Human-computer interaction in medial image analysis
Delgado Olabarriaga, S.

Citation for published version (APA):
Olabarriaga, S. D. (1999). Human-computer interaction in medial image analysis Amsterdam

General rights
It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations
If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: http://uba.uva.nl/en/contact, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.
Chapter 4

Piecewise DM: A Locally Controllable Deformable Model*

“No meio do caminho tinha uma pedra
 tinha uma pedra no meio do caminho
 tinha uma pedra
 no meio do caminho tinha uma pedra.”
in No meio do caminho, by Carlos Drummond de Andrade.

Deformable model methods (DM) refer to a large and popular group of segmentation techniques where an initial curve is deformed based on integral constraints upon the object's boundary. The constraints express a model for the boundary in ideal conditions, describing the expected values of local features derived from shape and image properties. Deformation is a consequence of the optimisation of an objective function measuring the difference between the model and the boundary of an object in the image, starting from a curve provided by a higher level mechanism.

These methods represent a promising platform for the implementation of interactive segmentation because they allow for the elegant combination of information derived from the image data, constraints expressing prior knowledge about the boundary, and information provided by the user. Several difficulties were found, however, when adopting existing DM to implement an interactive segmentation method based on the structured approach proposed in chapter 3. In that approach, the segmentation process consists of the following steps: the user provides an initial curve that is deformed on the basis of a boundary model containing prior knowledge about the segmentation problem. Occasionally, the optimised boundary does not correspond to the desired solution, and the user can edit it with special interactive tools. In

*This work was developed with Pia Pfluger, Mathematics Department - WINS, University of Amsterdam. A paper based on this chapter has been submitted to the journal Computer Vision and Image Understanding.
such cases, the parameters for the DM are locally adjusted based on the information provided by the user, and optimisation is repeated.

In general, the difficulties were related to limitations in representing the boundary of objects, in describing the shape and image properties of the ideal boundary, and in finding the right balance between the shape and image components in the model. Moreover, a general formulation was missing to enable the integration of interesting ideas scattered among the existing methods. These difficulties motivated the development of a new method, called "Piecewise DM." Seen in a broader context, the goal of Piecewise DM is to provide a general framework for the implementation of a flexible and controllable method with a larger rate of success in real segmentation problems.

This text is organised in three main parts that can be seen as individual contributions. In the first part (section 4.1), we identify the basic elements of a DM, with examples of implementations found in the literature. In section 4.2, we define requirements posed on a DM to address the demand for a more flexible and controllable method, reviewing eligible methods described in the literature. In section 4.3 we present the Piecewise DM, an extension to the class of DM addressing these requirements. This method is applied in section 4.4 to a complex segmentation task with a relatively simple customisation.

4.1 Basic Components of Deformable Models

Deformable models originated from the classical "snakes" described in [12]. The essence of most existing DM can be captured in terms of the following aspects: how the boundary geometry is represented, how the boundary model is defined, how the objective function is constructed, how optimisation deforms the curve and how the whole process is initialised.

1. **Representation of the boundary geometry.** A boundary is represented by a parameterised curve \( C \) in the image \( I \):\(^\dagger\)

\[
C(t) = [x(t), y(t)],
\]

where \( t \) is the path along the boundary and \([x(t), y(t)]\) are the curve positions in the image grid.

The type of representation determines the domain of objects that can be delineated with the DM, e.g.: only closed curves [38], only open curves [25], free-form curves ([12] and [23]), surfaces enclosing a volume ([24], [42] and [18]), instances of parametric templates [47] and disconnected components ([18] and [30]).

2. **The boundary model** \( M \) defines ideal properties observed for the object of interest, characterised in terms of local features based on shape and image data (e.g. curvature and image gradient). The types of local features determine the model's descriptive power, i.e. its capability to represent knowledge about...

\(^\dagger\)For simplicity, in this text we use a notation for planar curves, but all the concepts presented here also apply to 3-D curves or surfaces.
4.1. Basic Components of Deformable Models

the boundary of interest. Example: a feature based on the gradient of the image intensity ([44] and [2]) can only describe boundaries located at step-edges, while more generic DM use an “image potential” computed from different image features ([12], [33] and [3]). Likewise, if the smoothness constraint is posed on the curve by minimising its curvature [12], the model cannot describe boundaries with sharp corners; instead, more flexible shape features can be used ([32], [28] and [36]).

3. The objective function ($\Theta$) defines how the ideal boundary model $M$ is combined with the curve $C$ representing the boundary in the image. In many cases, $\Theta$ is composed of a weighted sum of terms $T_i(\cdot)$:

$$\Theta(C) = \sum_{i=1}^{N} W_i T_i(C, M), \quad (4.2)$$

where $N$ is the number of features used to describe the ideal boundary, $T_i(\cdot)$ measures the difference between $C$ and $M$ with respect to the local feature of type $i$, and $W_i$ is the relative importance of each term.

