A general finite element model for segmentation in 2, 3, and 4 dimensions

Jörg Bredno, Thomas Lehmann, Klaus Spitzer

Institute of Medical Informatics, Medical Faculty

Aachen University of Technology (RWTH), D–52057 Aachen, Germany

ABSTRACT

Medical imaging modalities often provide image material in more than two dimensions. However, the analysis of voxel data sets or image sequences is usually performed using only two–dimensional methods. Furthermore, four–dimensional medical image material (sequences of stacks of images) is available already for clinical diagnoses. Contrarily, four–dimensional image processing methods are almost unknown. We present an active contour model based on balloon models that allows a coherent segmentation of image material of any desired dimension. Our model is based on linear finite elements and combines a shape representation with an iterative segmentation algorithm. Additionally, we present a novel definition for the computation of external influences to deform the model. The appearance of relevant edges in the image is defined by image potentials and a filter kernel function. The filter kernel is applied with respect to the location and orientation of finite elements. The model moves under the influence of internal and external forces and avoids collisions of finite elements in this movement. Exemplarily, we present segmentation results in 2D (radiographs), 3D (video sequence of the mouth), and 4D (synthetic image material) and compare our results with propagation methods. The new formalism for external influences allows the model to act on greylevel as well as color images without pre–filtering.

Keywords: spatio–temporal image data, geometric deformable model, image segmentation, finite element model

1. INTRODUCTION

Spatial and temporal information is nowadays available from many imaging modalities such as Video, CT, MR, and Ultrasound. A quantification of temporal or spatial data by hand can not be used in clinical routine as this is too time–consuming even for experienced observers.[1] For two–dimensional images, active contour models have proven to yield robust segmentation results.[2] The classic approach of KASS ET AL.[3] is widely used for medical image analysis and is still topic of current improvements and research.[4] Nevertheless, the development of new medical imaging systems exceeds the development of image analysis methods, especially for image material of higher dimensions. So far models to quantify image material of higher dimensions exist only for highly specialized applications[5] and are usually bounded to three dimensions. The spatial and temporal extension of image space gives need for new developments. When a tracking of objects and their movement in image sequences is needed, often the image sequence is segmented using propagation methods.[6] It is well recognized that this propagation of two–dimensional contours in an image sequence might result in fatal distractions of the contour at few ambiguous images in the sequence.[7] This problem is e.g. tackled using motion estimation.[8] For spatial data, the propagation of segmentation results through slices is still used[9] even though three–dimensional models have been published years ago.[10] More and more often, a medical diagnosis is based on image material of up to four–dimensions.[11] So far, no coherent model exists to represent and segment spatial objects that additionally move and change shape over a period of time under observation.

2. METHODS

Our model is a generalization of the well known balloon model.[10] It uses linear finite elements and mechanical influences to perform an iterative segmentation process in image material of d dimensions. The segmentation result is a geometrical representation of objects contained in image space can be used for many different tasks in the quantification of medical image data.

Further author information: Send correspondence to Jörg Bredno
Institute of Medical Informatics, Aachen University of Technology (RWTH), D–52057 Aachen, Germany
E–mail: jbredno@mi.rwth-aachen.de

In *Medical Imaging 2000: Image Processing*, Kenneth M. Hanson, Editor,
Proceedings of SPIE Vol. 3979 (2000) • 1605-7422/00/$15.00

2.1. Image Space

To allow a formulation of the model which is independent of the dimension, first we define an image as a discrete, convex subset of the d–dimensional space:

$$I \subset \mathbb{N}_0{}^d, d \in \{2, 3, 4 \ldots\} \tag{1}$$

So far, we assume a Cartesian image space with equally–spaced axes, i.e. square picture elements (pixels) and cubic volume elements (voxels). We denominate four–dimensional image elements as *stixels* (spatio–temporal image elements), but will use the term *image element* for all elements of I irrespective of its dimension d.

The image space I is mapped onto a space of image values W:

$$W \subset \mathbb{R}^m, m \in \{1, 2 \ldots\} \tag{2}$$

The image function $\vec{f}$ assigns an at least one–dimensional image value for each image element of I:

$$\vec{f} : \vec{x} \in I \rightarrow \vec{w} \in W \tag{3}$$

The dimension m of the image values is not limited in our model. E.g., it is $m = 1$ for grey scale images and $m = 3$ for color images. Medical imaging systems offer an increasing amount of multi–modal images with $m > 1$.

