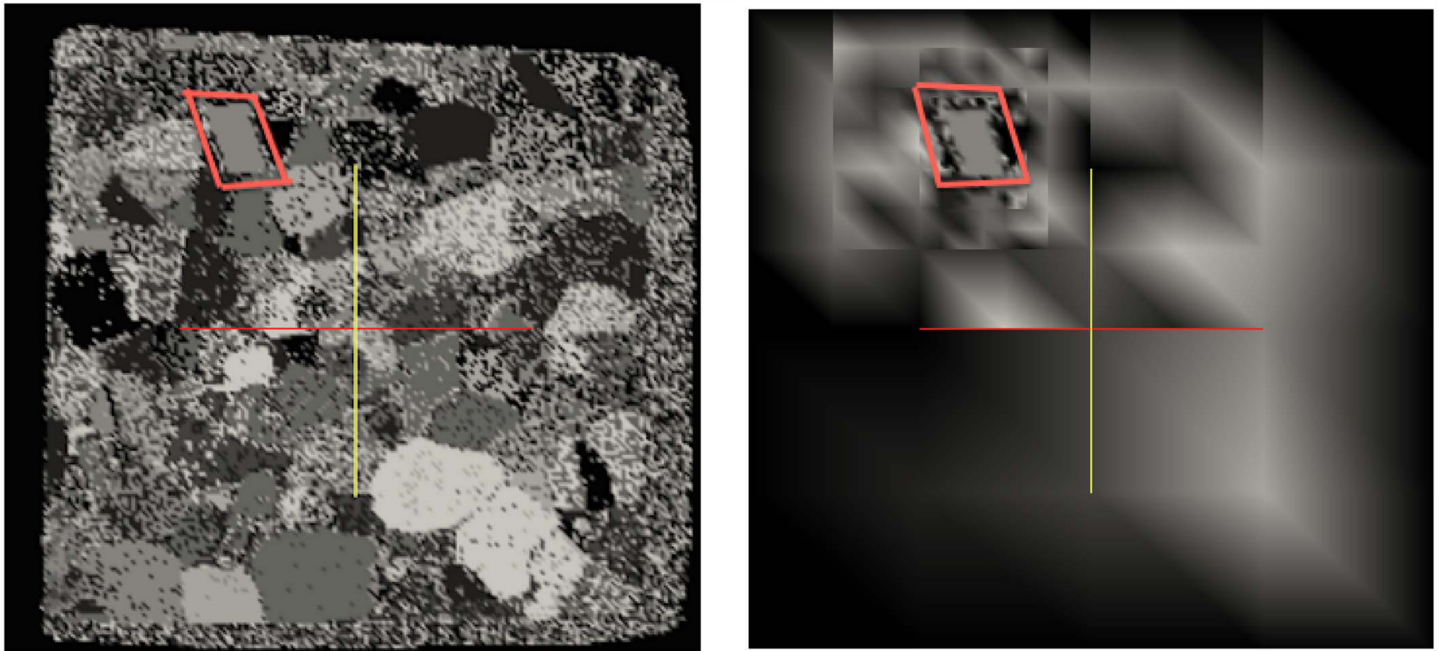


Fig 2. Quadtree refinement of an image.

doi:10.1371/journal.pone.0150889.g002

directions. For example, Fig 3 depicts a T-junction node, v_0 , with three neighboring nodes v_1 , v_2 and v_3 aligned in three Cartesian directions and one ghost neighboring node, v_g , replacing the missing grid node in the x -direction. Referring to Fig 3, the value, u_g^G , of the node-sampled function $u : \{v_i\} \rightarrow \mathbb{R}$ at the ghost node v_g is given by the third-order interpolation:

$$u_g^G = \frac{u_3 s_4 + u_4 s_3}{s_3 + s_4} - \frac{s_3 s_4}{s_1 + s_2} \left(\frac{u_1 - u_0}{s_1} + \frac{u_2 - u_0}{s_2} \right).$$

In three spatial dimensions, there are two ghost values (see Fig 4) that are defined as [22]:

$$\begin{aligned} u_6^G &= \frac{s_8 u_7 + s_7 u_8}{s_8 + s_7} - \frac{s_8 s_7}{s_2 + s_1} \left(\frac{u_2 - u_0}{s_2} + \frac{u_1 - u_0}{s_1} \right), \\ u_4^G &= \frac{s_{11} s_{12} u_{11} + s_{11} s_9 u_{12} + s_{10} s_{12} u_9 + s_{10} s_9 u_{10}}{(s_{10} + s_{11})(s_9 + s_{12})} \\ &\quad - \frac{s_{10} s_{11}}{s_2 + s_1} \left(\frac{u_2 - u_0}{s_2} + \frac{u_1 - u_0}{s_1} \right) \\ &\quad - \frac{s_9 s_{12}}{s_1 + s_4} \left(\frac{u_1 - u_0}{s_1} + \frac{u_6^G - u_0}{s_4} \right). \end{aligned}$$

The third-order interpolations defined above allow us to define finite difference formulas for the first- and second-order derivatives at every node using standard formulas in a dimension-by-dimension framework. For example, referring to Fig 5, we use the central difference formulas for u_x and u_{xx} :

$$D_x^0 u_0 = \frac{u_2 - u_0}{s_2} \cdot \frac{s_1}{s_1 + s_2} + \frac{u_0 - u_1}{s_1} \cdot \frac{s_2}{s_1 + s_2},$$