Since the shape and image features are measured locally, $\Theta$ is redefined as follows:

$$\Theta(C) = \int \sum_{i=1}^{N} W_i(t) T_i(t, C, M) \, dt. \quad (4.3)$$

In a simple case, the DM could have the following configuration: $T_1$ measures the boundary curvature $\kappa$, $T_2$ measures the image gradient magnitude $|\nabla I|$, and the weights are constant along the boundary and tuned for each new application.

4. The optimisation of $\Theta$ results in a deformation process by which the geometry of the initial curve $C_0$ is transformed into $C$, such that $\Theta(C)$ is minimal. In most existing methods, the optimisation operates on the boundary geometry only (e.g. [12]). Alternatively, other parameters can also be modified during the optimisation, such as the curve topology [21], the balance of terms in the objective function ([47] and [25]) and the number of degrees of freedom for deformation ([24], [17] and [48]). Usually, the objective function refers to a single connected boundary, with exceptions such as [47] and [18], where disconnected boundary components are deformed simultaneously.

Several optimisation strategies can be used to search for a local minimum of the objective function, such as dynamic programming [1], solving a system of equations ([12] and [23]), the conjugate gradients method ([44] and [18]), find a situation of stability in a dynamic system ([24] and [17]), embedding the function in a level set ([19], [2] and [26]) and simulated annealing [35].

5. Initialisation of a DM involves the construction of an initial curve and an objective function corresponding to the boundary model for the object at hand. The initial curve is usually provided by the user with free-hand drawing tools.
([12] and [25]) or by adjusting a template to the image ([47], [42] and [48]). Other possibilities are to create the initial curve from a coarse boundary determined during a pre-processing step [18], from a neighbouring slice [45], or from statistical knowledge ([40] and [35]). Note that the resulting segmentation may depend greatly on the initial curve if local optimisation is used.

The objective function is usually hardwired in the method and cannot be easily adapted to specific knowledge about the segmentation problem (e.g. [12], [47], [44], [18]). Exceptions to this general rule are described in [20] and [25], where the boundary model and the corresponding objective function are built and updated based on pictorial input provided by the user during the segmentation process.

Note that the first two components (boundary representation and model) define the domain of objects that can be segmented with the method, while the remaining three determine the behaviour of deformation and the flexibility of configuration to different segmentation problems.

Different choices for the implementation of each component originated many existing flavours of DM - see [22] for a review. As a general rule, these methods address specific needs imposed by the applications that motivated their development, forming a scenario with spread solutions that cannot be combined easily to address segmentation problems in a more generic context.

### 4.2 Requirements Posed on Deformable Model Methods

The following requirements are posed on the basic components of a DM to support the implementation of methods applicable to a large variety of segmentation problems and that allow for enhancement based on information derived from user interaction:

1. **Representation of the boundary geometry.** For a generic method, it is obvious to impose the following requirements:

   - **Requirement #1:** Allow for open and closed boundaries.
   - **Requirement #2:** Allow for free-form continuous boundaries, i.e., the boundary can be represented by any curve in the image grid.
   - **Requirement #3:** Allow for disconnected components, i.e., the boundary of interest can be represented as a set of disconnected parts.

2. **Boundary model.** The model must be flexible and complete to describe the boundary of objects in images obtained using different imaging techniques. The goal here is to be capable of describing prior knowledge about how the objects are ideally represented in the image in terms of image and shape features, as well as knowledge derived from user interaction.
Specifically, a boundary model $M$ should address the following demands:

- **Requirement #4:** *Provide a varied repertoire of local features.* The method should support a wide repertoire of image and shape features $F(t)$ to be used in the description of the boundary model:

$$F(t) \in \{F_1(t), F_2(t), F_3(t), \ldots\}, \quad (4.4)$$

where $F_i(t)$ is a function measuring the value of some local property at the boundary position $t$, e.g. boundary curvature and magnitude of the image intensity gradient.

- **Requirement #5:** *Allow for the expression of expected local feature values and their variation in an ensemble of allowed segmentation solutions.* The measure of deviation from the model should account for variation among possible segmentation results observed in sample data or determined from prior knowledge, such that:

$$|F(t) - \hat{F}(t')| \leq \Delta(t'), \quad (4.5)$$

where $F(t)$ is the measured value, $\hat{F}(.)$ is the expected value (e.g. mean) and $\Delta(.)$ is the tolerance (e.g. standard deviation). The path parameters $t$ and $t'$ correspond respectively to the boundary in the image and the model.

- **Requirement #6:** *Allow for heterogeneous boundary models.* Enable different parts of the boundary model to be described in terms of different expected values or types of image- and shape-based features $F_i(.)$.