2.2. Shape Representation

An object represented by the model is a subset of I. The shape representation is the surface of this object with $d-1$ degrees of freedom. For its definition, we use a set V of discrete vertices v_i and a set E of finite elements e_j which are also called edge elements or *edgels*.[10] All vertices v_i are elements of the image space I, they determine a position in I with their vector $\vec{v}_i$.

$$V = \{v_i\},\ v_i \in I \tag{4}$$

The finite elements or edgels are affine simplices K of dimension $d-1$. Each finite element e_j provides the connection and orientation for d vertices $v_{j,1\ldots d}$ that support this edgel:

$$E = \{e_j\},\ e_j = K(v_{j,1}, \ldots, v_{j,d}),\ v_{j,k} \in V \tag{5}$$

The general affine simplex K of dimension n is defined as:

$$K(v_0, v_1, \ldots, v_n) = \left\{ \lambda_0 \vec{v}_0 + \ldots + \lambda_n \vec{v}_n \;\middle|\; 0 \leq \lambda_k \leq 1, \sum_{k=1}^{n} \lambda_k = 1 \right\} \tag{6}$$

For each edgel e_j, the unity normal vector $\vec{n}_j$ pointing outward and its size $|e_j|$ are available. Further, we define $U^v_\Delta(v_i)$ as a set of vertices which lay no more that Δ edges afar of v_i. Here an edge is the straight connection between two vertices v_i and v_k which share at least one edgel e_j. $U^e(v_i)$ is defined as the set of all edgels e_j which are supported by v_i. The shape representation yields the consistent surface of a coherent subset of I if the following necessary requirements are met:

- Every vertex v_i supports at least d edgels e_j.
- Every edgel e_j has exactly d different adjacent edgels. Two edgels are adjacent if they share $d-1$ vertices.
- No edgels e_j and $e_k, j \neq k$ exist that share d vertices.
- The sequence $v_{j,1\ldots d}$ of vertices, which defines the orientation of e_j, complies with the orientation of the shape's surface.
- The edgels e_j do not intersect with each other, that is $\{e_j \cap e_k\} = \{\}\ \forall\, j \neq k$.

- A path can be found to reach each edgel e_j from every other edgel e_k by stepping over adjacent edgels.

So far, we have not proved the sufficiency of these requirements. Note that the last requirement leads to the representation of one single coherent object. If this requirement is not enforced, it becomes possible for the model to represent more than one object contained in I. Instances of this generalized model for 2,3, and 4 dimensions result in the following shape representations:

- In image space of two dimensions x and y, an object is represented by a polygonal shape.
- For spatial image data ($d = 3$), the finite element model forms a triangulated surface.
- A sequence of 2D images also results in an image space with $d = 3$, the image space is accessed using the coordinates x, y and t. The model consists of triangular finite elements where each vertex defines a position (x, y) in the image plane and its time of occurrence t. Each finite element represents a line segment of a polygonal shape in different images of the sequence. This results in segments which are scaled and translated over a short period of time. The line segment at the time t_c is obtained when the edgel is cut with a plane defined by $t = t_c$.
- An image space of four dimensions comprises a spatial object and its movement and change of shape in time. The model results in a set E of tetrahedrons that connect four vertices which each define a position (x, y, z) in space and a time of occurrence t. Each edgel represents a moving facet of a triangulated object. Here, the cut of an edgel with a hyper plane defined by $t = t_c$ may result in one or two triangles, the triangles obtained by cutting all edgels with that hyper plane result in the triangulated object at time t_c. The triangulated facets are scaled and translated during the existence of each edgel e_j which is given by the minimal and maximal time coordinate of the vertices $v_{j,k}$.

2.3. Mechanical Influences

Influences act on the model and result in movements towards the surface of objects contained in an image. A mechanical formulation was used because forces allow a vectored definition of influences independent of the image's dimension d. Different notations are used in literature to denote such influences. Internal[12] or intrinsic[13] influences are used to describe a preferenced shape of active contour models. The influences are used to control the reconstruction of ambiguous parts of an image. They may also include tendencies of the contour to seek object surfaces. Significant image information is used to attract or stabilize an active contour model at object boundaries. These external or extrinsic influences are calculated from image values $\vec{w} \in W$. In different models, these forces are either imposed on edgels or vertices. In our model, forces $\vec{F}_{e_j}$ that act on edgels are later transformed to their resulting forces $\vec{F}_{v_i}$ acting on the vertices as well, because only vertices define a position in the image space and can be moved. The location of edgels is always defined by their supporting vertices.

2.3.1. Pressure

The mechanical formulation of a tendency to seek object boundaries is realized by an internal or external pressure p that results in a pressure force $\vec{F}^p_{e_j}$ that acts on every edgel e_j:

$$\vec{F}^p_{e_j} = p \cdot |e_j| \cdot \vec{n_j} \tag{7}$$

The pressure force $\vec{F}^p_{e_j}$ is directed along the edgel's normal vector $\vec{n}_j$ and proportional to the size $|e_j|$ of the edgel and the chosen pressure p. Positive pressure results in a seek of the contour to the outward, negative pressure to a tendency of the contour to shrink.