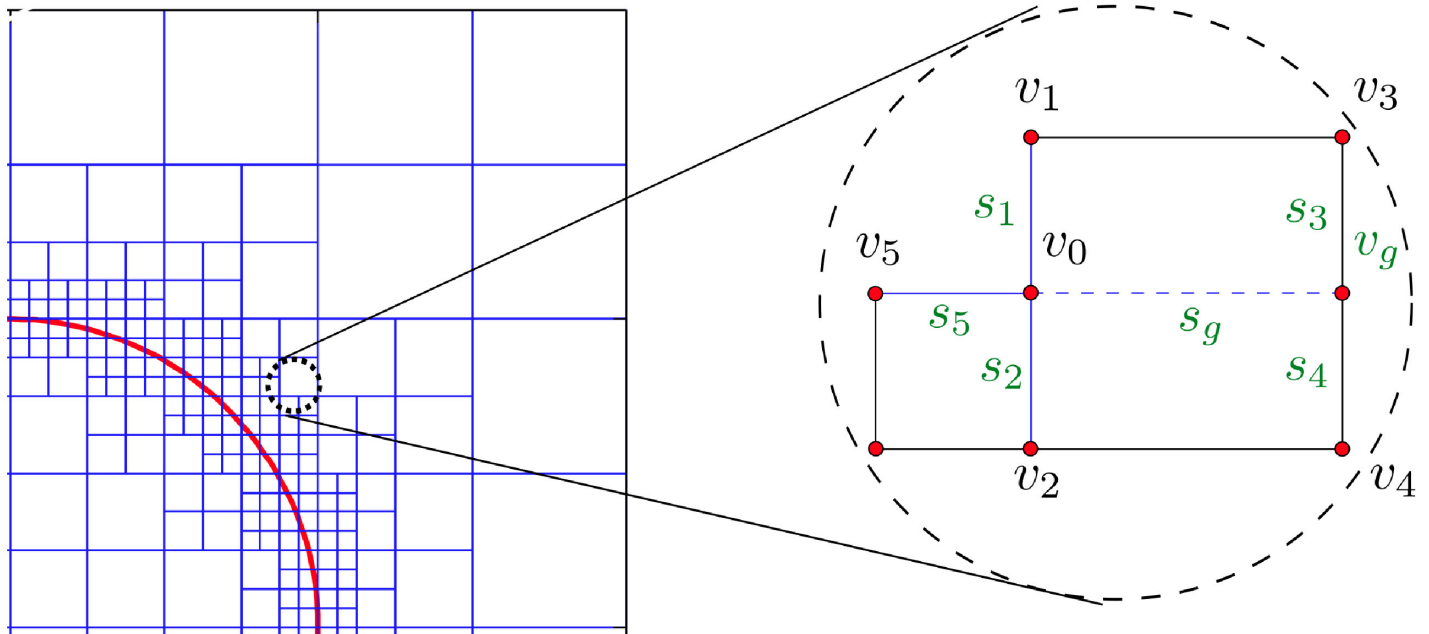


Fig 3. Local grid configuration near a node v_0 . The schematic on the right describes a T-junction where a node is missing in the x-direction.

doi:10.1371/journal.pone.0150889.g003

$$D_{xx}^0 u_0 = \frac{u_2 - u_0}{s_2} \cdot \frac{2}{s_1 + s_2} - \frac{u_0 - u_1}{s_1} \cdot \frac{2}{s_1 + s_2},$$

the forward and backward first-order accurate approximations of the first-order derivatives:

$$D_x^+ u_0 = \frac{u_2 - u_0}{s_2},$$

$$D_x^- u_0 = \frac{u_0 - u_1}{s_1},$$

and the second-order accurate approximations of the first-order derivatives:

$$D_x^+ u_0 = \frac{u_2 - u_0}{s_2} - \frac{s_2}{2} \min\text{mod}(D_{xx}^0 u_0, D_{xx}^0 u_2),$$

$$D_x^- u_0 = \frac{u_0 - u_1}{s_1} + \frac{s_1}{2} \min\text{mod}(D_{xx}^0 u_0, D_{xx}^0 u_1),$$

where the minmod slope limiter [33, 34], defined as:

$$\min\text{mod}(x, y) = \begin{cases} x & \text{if } |x| > |y|, \\ y & \text{otherwise,} \end{cases}$$

is used to avoid differencing across regions where gradients are large (i.e. near kinks). Similarly, approximations for first-order and second-order derivatives are obtained in the y- and z- directions. These numerical approximations are sufficient to discretize all of the equations described in section 3.