### 3.4. Objective Function and Optimisation

To support on-line corrections resulting from interaction, the behaviour of deformation must be controllable, allowing for the modification of parameters such as the geometry of the initial curve $C_0$, the boundary model $M$ (local feature types and interval of allowed values) and the weights of terms in the objective function $\Theta$. We refer to the parameters and result obtained after correction as $C_0^*, M^*, \Theta^*$ and $C^*$. Besides allowing for corrections, it is also important to estimate their impact on $C^*$ as compared to the result $C$ obtained otherwise.

Specifically, the following demands are posed on the deformation behaviour resulting from the objective function optimisation:

- **Requirement #7:** *Enable local control of corrections.* Limit the domain of influence of corrections, such that modification of one boundary part does not affect substantially other portions that are already correct. The curve remains the same outside the interval of correction:

$$C^*(t) \simeq C(t), \forall t \notin [t_1, t_2], \quad (4.6)$$

where $[t_1, t_2]$ is the curve interval where the modification applies.
• **Requirement #8:** Display predictable behaviour in response to the optimisation of a given objective function \( \Theta \). It should be possible to predict the direction of boundary displacement from the initial curve:

\[
\hat{v}(t) \approx \frac{C(t) - C_0(t)}{\|C(t) - C_0(t)\|}
\]  

(4.7)

where \( \hat{v}(t) \) is the estimated direction of displacement for boundary position \( t \). Additionally, small corrections in the initial curve in a given direction should have similar impact on the resulting curve, i.e.:

\[
\frac{C^*_0(t) - C_0(t)}{\|C^*_0(t) - C_0(t)\|} = \vec{v}_0(t) \Rightarrow \frac{C^*(t) - C(t)}{\|C^*(t) - C(t)\|} \approx \vec{v}_0(t), \forall t \in [t_1, t_2].
\]  

(4.8)

where \( \vec{v}_0 \) is the direction of modification in the initial curve at \( t \) and \([t_1, t_2]\) is the interval where the modification applies.

5. **Initialisation** To address varied segmentation problems and to support new knowledge introduced as a consequence of user interaction, the following demand is posed on the DM:

• **Requirement #9:** Support easy configuration of the boundary characteristics in terms of geometry and local features in the model.

<table>
<thead>
<tr>
<th>Component</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boundary</td>
<td></td>
</tr>
<tr>
<td>Geometry</td>
<td>1 Open and closed boundaries</td>
</tr>
<tr>
<td></td>
<td>2 Free-form boundaries</td>
</tr>
<tr>
<td></td>
<td>3 Disconnected components</td>
</tr>
<tr>
<td>Boundary</td>
<td></td>
</tr>
<tr>
<td>Model</td>
<td>4 Varied repertoire of local features</td>
</tr>
<tr>
<td></td>
<td>5 Define expected feature values</td>
</tr>
<tr>
<td></td>
<td>6 Heterogeneous model</td>
</tr>
<tr>
<td>Objective Function and Optimisation</td>
<td>7 Local control</td>
</tr>
<tr>
<td>Initialisation</td>
<td>8 Predictable deformation behaviour</td>
</tr>
<tr>
<td></td>
<td>9 Easily configurable</td>
</tr>
</tbody>
</table>

Table 4.1: Summary of requirements posed on the basic components of a DM.

Table 4.1 summarises the requirements posed on a DM to support the implementation of a controllable and configurable segmentation method. Apart from these nine requirements, interactive segmentation imposes additional demands on the DM as a whole. To support real-time feedback for efficient user-computer interface, the DM should be fast enough to provide results at interactive response time and it should provide visual feedback about the behaviour of deformation in an intuitive way.
From the wide spectrum of existing methods, the following ones have been considered for the implementation of interactive segmentation: [12], [23], [4], [24], [15], [42], [38], [47], [33], [45], [14], [34], [41], [32], [2], [19], [17], [40], [28], [49], [20], [43], [44], [3], [26], [46], [25], [48], [37], [39], [18], [30] and [10]. This list is not complete, but it is representative of the DM capabilities that are relevant for our purpose.

This study showed that, in most cited methods, the boundary model is homogeneous and cannot be easily edited during the segmentation session, the boundary representation is limited to a small class of objects and local control of the objective function is not possible. Moreover, the image and shape features cannot be configured easily for a new segmentation problem and the deformation behaviour is neither intuitive nor predictable, since the tuning of parameters is often mentioned as a difficult task. Exceptions to this general rule are the methods presented in tab.4.2.

Table 4.2: Selection of promising DM to support controllable and configurable interactive segmentation methods. "+" indicates that the method fulfils the corresponding requirement.