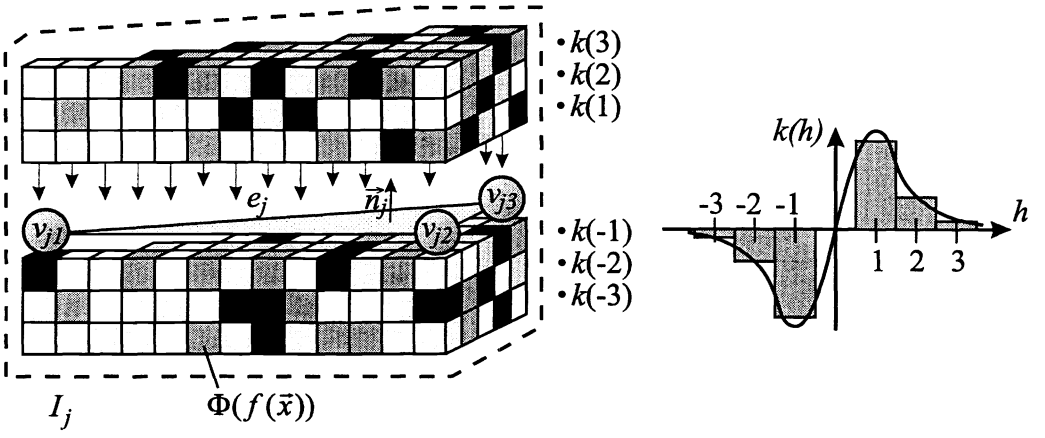

Figure 1. External influences for a three–dimensional edgel e_j and one exemplary filter kernel $k(h)$.

2.3.2. Image information

Most importantly, edges pick up image information. Edgels e_j are influenced by all image values that are read from the image subset I_j. This subset is defined as a set of image elements in the vicinity of e_j:

$$I_j = \left\{ \vec{x} \in I \;\middle|\; \vec{x} = \vec{x}_e + h \cdot \vec{n}_j, \vec{x}_e \in e_j, h_{\min} \leq h \leq h_{\max} \right\} \tag{8}$$

This set contains all image elements that are reached by adding the normal vector $\vec{n}_j$ scaled in the range $[h_{\min}, h_{\max}]$ to the image elements in the affine simplex of e_j. In our model, the dimension $\dim(I_j)$ of the subset from which external influences are read, is the dimension d of the image material itself.

For the computation of the external influences, for every component in the image values $\vec{w} \in W$, a corresponding entry in the image potentials $\vec{\Phi}(\vec{w})$ is assigned. The image potentials are chosen to describe the appearance of objects in different color channels. E.g., for $m = 1$, $\Phi(w)$ is a scalar function for greylevels. For RGB color spaces with $\vec{w} = (w_r, w_g, w_b)^\top$, $\vec{\Phi} = (\Phi_r(w_r), \Phi_g(w_g), \Phi_b(w_b))^\top$ assigns individual potentials for the values in different color channels. Furthermore, we use a filter kernel $\vec{k}(h)$ with entries for each dimension of the value space. This adaptable filter kernel defines the strength and scale of gradients in the image at the border of objects of interest. So far, we use nearest neighbor interpolation to map continuous positions $\vec{x}$ into the discrete image space I. External influences acting on an edgel e_j are defined with respect to $\vec{k}(h)$ and $\vec{\Phi}(\vec{w})$. The absolute value of the external force $\vec{F}^e_{e_j}$ is the unstandardized correlation of the image potentials $\vec{\Phi}$ and the filter kernel $\vec{k}(h)$ applied in direction of the normal vector $\vec{n}_j$:

$$\vec{F}^e_{e_j} = -\vec{n}_j \sum_{\vec{x}_e \in e_j} \sum_{h=h_{\min}}^{h_{\max}} \vec{k}(h) \cdot \vec{\Phi}\left(f(\vec{x}_e + h \cdot \vec{n}_j)\right) \tag{9}$$

The image potentials are summed up and individually weighted by the filter kernel $\vec{k}(h)$ in respect to their distance h to the edgel (Fig. 1). This force is always directed against the normal vector of each edgel. The choice of $\vec{k}(h)$ allows the model to find high gradients in the image as well as the occurrence of image potentials with a magnitude that equals the pressure p. It can be interpreted as a directional adaptive filter of adaptable scale, the image is always filtered normal to the direction of edgels. Therefore, our model works on the original image material with no pre–filtering. In Figure 1, external information is read from the three–dimensional image subset I_j of a triangular edgel e_j. The exemplary filter kernel of size 7 is the discrete approximation of the first order derivative of a Gaussian function with a standard deviation of one pixel.

2.3.3. Supporting vertices

The forces $\vec{F}^p_{e_j}$ and $\vec{F}^e_{e_j}$ have to be transferred to the supporting vertices $v_{j,k}$. The pressure force $\vec{F}^p_{e_j}$ is equally divided:

$$\vec{F}^p_{v_i} = \frac{1}{d} \sum_{e_j \in U^e(v_i)} \vec{F}^p_{e_j} \tag{10}$$

Every vertex v_i supports $1/d$ of all pressure forces $\vec{F}^p_{e_j}$ in its vicinity. In contrast, the external force is divided to the supporting vertices with respect to the position of image elements and their potentials. Vertices are more influenced by image potentials from close image elements. Assuming that the position $\vec{x}_e$ of image elements of edgel e_j is composed as

$$\vec{x}_e = \sum_{l=1}^{d} \lambda_{l,j} \cdot \vec{v}_{l,j}, \tag{11}$$

(compare Eq. (6)), the $\lambda_{l,j}$ in the range $[0,1]$ are near to 1 if x_e is near to $v_{l,j}$ and decrease with increasing distance. The external force $\vec{F}^e_{v_i}$ acting on vertices is computed as

$$\vec{F}^e_{v_i} = - \sum_{e_j \in U^e(v_i)} \vec{n}_j \sum_{\vec{x}_e \in e_j} \sum_{h=h_{\min}}^{h_{\max}} \vec{k}(h) \cdot \vec{\Phi}\left(f(\vec{x}_e + h \cdot \vec{n_j})\right) \cdot \lambda_{l,j}. \tag{12}$$