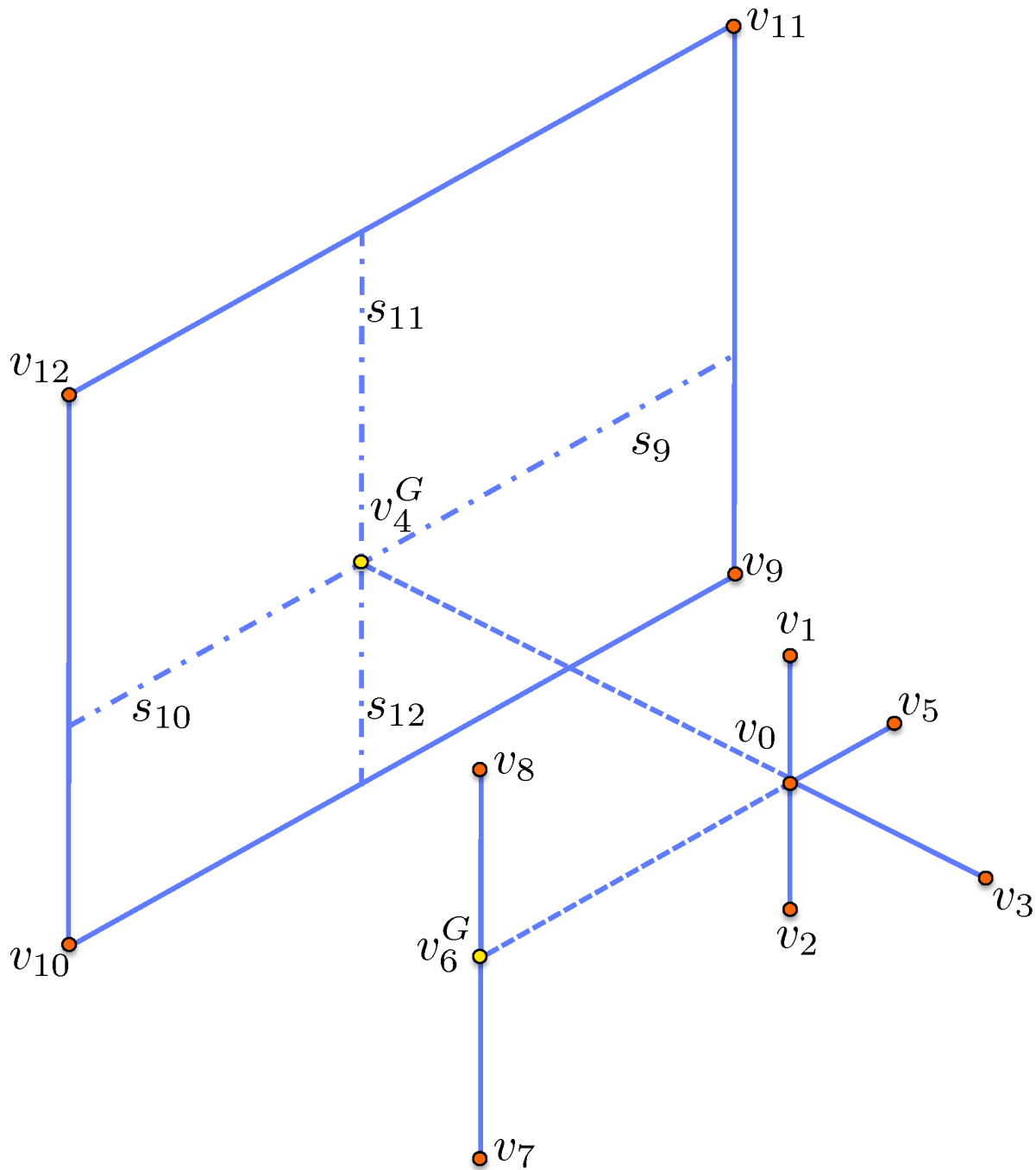


Fig 4. Neighboring vertices of a node v_0 in three spatial dimensions.

doi:10.1371/journal.pone.0150889.g004

5 Experimental Results

In this section, we first present results on two and three dimensional image sequences. When available, we compare our results to those obtained in Ghosh *et al.* [16], which allows us to compare the effectiveness of both methods in term of robustness as well as in terms of computational efficiency. These numerical examples illustrate that the present method is robust even in the presence of severe occlusions and provides a speed up over the approach of Ghosh *et al.* [16].

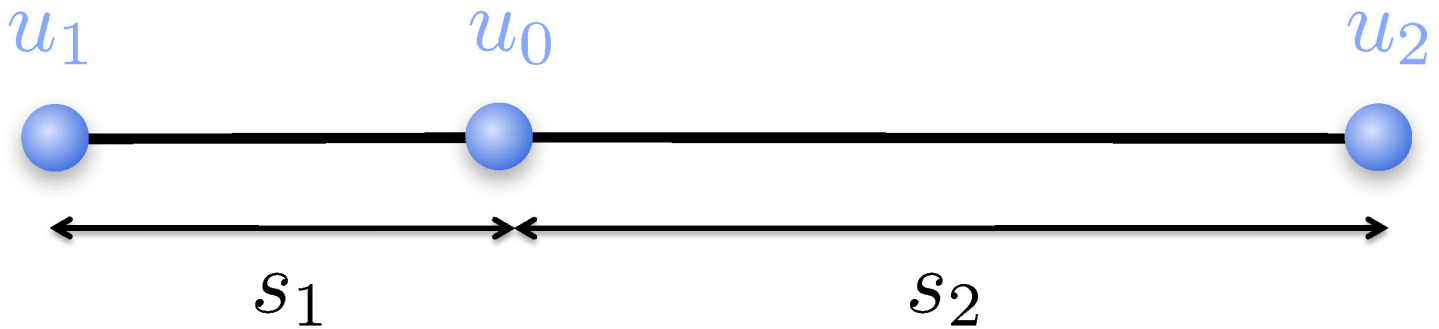


Fig 5. A one-dimensional adaptive grid.

doi:10.1371/journal.pone.0150889.g005

The advantage of the present method is that it significantly reduces the number of degrees of freedom processed at each iteration. In fact, since most degrees of freedom are those adjacent to the interface, this approach lowers the dimension of the problem. On the other hand, there are additional costs associated to the handling of the data structure, especially the $O(\log(n))$ cost to access data stored in the leaves of the tree and the remeshing procedure. Despite this additional cost, the performance of the current approach is superior than that on uniform grids.

5.1 Parameter Setting

We keep all the parameters constant throughout our experiments. The parameters α_u , α_{g_1} , and α_{g_2} are empirically set to 8×10^{-4} , 9×10^{-5} , and 3×10^{-3} , respectively. The weight for the dynamical prior term d_p^2 and the combination term d_c^2 are both approximately set to 2×10^{-5} . The coefficient α_{s_1} corresponding to Eq (8) is selected randomly in the range [0.02, 0.05].

We first extract a simple gray scale histogram (for both natural and synthetic images) of the foreground and background from a few frames. These are then used to obtain the likelihood values $\eta(x)$ for the rest of the frames. In the current implementation, we also use the edge information of the image. The edge information is easily incorporated by minimizing $TV_{g+d_p^2+d_c^2}$ instead of $TV_{d_p^2+d_c^2}$ in Eq (9). Here, $g = \frac{1}{1+\gamma|\nabla I_\sigma|^2}$ is the inverse of the smoothed gradient magnitude function. In our experiments, we keep $\gamma = 600$.

The proposed tracking framework is implemented in C++ on a 2.3 GHz Intel Core i5 CPU. In our implementation we update \mathbf{u} , ϕ^0 , and G twice for each time frame.

5.2 Two Dimensional Data

We consider a medical data sequence where the objects of interest are vessels. In the first slice, the objects of interest are four cross-sections of vessels, which merge into two larger vessels and then into one large vessel in subsequent slices. This example illustrates the error-free tracking effectiveness. Fig 6 compares the results obtained with the approach of Ghosh *et al.* [16] (top subfigures) with the results of the current approach (bottom subfigures). These results demonstrate that the four initial vessels are indeed properly detected and that the algorithm enables changes in topology (here merging). Fig 7 depicts the three-dimensional segmentation obtained from the series of two-dimensional segmentations.