<table>
<thead>
<tr>
<th>Method</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Deformable templates [47]</td>
<td>+</td>
</tr>
<tr>
<td>PDM [40]</td>
<td>+</td>
</tr>
<tr>
<td>Grammar-based [28]</td>
<td></td>
</tr>
<tr>
<td>BASOC [20]</td>
<td></td>
</tr>
<tr>
<td>Ziplock snakes [25]</td>
<td></td>
</tr>
<tr>
<td>IIS ([37], [27])</td>
<td>+</td>
</tr>
<tr>
<td>Coupled surfaces [18]</td>
<td>+</td>
</tr>
<tr>
<td>Necklaces [10]</td>
<td>+</td>
</tr>
</tbody>
</table>

Table 4.2 is not absolutely precise because it tries to express the fulfilment to requirements in binary terms. This leads to situations where the fulfilment is partial, and the method gets a "+", while in others the fulfilment could be achieved with minor modifications in the method, but it nevertheless gets nothing. However, the table provides a general impression supporting the conclusion that none of the studied methods fulfils all requirements for interactive segmentation simultaneously. A more generic and flexible formulation is therefore needed to combine the individual strengths of isolated methods into a more powerful segmentation solution.

4.3 A New Method: Piecewise DM

The new method addresses the requirements above on the basis of simple, generic and flexible concepts.
4.3.1 Representation of the Boundary Geometry

The boundary geometry $C$ is represented by a cubic B-Spline curve – see a complete definition in [8] and [29].

Points on a B-Spline curve are computed as a weighted sum of geometric coefficients or control points $P_j$. The set of all control points, the control polygon \( \{ P_1, P_2, \ldots, P_Q \} \), defines the curve geometry as follows - see figure 4.1-a:

$$ C(t) = [x(t), y(t)] = \left[ \sum_{j=1}^{Q} X_j B_j(t), \sum_{j=1}^{Q} Y_j B_j(t) \right], $$

where $Q$ is the number of control points, $[X_j, Y_j]$ is the position of the control point $P_j$ in $\mathbb{R}^2$, and $B_j(t)$ is its weight, determined by the value of the corresponding basis function at path position $t$.

Basis functions are implemented as piecewise polynomials of an arbitrary degree $d$ with local support:

$$ B_j(t) \neq 0 \iff t \in [t_1, t_2], $$

where $[t_1, t_2]$ defines a curve segment composed of $d+1$ intervals determined by a set of knots $\{ k_1, k_2, \ldots, k_n \}$. Knots are points in the path parameter interval relating to the control points; the set of all knots defines the parameterisation and the distribution of basis functions along the curve.

![Original CP](image1) ![Modified CP](image2)

(a) (b)

Figure 4.1: Manipulation of a cubic B-Spline curve. (a) The original control polygon (CP) with four vertices, a modified CP with two translated vertices and the corresponding B-Spline curves. The whole curve changes. (b) The original CP, a modified CP with an additional vertex and the corresponding curves. The curve is affected only in the first quadrant.

B-Splines display the following properties complying with the requirements formulated in section 4.2:

- B-Splines can represent open and closed free-form curves.
• The smoothness of the curve can be prescribed by special choices of the B-Spline (degree and distribution of knots).

• Given a distribution of knots and the corresponding basis functions, the curve is entirely defined by the control points $P_j$, thus the manipulation of the curve can be expressed in terms of the control polygon and vice-versa, with predictable behaviour. See figure 4.1-a for an illustration where the curve is “attracted” to the translated control points, as expected.

• The influence of a given control point is limited to a well-defined curve interval, providing local control to curve manipulations.

• It is straightforward to control the degrees of freedom locally for curve deformation by inserting or removing knots at desired locations, e.g. figure 4.1-b.

In conclusion, B-Splines allow for open and closed free-form boundaries with local control and predictable behaviour, addressing requirements #1, #2, #7 and #8. Furthermore, they allow for performance compatible with interaction, since the number of control points that influence the curve at $t$ is limited to $d+1$, and does not depend on the total number of control points used to represent the curve.

### 4.3.2 Boundary Model

The boundary model $M$ contains prior knowledge about the object of interest in terms of basic characteristics of the curve and the expected value of local features. The basic curve characteristics define the path $t$ and special positions used as references for the representation of heterogeneous knowledge (landmarks). The curve segments defined by landmarks are called pieces, which correspond to boundary parts characterised by different image and shape features. For each piece $k$, the local model $M_k$ determines the curve interval where the piece is defined and the type, expected values and relative importance of the local features in an ideal situation.

Local features $F(t)$ measure the shape or image properties in the neighbourhood of a boundary position $C(t)$. In the case of image features, the measurement is performed in the image grid $[x(t), y(t)]$, in a neighbourhood of given size $\sigma$. The types of features used in the model can be chosen from the repertoire described below.