Here l is the corresponding index of vertices in e_j so that $v_i = v_{l,j}$. Eq. (12) is motivated to achieve an equilibrium not only of forces but also of turning moments for all external influences and supporting forces $\vec{F}^e_{v_i}$. The equilibrium of turning moments was chosen because it enhances the property of finite element models to find segmentation results independent of their sampling.[2]

2.3.4. Deformation force

Internal influences from deformation energies directly result in forces $\vec{F}^d_{v_i}$ on vertices. The deformation energy of a flexural rigid body is proportional to its absolute 2^{nd} order derivative. Therefore, the aim of the deformation force is to reduce the models 2^{nd} order derivative. This is achieved easily for the model with discrete vertices,

$$\vec{F}^d_{v_i} = \frac{s_d}{|U^v_\Delta(v_i)|} \sum_{v_k \in U^v_\Delta(v_i)} \vec{v_k} - \vec{v_i}, \tag{13}$$

if $\Delta = 1$. Then every vertex is pulled with adjustable strength to the average position of all vertices in its vicinity. For $\Delta > 1$, we use weights w_n in respect to the distance $\Delta_{i,k}$ to calculate the average position of $U^v_\Delta(v_i)$. Here $\Delta_{i,k}$ is the number of straight edges between v_i and v_k.

$$\vec{F}^d_{v_i} = \frac{s_d}{\sum_{v_k \in U^v_\Delta(v_i)} w_n(\Delta_{i,k})} \sum_{v_k \in U^v_\Delta(v_i)} w_n(\Delta_{i,k}) \cdot (\vec{v_k} - \vec{v_i}). \tag{14}$$

The decreasing sequence $w_n(\Delta_{i,k}) : \mathbb{N} \to \mathbb{R}$ is heuristically chosen. We use

$$w_n(\Delta_{i,k}) = \frac{A}{A + \Delta_{i,k}} \tag{15}$$

with $A \in [1;3]$.

2.4. Iterative Segmentation

The model detects structures in image space I with an iterative segmentation algorithm. During segmentation, vertices move until they are in contact with significant edges whose appearance is coded in the filter kernel $\vec{k}(h)$. The segmentation starts with an initial contour. This contour is either a seed for an inflating segmentation ($p > 0$) or an instance of the model that encloses the whole image space for a deflating process ($p < 0$). An initial seed contour is a contour that represents the subset I_s defined as

$$I_s = \left\{ \vec{x} \in I \;\middle|\; (\vec{x} - \vec{x}_s)^2 \leq r_s^2 \right\} \tag{16}$$

with small initial radii r_s and $\vec{x}_s$ laying in the center of an object of interest.

After the initialization of the contour, all forces $\vec{F}_{v_i}$ acting on vertices are calculated. The vertices are then moved under these influences. The forces are normalized so that the movement of the contour is independent of the size of edgels. For this normalization, we define the supported size O_i for every vertex v_i.

$$O_i = \frac{1}{d} \sum_{e_j \in U^e(v_i)} |e_j| \tag{17}$$

O_i gives the average amount of image elements in the contour that have influence on $\vec{F}^e_{v_i}$. For normalization of the deformation force $\vec{F}^d_{v_i}$, we use the sampling $\overline{L}$, this is the average distance of vertices in the contour. All vertices are moved proportional to the sum of forces:

$$\Delta\vec{v_i} = \frac{1}{\beta}\left(\frac{1}{\overline{L}}\vec{F}^d_{v_i} + \frac{1}{O_i}(\vec{F}^p_{v_i} + \vec{F}^e_{v_i})\right) \tag{18}$$

This movement corresponds to the movement of objects under the influence of liquid friction with the coefficient β. The shift $\Delta\vec{v_i}$ of vertices needs not to be whole–numbered. We therefore store the coordinates $\vec{v_i}$ as vectors of $\mathbb{R}^d$ and limit this space with the bounds of I.

The movement of vertices changes their distance and therefore the size of finite elements. After each iterative movement, the contour is resampled. This resampling regards all Cartesian distances $L_{i,k}$ of adjacent vertices v_i and v_k.

- If the distance $L_{i,k}$ is above an upper bound $L_{\max}$, a new vertex v_l is created at position $\frac{1}{2}(v_i + v_k)$. All edgels that are supported by both, v_i and v_k, are duplicated. In one set of duplicates, v_i is exchanged for v_l, in the other set this exchange is done with v_k.
- If $L_{i,k}$ is below a lower bound $L_{\min}$, also a new vertex v_l is created at position $\frac{1}{2}(v_i + v_k)$. All edgels that contain both, v_i and v_k, are deleted. For edgels that contain either v_i or v_k, this vertex is replaced by v_l. Thereafter, v_i and v_k are deleted as well.