The use of the adaptive grid significantly reduces the number of pixels processed. Specifically, the uniform grid contains $201 \times 201 = 40,401$ cells, whereas the adaptive grid processes about 1000 cells at each iteration. Therefore, even though the adaptive approach requires the data structure’s management, the remeshing at each step and the $O(\log(n))$ complexity for

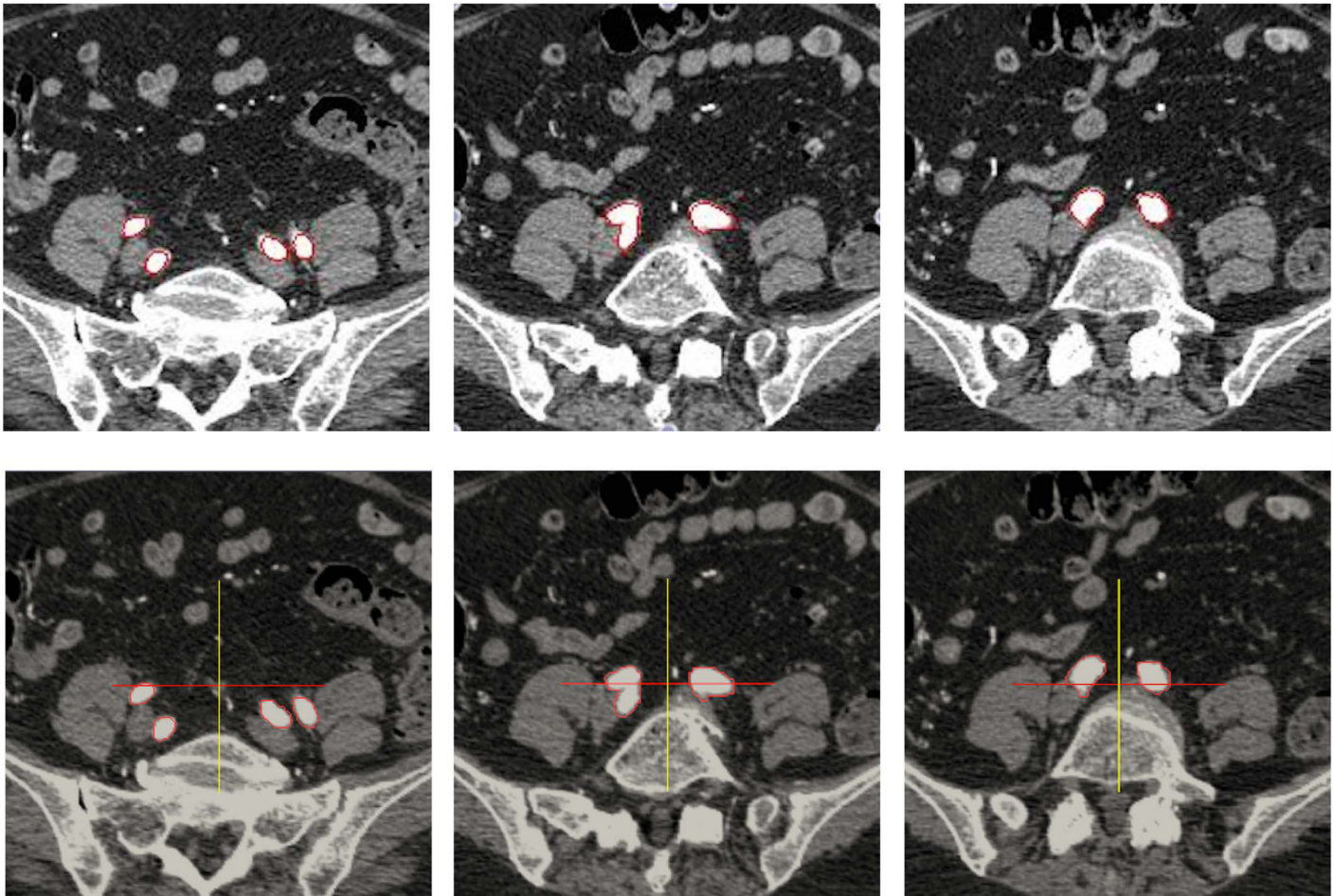


Fig 6. Medical data processed on uniform grids (top row) and on adaptive grids (bottom row) illustrating the similarity in results. In both cases, 268 frames with size 202×201 are processed. Timing is 25 minutes (1500 seconds) in the case of uniform grids and about 3.5 minutes (216 seconds) in the case of the current approach on adaptive grids.

doi:10.1371/journal.pone.0150889.g006

accessing data (n being the number of cells processed), the adaptive approach increases the speed in a meaningful way: on this example, the process time is more than 25 minutes on the uniformly sampled image, compared to less than 5 minutes with the same parameters and the same number of frames in the case of adaptive grids. We note that no attempts has been made to optimize the implementation, for which, in average, the overhead time for considering the adaptive grid is about 0.03 second per slice. This amounts to 7.71 seconds over the course of the total segmentation that takes a total computational time of 201.66 seconds. For this example, the overhead for using adaptive grids is thus about 4%.

5.3 Three Dimensional Data

We consider three different sequences of three dimensional medical data consisting of snapshots of a beating heart, available from the BIRN data base [35]. Each frame has dimensions $100 \times 100 \times 34$. The sequences, corresponding to patient 12, 13 and 15 consist of 20, 21 and 12 frames, respectively.

Figs 8, 9 and 10 depict the results of the segmentation process using the current approach. In each case, three different cross sections (corresponding to a top, medium and bottom cross-