To measure local shape, two curvature-based features are currently available: curvature $F_k(t)$ and the change of the turning angle $F_{\varphi'}(t)$ [11]:

\[
F_k(t) = \frac{x'(t)y''(t) - x''(t)y'(t)}{(x'(t)^2 + y'(t)^2)^{\frac{3}{2}}}, \quad (4.11)
\]

\[
F_{\varphi'}(t) = \frac{x'(t)y''(t) - x''(t)y'(t)}{x'(t)^2 + y'(t)^2}. \quad (4.12)
\]

where $x'(t)$ and $x''(t)$ are the first and second derivatives of the $x$ coordinate and
likewise for \( y \). In a B-Spline curve, \( x'(t) \) is defined by:

\[
x'(t) = \sum_{j=1}^{Q} x_j B_j^1(t),
\]

where \( B_j^1(t) \) is known \[5\], and similarly for \( y \) and derivatives of higher order.

The preference for curvature-based measures is due to their rotation and location invariance and their power to describe shape \[7\]. In addition, \( F_{yy}'(t) \) is also invariant to the curve length, and as such more suited for shape description in a normalised boundary model. Note that, in contrast with the classic snake, no elasticity term is needed, since it is implicitly minimised in B-Spline curves.

Image-based features measure visual evidence of an object, i.e. any type of information derived from the image data that can indicate the presence of boundaries or regions corresponding to objects in the image. In the Piecewise DM, detectors of local image structure such as edges, ridges, and corners are used to measure image features. These detectors are filters \( \Phi \) that operate on the grey image \( I(x, y) \) to produce another image \( D(x, y) \) indicating the presence of a given image structure at each grid position. The filters simulate early vision operators based on the linear scale-space theory \[16\], where normalisation to scale is adopted to allow the combination of responses obtained at different apertures - see examples in tab.4.3. We consider that the choice between a rotation invariant detector (e.g. \( \mid \nabla I \mid \)) and an orientation-dependent detector (e.g. \( \mid \nabla_y I \mid \)) depends on the application, thus both types are available in Piecewise DM.

<table>
<thead>
<tr>
<th>Visual Evidence</th>
<th>Image Structure</th>
<th>Filter ( \Phi[\sigma] )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bright-to-dark edge ( (F_{\nabla I}) )</td>
<td>Gradient</td>
<td>( \sigma \sqrt{I_x^2 + I_y^2} )</td>
</tr>
<tr>
<td>Horizontal edge ( (F_{\nabla x I}) )</td>
<td>Magntitude</td>
<td>( \sigma</td>
</tr>
<tr>
<td>Vertical edge ( (F_{\nabla y I}) )</td>
<td>(1st order)</td>
<td>( \sigma</td>
</tr>
<tr>
<td>Bright line ( (F_{\nabla^2 I}) )</td>
<td>Laplacian</td>
<td>( \sigma^2</td>
</tr>
<tr>
<td>Bright line ( (F_{\kappa I}) )</td>
<td>Curvature</td>
<td>( -\sigma \frac{I_y^2}{\sqrt{I_x^2 + I_y^2}} )</td>
</tr>
<tr>
<td>Horizontal bright line ( (F_{\kappa y I}) )</td>
<td>(2nd order)</td>
<td>( -\sigma \frac{I_{xx}}{\sqrt{I_x^2 + I_y^2}} )</td>
</tr>
<tr>
<td>Vertical bright line ( (F_{\kappa x I}) )</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4.3: Examples of detectors of visual evidence based on local image structure filters described in \[9\]. \( I_x, I_{xx} \) and \( I_{xy} \) denote the partial image derivatives \( \frac{\partial}{\partial x} I, \frac{\partial^2}{\partial x^2} I \) and \( \frac{\partial}{\partial x} \frac{\partial}{\partial y} I \), and \( \sigma \) is the size of the kernel used to compute Gaussian derivatives. Analogous for dark lines.

To allow for generic treatment in the objective function, the dynamic range of
4.3. A New Method: Piecewise DM

\( F_D(t) \) is normalised between [0, 1] and inverted if applicable, with zero meaning maximal response of the detector of visual evidence.

In conclusion, the boundary model adopted in the Piecewise DM provides the necessary support for heterogeneous boundary models with local control based on features chosen from a varied repertoire (requirements #4, #6 and #7). Moreover, the model is easily configurable to different segmentation problems (requirement #9).

4.3.3 Objective Function

The objective function \( \Theta \) in eq.4.3 is redefined to accommodate for a heterogeneous boundary model with pieces \( k \):

\[
\Theta(C) = \int_t \sum_{k=1}^{K} \mathcal{W}_k(t) \mathcal{T}_k(t, C, \mathcal{M}_k) \, dt, \tag{4.14}
\]

where \( \mathcal{W}_k(.) \) is the importance of piece \( k \) in the global model and \( \mathcal{T}_k(.) \) is the local objective function measuring the difference between a curve segment in \( C \) and the local boundary model described by \( \mathcal{M}_k \).