The resampling results in changes of the represented shape. These changes can be minimized if the order of resampling operations is optimized and the new vertex v_l is placed at a more appropriate position compared to the center of v_i and v_k.[14] So far, we heuristically set $\frac{1}{3}L_{\max} \leq L_{\min} \leq \frac{2}{5}L_{\max}$ and estimate $\overline{L} = \frac{1}{2}(L_{\min} + L_{\max})$.

Every vertex keeps track of its own position during the previous iterations. If the Cartesian distance from the actual position to one of the prior positions is below a minimum value L_f, the vertex is frozen. No forces are calculated for frozen vertices, they remain at their current position. The segmentation process ends if all vertices are frozen or if a maximum count of iterations is reached. The position of frozen vertices still contributes to the deformation force $\vec{F}^d_{v_i}$. This bounds the movements of adjacent active vertices and contributes to an implicit annealing process of the contour, the activity of vertices decreases in the vicinity of already frozen vertices.

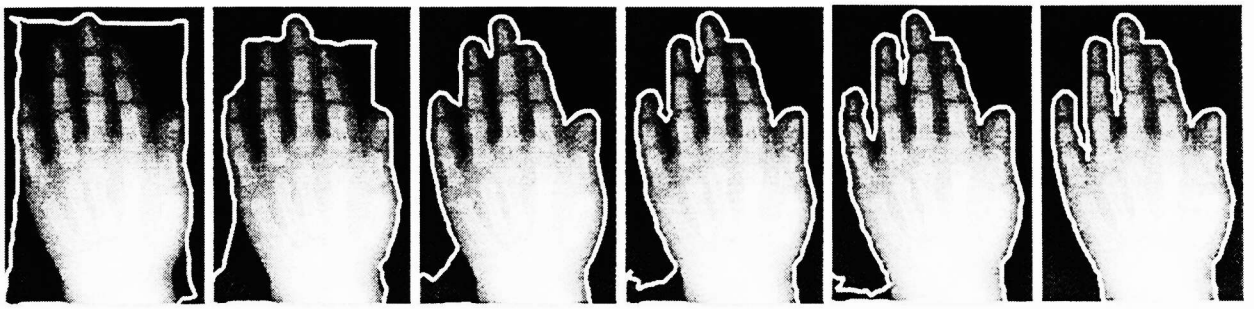

Figure 2. 2-D-segmentation result of a radiograph.

2.5. Handling of Collisions

Following the movement and resampling of the contour, collisions of finite elements might occur. A collision of finite elements is detected if $\{e_j \cap e_k\} \neq \{\}\ \forall\, j \neq k$, that is if image elements are found that belong to the affine simplices of two different edgels e_j and e_k. For the two-dimensional instance of the model, this cut of two edgels can easily be solved by exchanging the two vertices $v_{j,1}$ and $v_{k,1}$. In higher dimensions, the handling of collisions is more complicated and requires the start of the following algorithm:

- First all finite elements that are involved in collisions are deleted. The resulting contour does not fulfill the conditions for a consistent surface given in Sec. 2.2.
- Afterwards, vertices are deleted from the contour if they support no edgels.
- Then the cuts that exist in the inconsistent model are extracted. The resulting cut objects are contours of dimension $d-1$ that are based on vertices in image space of dimension d. E.g., a cut object in a triangulated surface is a non-planar polygon. For the four-dimensional model, this polygon occurs at one point in time, changes shape and size and vanishes again. Such cut objects are found by tracking coherent inconsistencies in the model. These inconsistencies are formed by $d-1$ vertices that support only one edgel. Such vertices are used to create an edgel in the cut object of dimension $d-1$.
- Then all vertices are checked. If they support edgels in more than one cut object, the vertices as well as all supported edgels are deleted. This constellation can not be solved by the following consistency operations.
- For every cut object, its size, the location of its center of gravity and an average normal vector are computed.
- If two cut objects approximately match in size, position and orientation, they are connected with new finite elements to close both cuts.
- If no matching cut object is found, single objects are closed by the creation of a new vertex in its center of gravity and the creation of new finite elements that connect this new vertex with all vertices in the cut object.

After this operation, one or more coherent subsets of I are represented by the model. In dependency of the quantification task, supernumerary objects are either accepted or deleted.

3. RESULTS

The model has been implemented in C++ on different UNIX-based platforms. The implementation consists of polymorphistic objects which create an adequate instance of the model according to the present image material. We present examples for instances of the model in 2, 3, and 4 dimensions.