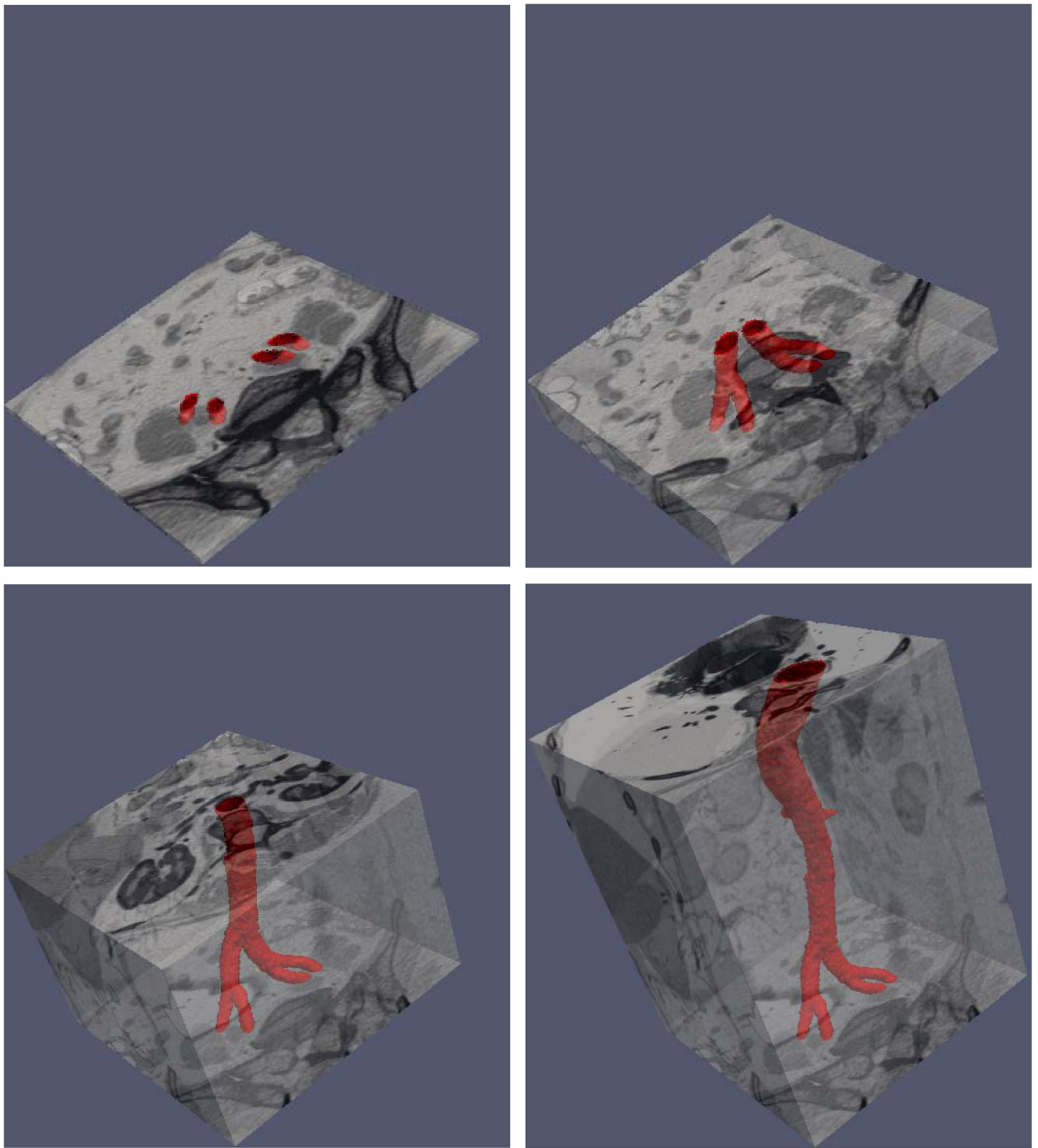


Fig 7. Three dimensional representation of the segmentation of Fig 6, which illustrates the changes in topology.

doi:10.1371/journal.pone.0150889.g007

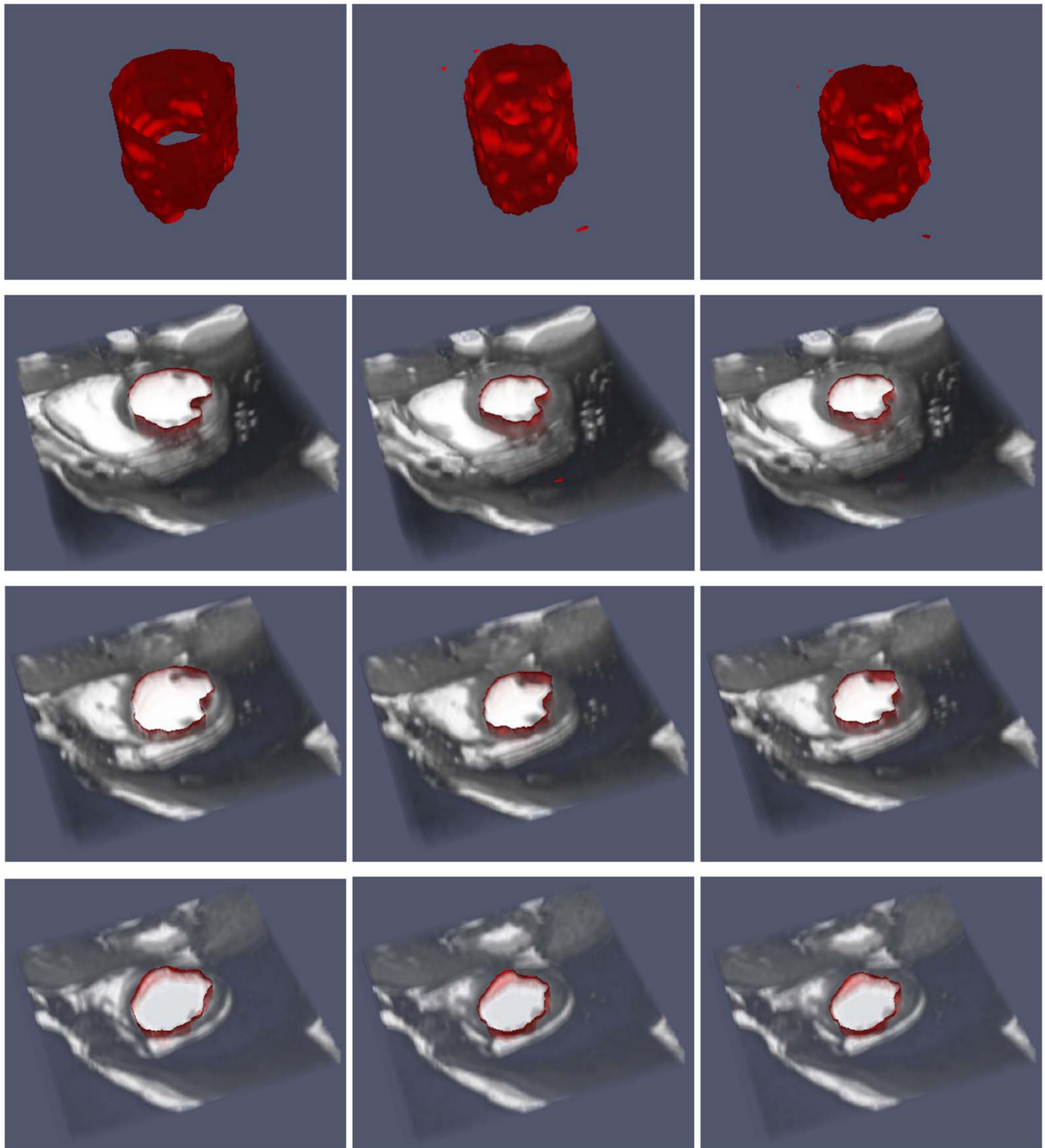


Fig 8. Image sequence for a half period ($T/2$) for patient 12. The top row gives the segmentation in 3D, while the remaining rows give the top, middle and bottom cross-sections of the segmentation on top of the data for $t = 0$, $t = T/4$ and $t = T/2$, respectively.