The deviation between the boundary and the model \( \mathcal{T}_k \) is based on the Mahalanobis distance [6] to ensure results normalised to the expected variation for the segmentation problem at hand. The Mahalanobis distance is implemented as follows:

\[
\Delta_{\mathcal{F}}(t) = \left( \frac{F(t) - \hat{F}(t)}{\hat{\Delta}(t)} \right)^2, \tag{4.15}
\]

where \( \hat{F}(t) \) and \( \hat{\Delta}(t) \) correspond to the range of allowed values defined by the local model \( \mathcal{M}_k \), with:

\[
\Delta_{\mathcal{F}}(t) \in [0, 1] \iff |F(t) - \hat{F}(t)| \leq \hat{\Delta}(t). \tag{4.16}
\]

The consequence of such a normalisation is a controllable deviation measure, as required for intuitive and predictable deformation behaviour.

The deviation measure \( \mathcal{T}_k(.) \) is implemented as a weighted sum of terms measuring the distance \( \Delta_{\mathcal{F}}(.) \) for different types of features defined in the boundary piece model \( \mathcal{M}_k \). To accomplish local control, \( \mathcal{T}_k(.) \) is defined in a piecewise fashion:

\[
\mathcal{T}_k(t, C, \mathcal{M}_k) = \begin{cases} 
\sum_{i=1}^{m} w_i(t) \Delta_{\mathcal{F}_i}(t) & : t \in [t_1, t_2], \\
0 & : \text{otherwise},
\end{cases} \tag{4.17}
\]

where \([t_1, t_2] \) is the curve segment where the piece is defined, \( i \) specifies a type of local feature, \( w_i \) is the weight of the respective feature and \( m \) is the number of local features in the local model. These parameters are defined by \( \mathcal{M}_k \).

Two types of weights are used in the composition of \( \Theta \): \( \mathcal{W}_k \) in eq.4.14 regulates the importance of each piece along the boundary and \( w_i \) in eq.4.17 determines the relative importance of local features in the boundary model.
The weights $W_k(t)$ are determined by the extent of boundary pieces described in the model $M$. The following constraints are imposed to guarantee smooth and normalised functions:

$$W_k(t) = \begin{cases} > 0 & : t \in [t_1, t_2], \\ 0 & : \text{otherwise}, \end{cases}$$ (4.18)

$$\sum_{k=1}^{K} W_k(t) = 1,$$ (4.19)

being continuous along the path interval where the piece is defined and differentiable everywhere. Note that $W_k$ regulates the influence of individual pieces in the optimisation process. This mechanism can be used to control the deformation of an individual piece, for example, by switching it off ($W_k(t) = 0$) or by giving maximum priority to it ($0 < W_k(t) < 1$).

As mentioned before, tuning the feature weights $w_i(t)$ is a largely acknowledged problem that reduces the applicability of existing DM to new domains. This problem is reduced in Piecewise DM because the measure of deviation between the boundary and model is normalised by the Mahalanobis distance, with predictable behaviour within a known range of values. As a consequence, the feature weights $w_i(t)$ are determined in an intuitive way, as proportions that comply with the following conditions:

$$w_i(t) \geq 0,$$ (4.20)

$$\sum_{i=1}^{m} w_i(t) = 1.$$ (4.21)

In summary, the objective function used in Piecewise DM, has the necessary conditions to allow for the definition of expected local feature values, local control and predictable behaviour, addressing requirements #5, #7 and #8.

### 4.3.4 Optimisation

The optimisation of $\Theta(C)$ only affects the boundary geometry, by changing the position of the control points and keeping the same path parameterisation. With respect to this optimisation strategy, $\Theta$ is considered as a function of the control polygon, i.e.:

$$\Theta(C) = \Theta(\{P_1, P_2, \ldots, P_Q\}).$$ (4.22)

The parameters for optimisation are therefore the position of control points, representing a reduction of search dimensions as compared to the complete curve.

To guarantee performance compatible with interactive processing, the current implementation uses the conjugate gradient method [31] for optimisation. This method uses information about the gradient of the objective function, which is a vector of $2Q$ components:

$$\nabla \Theta = \left[ \frac{\partial \Theta}{\partial P_1}, \frac{\partial \Theta}{\partial P_2}, \ldots, \frac{\partial \Theta}{\partial P_Q} \right],$$ (4.23)
with:

$$\frac{\partial}{\partial P_j} \Theta = \left[ \frac{\partial}{\partial X_j} \Theta, \frac{\partial}{\partial Y_j} \Theta \right]. \quad (4.24)$$