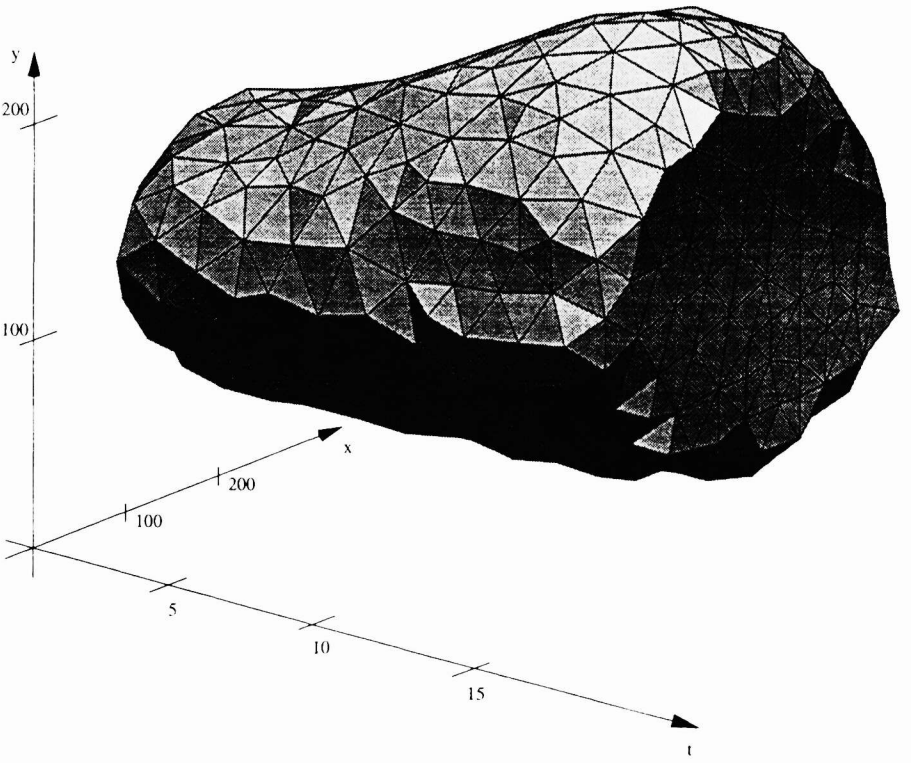

Figure 3. A coherent segmentation result of an image sequence.

3.1. 2D–Instance of the Model

We use the model to detect the contour of body parts in radiographs. The shape of the detected contour is used as one method to categorize radiographs which is necessary for content–based image retrieval in medical applications.[15] To detect a contour in radiographs, we initialize the model at the border of the image and apply negative pressure ($p = -2$). An exemplary radiograph of a right hand (Fig. 2) is 357×540 pixel in size and has 256 gray values. The contour is sampled with $L_{\max} = 20, L_{\min} = 8$. We set the image potential Φ proportional to the image values w using $\Phi(w) = 0.0138w$. The filter kernel is located only on the inside of the contour. Its size is $h_{\min} = -3, h_{\max} = -1$ and its values are set to $k([-3, -2, -1]) = [\frac{1}{2}, 1, \frac{3}{2}]$. The strength of the deformation forces is set to $s_d = 0.2$. The segmentation result is found after 320 Iterations i . These parameters can be applied on a wide variety of skeletal radiographs and were set using an automated method.[16]

3.2. 3D–Instance of the Model

As a segmentation result in 3D, we present the tracking of a human lip in an image sequence acquired with a CCD–camera. The images where stacked to create a three–dimensional image space. The green channel of the RGB image sequence was used for the detection. The images are 274×205 pixel in size, the exemplary short sequence consists of 19 images and shows the pronunciation of the syllable 'ba'. The whole study consists of many sequences that show the pronunciation of different selected syllables and is used for phoniatric diagnoses. A three–dimensional instance of the model segmented the outer contour of the lips in this image stack. Whereas prior methods, which used propagation methods, often drifted away in later images in the stack, the three–dimensional model yields a coherent segmentation result (Fig. 3). Note that the reduction of the 2^{nd} oder derivative of the contour along the t–axis corresponds to a local motion estimation of the tracked object.

To obtain segmentation results for every image in the sequence, the resulting three–dimensional object was cut into slices along the t–axis. The resulting two–dimensional contours where superimposed into the original images (Fig. 4).