doi:10.1371/journal.pone.0150889.g008

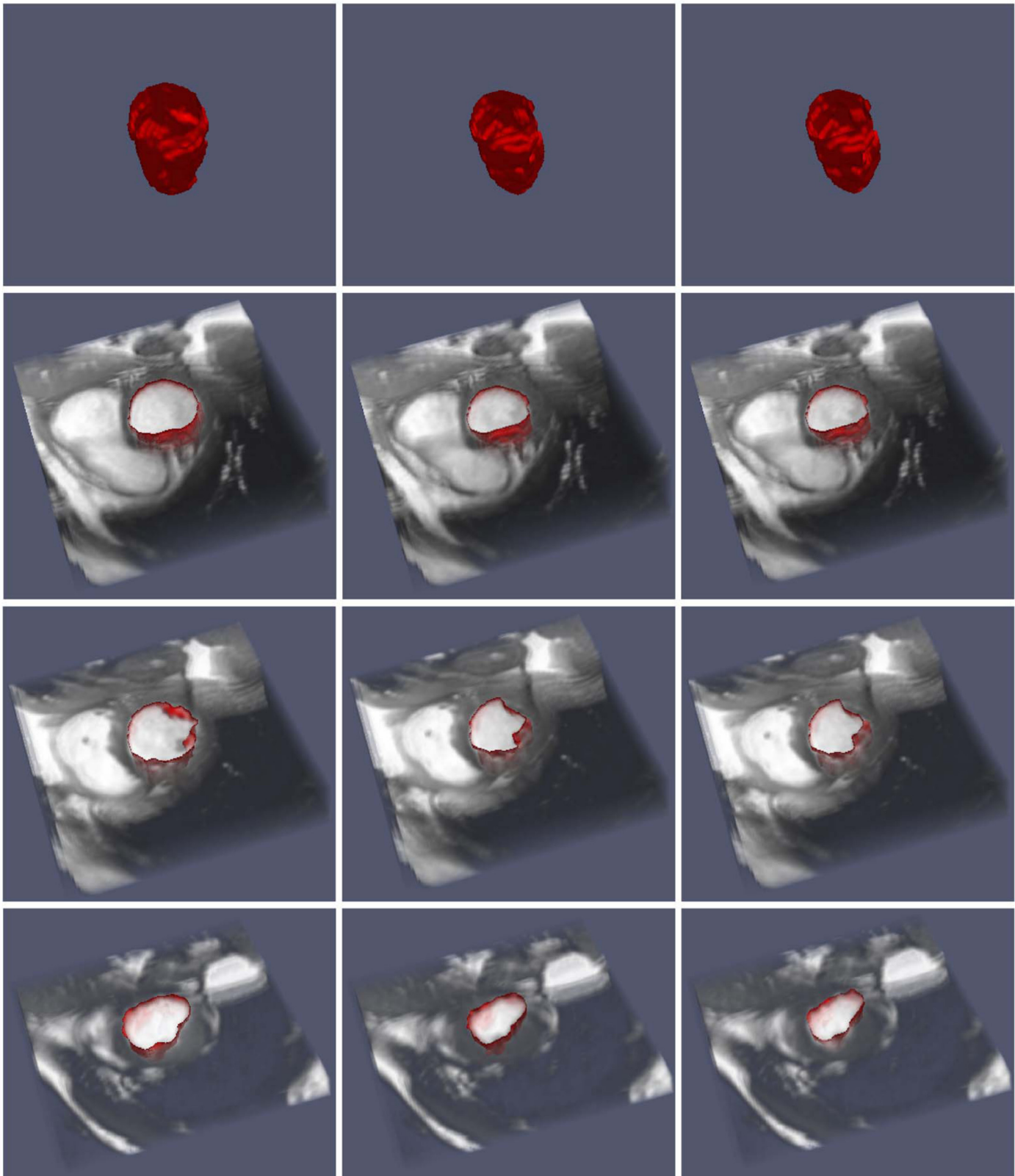


Fig 9. Image sequence for a half period (T/2) for patient 13. The top row gives the segmentation in 3D, while the remaining rows give cross-sections of the segmentation on top of the data.

doi:10.1371/journal.pone.0150889.g009



Fig 10. Image sequence for a half period ($T/2$) for patient 15. The top row gives the segmentation in 3D, while the remaining rows give cross-sections of the segmentation on top of the data.