Note that \( \frac{\partial}{\partial P_j} \Theta \) must be known explicitly, imposing constraints on the functions used in the composition of \( \Theta \) for which the conjugate gradients method can be applied. This limitation is not considered severe when balanced against the gain in performance: comparison with other non-linear optimisation techniques (Downhill Simplex and Powell) in an experimental set-up showed that, under common circumstances, all provide similar results in terms of convergence and outcome, but conjugate gradients is at least an order of 10 faster.*

Since \( V \Theta \) points in the direction of the steepest descent of the objective function, it also serves to predict the expected deformation behaviour by the analysis of deformation forces \( \vec{V}_j \) acting on the control polygon vertices. These vectors correspond to the preferred motion of control points \( P_j \) as a consequence of the optimisation process, defined as follows:

$$\vec{V}_j = \vec{V}_j \left[ \frac{\partial}{\partial X_j} \Theta, \frac{\partial}{\partial Y_j} \Theta \right]. \quad (4.25)$$

The magnitude of deformation forces \( ||\vec{V}_j|| \) indicate the amount of motion expected for \( P_j \), while their orientations \( \frac{\vec{V}_j}{||\vec{V}_j||} \) show the expected direction of motion for a normalised change in \( P_j \). The forces can be separated into components referring to different pieces or types of local features described in the boundary model, indicating how the model constraints influence the boundary deformation individually. At last, the definition of deformation forces at boundary positions \( \vec{V}(t) \) is a straightforward consequence of the B-Spline definition in eq.4.9. In all cases above, the analysis of deformation forces provides useful information about the expected local boundary deformation, permitting a rough prediction of the result obtained after optimisation. When displayed along the boundary, they provide a valuable and intuitive visual feedback mechanism for interactive segmentation.

In conclusion, the deformation resulting from optimisation is local and predictable, fulfilling requirements #7 and #8. Furthermore, it admits interactive processing in terms of performance and visual feedback for the user.

### 4.3.5 Initialisation

The initial curve \( C_0 \) and the objective function \( \Theta \) are built during initialisation based on a curve provided by the user and the information described in a given boundary model \( M \).

---

*We are aware of limitations of the conjugate gradients method for the optimisation of non-convex functions [13], as is typically the case of the image-derived term, because the boundary can be easily trapped into local minima. Since we consider an interactive set-up, we leave the solution of this problem to the user, who can easily move the curve out of it, rather than including extra forces or parameters (e.g. [4], [48]) that could make the DM more complex and less controllable.
The initial curve is created as follows: an elastic template is displayed on the screen in a preferred size and position, and the user adjusts it to the object in the image by manipulating the vertices with the mouse. The adjusted geometry is used to create $C_0$ with local cubic curve interpolation \cite{29}. The new curve is built based on the basic curve characteristics described in the model, such that the path positions of landmarks in $C_0$ are known. The distribution of control points along the curve is based on heuristics about the expected curve and the degrees of freedom needed to represent it. As the last step in the initialisation process, the terms in $\Theta$ are configured based on the local knowledge about the boundary pieces $M_k$ defined in the model.

This initialisation procedure is generic and flexible, supporting on-line configuration of the deformable model (requirement #9).

### 4.4 An Example of Piecewise DM

As an illustration, we present how Piecewise DM was applied to segment the joint space in osteoarthritic ankles - see chapter 6 for more details.

The space between the tibia and the talus at the ankle joint is delineated by two open curves in X-ray images, as illustrated in figure 4.2. Due to faint visual evidence of the ankle joint space boundary\footnote{The boundary is faint to the degree that printing the image on paper typically destroys the visual evidence.}, this application requires an interactive solution. A method based on the structured approach suggested in chapter 3 was developed, combining a Piecewise DM with interaction such that the boundary model is modified (or “corrected”) on the basis of information provided by the user.

![Figure 4.2](image.jpg)

Figure 4.2: (a) Digitalised X-ray image of a normal ankle - courtesy of the Image Sciences Institute, University Medical Center Utrecht. (b) Scheme showing the boundaries of interest (plain lines), the joint space (shaded area), and misleading boundaries (dotted lines).

The upper and lower boundaries of the ankle joint space are implemented as two independent Piecewise DM illustrated in figure 4.3. For both boundaries, the model is composed of five pieces with different image and shape properties summarised in tab.4.4. The image intensity profile at the upper and the lower boundaries correspond
Figure 4.3: Pieces and corresponding landmarks (circles) in the boundary model for the ankle joint space.

respectively to a bright line and a step-edge. Detectors of horizontal features are used for the central pieces to obtain a stronger response under faint visual evidence, and the scale \( \omega \) is automatically determined based on the size of the region of interest indicated by the user with a rectangle. The change of the turning angle \( (F_{\varphi'}) \) is used to describe shape. Three pieces are roughly straight stretches connected by two corners with high curvature.