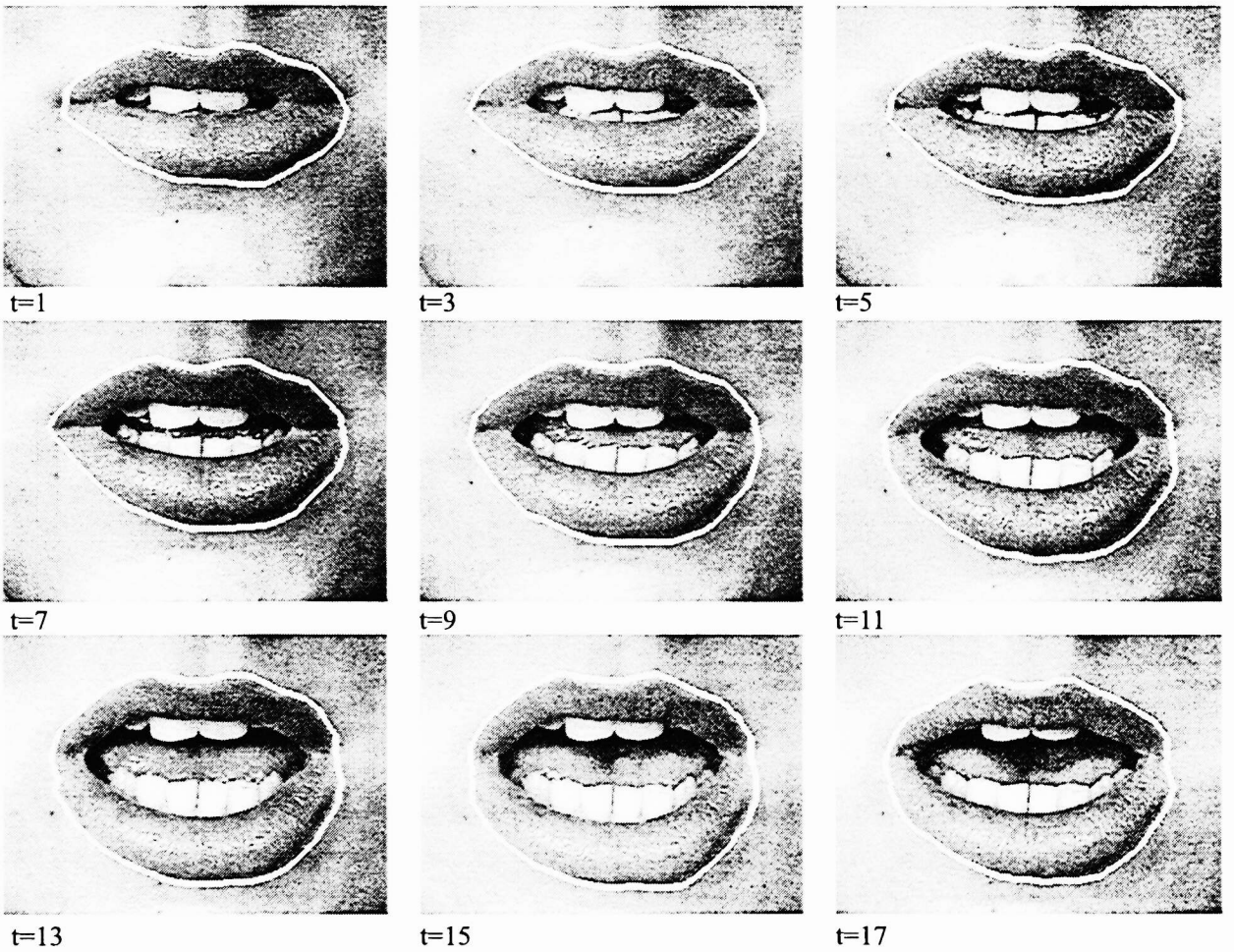

Figure 4. Result of the segmentation for planes in the image sequence.

Not all sequences acquired within this study can be sufficiently segmented using only the green channel. We will therefore extend the detection method to a three–dimensional color space.

3.3. 4D–Instance of the Model

We still gain experiences with the four–dimensional model. As an example, we present the segmentation result of a synthetic dataset. This dataset is an image space $I \subset \mathbb{N}^4$ that consists of 32 successive volume datasets of the size $32 \times 32 \times 32$ voxel. Each voxel set contains a binary image of a sequence in which a cube is morphed into a sphere. The initialization of the model was a affine 4–simplex consisting of five vertices and five edgels in the center of this four–dimensional image space. Then the model performed 25 segmentation iterations with a sampling of $L_{\max} = 10, L_{\min} = 3$. The resulting contour consists of 644 vertices in $\mathbb{R}^4$ which are connected by 3559 edgels. This contour was then cut with different hyper planes $t = t_c$ (Fig. 5). Each edgel results in 0, 1, or 2 triangular edgels in $\mathbb{R}^3$, all these edgels form a triangulated surface of the object at the time t_c.

Additionally, to compare our model with propagation methods, the segmentation process was also started for subsets of I in two and three dimensions. In order to achieve the same sampling, when changing from dimension d_1 to d_2, the average distance between vertices $\overline{L}$ has to be readjusted so that $\overline{L_1}/\sqrt{(d_1)} \approx \overline{L_2}/\sqrt{(d_2)}$. One spatial data set of $32 \times 32 \times 32$ voxel would result in 86 vertices and 168 edgels with $L_{\max} = 8, L_{\min} = 3$. The incoherent dataset of 32 three–dimensional contours would therefore consist of 2752 vertices and 5376 edgels. One central slice of 32×32 pixel was segmented using $L_{\max} = 7, L_{\min} = 3$. The resulting contour consists of 16 vertices and edgels.

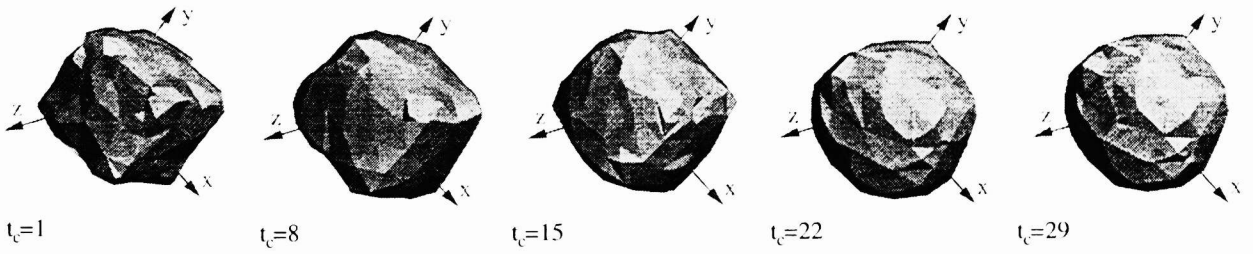

Figure 5. A coherent 4–D contour.