doi:10.1371/journal.pone.0150889.g010

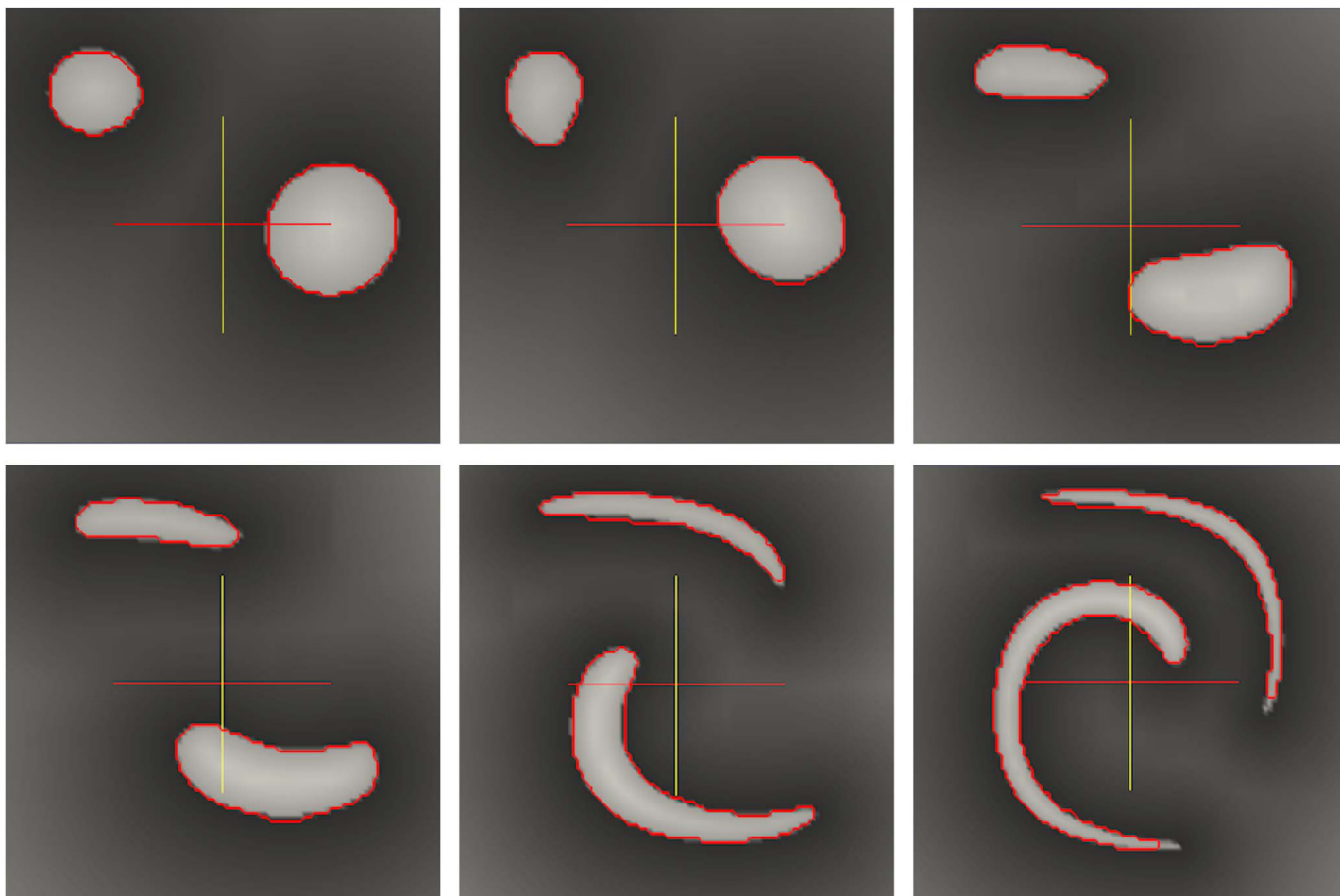


Fig 11. Two vortex in 2D: 500 frames with size 100 × 100, performed in 637.65 seconds.

doi:10.1371/journal.pone.0150889.g011

section) are depicted at three different time in the image sequence. In all cases, the segmentation results are satisfactory. We note that since the algorithm requires an initial segmentation, we used one 2D cross-section of the first frame and applied the 2D algorithm to construct the initial 3D segmentation. Therefore, overall, only an initial 2D segmentation is needed to process the time sequence of three dimensional data. The algorithm takes about 55 seconds to process one 3D frame with the Octree representation and about 560 seconds for the uniform one.

5.4 Large Deformation and Disconnected Sets

We consider a synthetic example to demonstrate that the method is applicable to the case where the shape of interest undergoes large deformations. It also demonstrates that disconnected sets can be readily considered. We first built an image sequence consisting of two disks deforming under a given velocity field: in a domain $\Omega = [0, 1]^2$, we construct a disk of radius.15 and center (.5,.75) and a second disk with a radius of.1 and center (.2,.2). These two disks define the initial zero level set contour. The image intensity values are then randomly set to be between [200, 255] inside the zero level set with a value of 200 near the level set contour and between [50, 155] outside the zero level-set with a value of 50 near the contour. The image is

then deformed under the divergence free velocity field $\mathbf{u} = (u, v)$ given by:

$$\begin{aligned} u(x, y) &= -\sin^2(\pi x)\sin(2\pi y), \\ v(x, y) &= \sin^2(\pi y)\sin(2\pi x), \end{aligned}$$

and the algorithm on quadtree is used to segment the image sequence. Fig 11 illustrates that the algorithm on adaptive grids enables the segmentation of a disconnected shape that undergoes large deformations. The computational time on this 100×100 image is slightly more than a second per frame.

6 Conclusion

We have extended the approach of [17] to the case of adaptive Cartesian grids using Quad-/Oc-tree data structures. This approach is based on a robust variational framework for level-set-based simultaneous registration and segmentation procedures that can be applied to natural and biological image sequences. In addition, the use of adaptive grids produces results that can be processed much faster than [17] while retaining its accuracy and robustness. Using numerical results on image sequences in two and three spatial dimensions as well as timings, we have illustrated the computational efficiency of the approach. While we have presented the use of adaptive grids using a particular functional, the advantages of adaptive grids (reduction in the number of processed data and the ability to focus on processing data near the moving front) could be used with other functionals used in level-set formulations. The strategy presented in this paper could be used to the segmentation of large data sets, such as those generated by new generation CT acquisition scanners or by structural MRI scans.

Acknowledgments

This research was supported by NSF CHE 1027817.

Author Contributions

Conceived and designed the experiments: MP PG FG. Performed the experiments: MP. Analyzed the data: MP PG FG. Contributed reagents/materials/analysis tools: MP PG FG. Wrote the paper: MP PG FG.

References

1. Black MJ, Anandan P. The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. *Computer Vision and Image Understanding*. 1996; 63(1):75–104. doi: [10.1006/cviu.1996.0006](https://doi.org/10.1006/cviu.1996.0006)
2. Cremers D, Soatto S. A pseudo-distance for shape priors in level set segmentation. In: 2nd IEEE workshop on variational, geometric and level set methods in computer vision; 2003. p. 169–176.
3. Wedel A, Pock T, Zach C, Bischof H, Cremers D. An improved algorithm for TV-L 1 optical flow. *Statistical and Geometrical Approaches to Visual Motion Analysis*. 2009;p. 23–45.
4. Mitchell SC, Lelieveldt BP, der Geest RJV, Bosch HG. Multistage hybrid active appearance model matching: segmentation of left and right ventricles in cardiac MR images. *Medical Imaging, IEEE Transactions on*. 2001 May; 20(5):415–423. doi: [10.1109/42.925294](https://doi.org/10.1109/42.925294)
5. Tsai A, Yezzi A, Wells W, Tempny C, Tucker D, Fan A, et al. A shape-based approach to the segmentation of medical imagery using level sets. *Medical Imaging, IEEE Trans on*. 2003 Feb; 22(2):137–154. doi: [10.1109/TMI.2002.808355](https://doi.org/10.1109/TMI.2002.808355)
6. Zhu Y, Papademetris X, Sinusas A, Duncan JS. Segmentation of left ventricle from 3D cardiac MR image sequences using a subject-specific dynamical model; 2008. p. 1–8.
7. Ishikawa H, Jermyn IH. Region extraction from multiple images. In: *Computer Vision, IEEE Int. Conference on*; 2001. p. 509–516.