<table>
<thead>
<tr>
<th>Boundary Piece</th>
<th>Upper Boundary</th>
<th>Lower Boundary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lateral stretch</td>
<td>( F_{V2,I} ) 0.8 0±30</td>
<td>( F_{V,I} ) 0.8 0±30</td>
</tr>
<tr>
<td>Lateral corner</td>
<td>( F_{V2,I} ) 0.9 -20±80</td>
<td>( F_{V,I} ) 0.9 25±80</td>
</tr>
<tr>
<td>Central stretch</td>
<td>( F_{V2,I} ) 0.7 0±20</td>
<td>( F_{V2,I} ) 0.7 25±20</td>
</tr>
<tr>
<td>Medial corner</td>
<td>( F_{V2,I} ) 0.9 15±80</td>
<td>( F_{V,I} ) 0.9 25±80</td>
</tr>
<tr>
<td>Medial stretch</td>
<td>( F_{V2,I} ) 0.8 0±30</td>
<td>( F_{V,I} ) 0.8 0±30</td>
</tr>
</tbody>
</table>

Table 4.4: Boundary model for the upper and lower boundaries of the ankle joint space, showing the type of image feature adopted for each piece, the weight of the image constraint \( (w_D) \), and the expected value and tolerance for the shape feature \( (F_{\varphi'}, \Delta_{\varphi'}) \), with \( w_{\varphi'} = 1 - w_D \).

A segmentation session is illustrated in figure 4.4. The user initialises the process by adjusting a template to the image with the mouse. The initial curve is built based on the adjusted template and the prescription for the curve characteristics defined in the boundary model. The objective function is configured based on the parameters defined by the model, using the position of the template vertices as landmarks to define the pieces. The initial curve and deformation forces are presented to the user on the screen. If the forces are correct, the curve is deformed to obtain the final segmentation result. Otherwise, i.e. when the deformation forces do not point toward the desired contour, interactive correction of the initial curve or the boundary model is necessary.
Figure 4.4: Example of an interactive session for the segmentation of the lower boundary of the ankle joint space. (a) Template adjusted to the image. (b) Initial curve and deformation forces. Images: courtesy of the Image Sciences Institute, University Medical Center Utrecht.

Figure 4.5 presents examples of results obtained with minimal user intervention, where no model corrections were necessary. This was possible because the model in the Piecewise DM is heterogeneous, exploring local knowledge about the boundary and increasing the rate of success of the automatic method. Note that the same boundary model was used in both cases, in spite of the different boundary appearance in these images.

Figure 4.5: Segmentation results obtained with the default boundary model. Images: courtesy of the Image Sciences Institute, University Medical Center Utrecht.

Figure 4.6 presents an example where user interaction was needed to correct the model for the central piece of the upper boundary, where the image intensity profile was different from expected: instead of a bright line, the boundary was located at a step-edge. The model for the central piece was modified, adopting $F_{VI}$ as local image feature. Note that modifications in the boundary model are simple operations in Piecewise DM due to local control, predictability and flexible model representation.
4.5 Conclusions

Our work starts with the isolation of the main components of deformable models, bringing some structure to scattered contributions.

Based on this general structure, we formulate requirements aiming at a generic and locally controllable method. The requirements are: allow for open and closed free-form contours with disconnected components; build the boundary model based on varied and heterogeneous features with predefined expected values; provide for local control and predictable behaviour of deformation; and enable easy configuration. These demands are a natural response to the large diversity of segmentation problems found in the real world, as well as to the condition frequently seen in practice where implicit and homogeneous assumptions about the boundary do not hold. Methods fulfilling such requirements are expected to be useful not only in the context of interactive segmentation, but also in a broader domain of applications which need robust and efficient segmentation solutions. The study presented here concluded that none of the existing methods found in the literature fulfils the listed requirements completely.

A new method was developed, extending the class of deformable models with a generic and locally controllable method. As we have argued at all stages of the description, the Piecewise DM method complies with all the requirements except for one, #3 (disconnected components). Note, however, that this limitation could be lifted by allowing for a discontinuous supporting curve and letting the boundary pieces refer to different curves, such as in [18].

Finally, Piecewise DM is based on a design scheme where it is relatively simple to customise the boundary model for a specific application. We demonstrated how the new method works to support an interactive segmentation solution for the ankle joint space, where the boundary model is heterogeneous and allows for on-line corrections on the basis of information obtained interactively. This application is an example where Piecewise DM was used to provide a simple, but practical and efficient solution.
for a difficult segmentation problem that could not be easily solved with existing methods.

Acknowledgements. Thanks to Anne Karien Marijnissen (Department of Rheumatology & Clinical Immunology) and Koen Vincken (Image Sciences Institute), University Medical Center Utrecht, The Netherlands, for collaboration in the development of the sample application, and to Julia Schnabel, for her comments to improve this chapter.

Bibliography