We estimate that the segmentation of the dataset in 1024 two–dimensional slices would result in more than 10.000 edgels and vertices with no coherence in depth and time. The benefit of our model is not only the coherence but also a reduced amount of data for the segmentation result.

4. DISCUSSION

Our generalized finite element model fulfills the task of image segmentation in up to four dimensions and automatically incorporates the knowledge of spatial and temporal coherence of objects. It is therefore more stable than propagation methods for segmentation of image sequences or stacks. Additionally, the knowledge of the coherence is automatically contained in the dataset and needs not to be recomputed from propagation slices.

Our new method for the computation of external influences allows the segmentation process to work on the original image data with no need for pre–filtering. The newly introduced adaptable kernel function for external influences allows the model to control the direction of this filter that is used to detect significant gradients in the image. The formulation also holds for a value space of higher dimension, e.g. RGB color images. Our model reads image information from subsets I_j of the image with $\dim(I) = \dim(I_j)$. This intensifies the ability of active contour models to consider global image information.

With these extensions, we also increase the complexity of the segmentation process. The order of magnitude is dominated by the computation of external influences and was estimated to $O(m \cdot d^2)$. As a consequence, segmentation results in 4D exist so far only for synthetic image material with coarse image space resolution.

For the use of the model on medical image data, different parameters have to be set. The model looses the ability to reproducibly segment image material if these parameters are heuristically chosen by experts. For the use of the model on medical images, we developed methods to determine the parameters using one hand segmentation result. This allows physicians to perform the required parameter adjustments.[16]

Though the concepts are formulated in d dimensions with no upper bound for d, instances of the model exist only for 2, 3, and 4 dimensions as real world space is bounded to 4D.

ACKNOWLEDGMENTS

The author holds a scholarship from the *Studienstiftung des Deutschen Volkes* (German Scholarship Foundation).

REFERENCES

1. T. McInerney and D. Terzopoulos, "Deformable models in medical image analysis: A survey," *Medical Image Analysis* **1**, pp. 91–108, 1996.
2. V. Metzler, J. Bredno, T. Lehmann, and K. Spitzer, "A deformable membrane for the segmentation of cytological samples," *Proc. SPIE* **3338**, pp. 1246–1257, 1998.
3. M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International Journal of Computer Vision* **1**(4), pp. 321–331, 1988.
4. D. J. Kang, "A fast and stable snake algorithm for medical images," *Pattern Recognition Letters* **20**, pp. 507–512, 1999.

5. D. Kucera and R. W. Martin, "Segmentation of sequences of echocardiographic images using a simplified 3D active contour model with region-based external forces," *Computerized Medical Images and Graphics* **21**, pp. 1–21, 1997.
6. C. L. Lam and S. Y. Yuen, "An unbiased active contour algorithm for object tracking," *Pattern Recognition Letters* **19**, pp. 491–498, 1998.
7. S. Chan, C. W. Ngo, and K. F. Lai, "Motion tracking of human mouth by generalized deformable models," *Pattern Recognition Letters* **20**, pp. 879–887, 1999.
8. N. Peterfreund, "The velocity snake: Deformable contour for tracking in spatio–velocity space," *Computer Vision and Image understanding* **73**(3), pp. 346–356, 1999.
9. B. Levienaise-Obadia and A. Gee, "Adaptive segmentation of ultrasound images," *Image and Vision Computing* **17**(8), pp. 583–588, 1999.
10. L. D. Cohen and I. Cohen, "Finite–element methods for actice contour models and balloons for 2–D and 3–D images," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **15**(11), pp. 1131–1147, 1993.
11. J. Duann, S. Lin, W. Hu, and J. Su, "Computer system for four–dimensional transoesophageal echocardiographic image reconstruction," *Computerized Medical Images and Graphics* **23**, pp. 173–179, 1999.
12. V. Chalana, D. T. Linker, D. R. Haynor, and Y. Kim, "A multiple active contour model for cardiac boundary detection on echocardiographic sequences," *IEEE Transactions on Medical Imaging* **15**, pp. 290–298, June 1996.
13. R. Grzeszczuk and D. Levin, "Brownian strings: Segmenting images with stochastically deformable contours," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **19**(10), pp. 1100–1114, 1997.
14. A. E. Johnson and M. Hebert, "Control of polygonal mesh resolution for 3–D computer vision," *Graphical Models and Image Processing* **60**, pp. 261–285, 1998.
15. T. Lehmann, B. Wein, J. Dahmen, J. Bredno, F. Vogelsang, and M. Kohnen, "Content-based image retrieval in medical applications: A novel multi–step approach," *Proc. SPIE* **3972**, 2000. (in press).
16. J. Bredno, T. Lehmann, and K. Spitzer, "Automatic parameter setting for balloon models," *Proc. SPIE* **3979**, 2000. (this issue).