8. Schoenemann T, Cremers D. Globally Optimal Shape-based Tracking in Real-time. In: IEEE Int. Conference on CVPR. Anchorage, Alaska; 2008. p. 1–6.
9. Schoenemann T, Cremers D. A Combinatorial Solution for Model-based Image Segmentation and Real-time Tracking. IEEE Trans PAMI. 2009;p. 1153–1164.
10. Schoenemann T, Cremers D. High resolution motion layer decomposition using dual-space graph cuts. In: Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on; 2008. p. 1–7.
11. Avidan S. Ensemble tracking. In: IEEE Int. Conference on Computer Vision and Pattern Recognition; 2005. p. 261–271.
12. Babenko B, Yang MH, Belongie S. Visual tracking with online multiple instance learning. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2010; 33:1619–1632.
13. Mei X, Ling H. Robust visual tracking using L^1 minimization. In: Proc. of Int. Conf. on Computer Vision; 2009. p. 1–8.
14. An J, Chen Y, Huang F, Wilson D, Geiser E. A variational PDE based level set method for a simultaneous segmentation and non-rigid registration. LNCS. 2005; 3749:286–293.
15. Unal G, Slabaugh G. Coupled PDEs for non-rigid registration and segmentation. CVPR, IEEE Int Conference on. 2005 June;1:168–175.
16. Ghosh P, Sargin ME, Manjunath BS. Robust Dynamical Model for Simultaneous Registration and Segmentation in a Variational Framework: A Bayesian Approach. In: Computer Vision, IEEE Int. Conference on; 2009. p. 709–716.
17. Ghosh P, Manjunath BS. Robust Simultaneous Registration and Segmentation with Sparse Error Reconstruction. IEEE Trans on Pattern Analysis and Machine Intelligence. 2013; 35(2):425–436. doi: [10.1109/TPAMI.2012.103](https://doi.org/10.1109/TPAMI.2012.103)
18. Ghosh P, Sargin ME, Manjunath BS. Generalized Simultaneous Registration and Segmentation. In: IEEE Int. Conference on Computer Vision; 2010. p. 1363–1370.
19. Osher S, Sethian JA. Fronts Propagating with Curvature-Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulations. Journal of Computational Physics. 1988; 79:12–49. doi: [10.1016/0021-9991\(88\)90002-2](https://doi.org/10.1016/0021-9991(88)90002-2)
20. Sussman M, Smereka P, Osher S. A Level Set Approach for Computing Solutions to Incompressible Two-Phase Flow. J Comput Phys. 1994; 114:146–159. doi: [10.1006/jcph.1994.1155](https://doi.org/10.1006/jcph.1994.1155)
21. Russo G, Smereka P. A Remark on Computing Distance Functions. J Comput Phys. 2000; 163:51–67. doi: [10.1006/jcph.2000.6553](https://doi.org/10.1006/jcph.2000.6553)
22. Min C, Gibou F. A second-order-accurate level set method on non-graded adaptive Cartesian grids. J Comput Phys. 2007; 225:300–321. doi: [10.1016/j.jcp.2006.11.034](https://doi.org/10.1016/j.jcp.2006.11.034)
23. Cremers D, Soatto S. Motion competition: A variational approach to piecewise parametric motion segmentation. International Journal of Computer Vision. 2005; 62(3):249–265. doi: [10.1007/s11263-005-4882-4](https://doi.org/10.1007/s11263-005-4882-4)
24. Brox T, Bruhn A, Papenberg N, Weickert J. High accuracy optical flow estimation based on a theory for warping. LNCS. 2004;p. 25–36.
25. Papenberg N, Bruhn A, Brox T, Didas S, Weickert J. Highly accurate optic flow computation with theoretically justified warping. International Journal of Computer Vision. 2006; 67(2):141–158. doi: [10.1007/s11263-005-3960-y](https://doi.org/10.1007/s11263-005-3960-y)
26. Chan TF, Esedoglu S, Nikolova M. Algorithms for finding global minimizers of image segmentation and denoising models. Algorithms. 2004; 66(5):1632–1648.
27. Bresson X, Esedoglu S, Vanderghyest P, Thiran JP, Osher S. Fast global minimization of the active contour/snake model. Journal of Mathematical Imaging and Vision. 2007; 28(2):151–167. doi: [10.1007/s10851-007-0002-0](https://doi.org/10.1007/s10851-007-0002-0)
28. Aujol JF, Gilboa G, Chan T, Osher S. Structure-texture image decomposition—modeling, algorithms, and parameter selection. International Journal of Computer Vision. 2006; 67(1):111–136. doi: [10.1007/s11263-006-4331-z](https://doi.org/10.1007/s11263-006-4331-z)
29. Taron M, Paragios N, Jolly MP. Registration with uncertainties and statistical modeling of shapes with variable metric kernels. Pattern Analysis and Machine Intelligence, IEEE Transactions on. 2008; 31(1):99–113. doi: [10.1109/TPAMI.2008.36](https://doi.org/10.1109/TPAMI.2008.36)
30. Samet H. The Design and Analysis of Spatial Data Structures. New York: Addison-Wesley; 1989.
31. Samet H. Applications of Spatial Data Structures: Computer Graphics, Image Processing and GIS. Addison-Wesley; 1990.
32. Strain J. Tree Methods for Moving Interfaces. J Comput Phys. 1999; 151:616–648. doi: [10.1006/jcph.1999.6205](https://doi.org/10.1006/jcph.1999.6205)

33. Shu CW, Osher S. Efficient Implementation of Essentially Non-Oscillatory Shock Capturing Schemes II (two). *J Comput Phys*. 1989; 83:32–78. doi: [10.1016/0021-9991\(89\)90222-2](https://doi.org/10.1016/0021-9991(89)90222-2)
34. Liu XD, Osher S, Chan T. Weighted Essentially Non-Oscillatory Schemes. *J Comput Phys*. 1996; 126:202–212.
35. Najman L. http://www.laurentnajman.org/heart/H_data.html; 2014.