Chapter 4

Probabilistic image segmentation

Abstract
A method is presented to segment multi-dimensional images using a multiscale (hyperstack) approach with probabilistic linking. A hyperstack is a voxel-based multiscale data structure whose levels are constructed by convolving the original image with a Gaussian kernel of increasing width. Between voxels at adjacent scale levels, child-parent linkages are established according to a model-directed linkage scheme. In the resulting tree-like data structure roots are formed to indicate the most plausible locations in scale space where segments in the original image are represented by a single voxel. The final segmentation is obtained by tracing back the linkages for all roots.

The present chapter deals with probabilistic or multi-parent linking, i.e., a setup in which a child voxel can be linked to more than one parent voxel. The multi-parent linkage structure is translated into a list of probabilities that are indicative of which voxels are partial volume voxels and to which extent. The output of the thus constructed hyperstack can be directly related to the opacities used in volume renderers. It is demonstrated by means of artificial images as well as real-world (medical) images that probabilistic linking gives a significantly improved segmentation as compared with conventional (single-parent) linking.

Furthermore, probability maps are generated to visualize the progress of weak linkages in scale space when going from fine to coarser scale. This is shown to be a valuable tool for the detection of voxels that are difficult to segment properly.

Keywords: Image segmentation, multiscale analysis, scale space, probability maps, partial volume artifact, data structures.

4.1 Introduction
Segmentation of volumetric image data plays a crucial role in image processing, in particular as a preprocessing step for quantitative analysis and volume visualization.
In the last decade, multiscale approaches (like pyramid \cite{16, 24, 23, 73, 11}, stack \cite{54, 86, 64} and wavelet \cite{40, 21, 70}) segmentation have gained considerable attention. Recently, we have developed a flexible data structure—the hyperstack—for multiscale segmentation of two- and three-dimensional (2D, 3D) images \cite{116, 118, 25}, which admits of extensions like outer scale reduction and probabilistic linking. Outer scale reduction—reducing the number of voxels as the scale increases—will speed up the process of building a hyperstack; this subject will be treated in a Chapter 5. Probabilistic linking—the subject of the present chapter—is introduced with the aim of improving the tuning of the segmentation to the subsequent rendering of volumetric structures. In this way the jaggedness of objects which were incorrectly segmented by a binary decision procedure, is sought to be reduced.

We first briefly discuss the design of the conventional (single-parent) hyperstack, which is characterized by the fact that a voxel at some level of the hyperstack is connected to at most one (parent) voxel in the next higher layer. We then discuss probabilistic (multi-parent) hyperstacks and pay special attention to the explosive growth of the number of linkages if no precautions are taken. It will be shown that this problem can be solved without significantly affecting the quality of the segmentation.

Finally, we show how conventional-like segmentations can be derived from the output of probabilistic hyperstacks, and why these segmentations are more robust and of better quality than those obtained by conventional (single-parent) hyperstacks. We will also present some segmentation results of probabilistic hyperstacks to show the surplus value of multi-parent linking over single-parent linking.

### 4.2 The conventional hyperstack

In the sequel we use the term level to denote an image at a specific scale. The original image (or ground level) corresponds with level 0, the top level—representing the most strongly blurred image—is at level \( L - 1 \). A hyperstack thus contains \( L \) levels at increasing scale. The terminology in this chapter will generally apply to 3D images, so we will call the elements of the hyperstack voxels rather than pixels. The set-up is equally valid for 2D images, however.

Building and employing a hyperstack consists of four different steps (see Fig. 4.1):

1. **Blurring.** Images at increasingly larger scales are obtained by convolving the original image with Gaussians of increasing width.

2. **Linking.** Child-parent linkages are established between voxels at adjacent scale levels.

3. **Root labeling.** Voxels having weak parent linkages and all voxels in the top-most level are marked as roots.

4. **Downward projection.** The original image is segmented by tracing back the linkages from the roots to the ground level.
4.2 The conventional hyperstack

The following subsections deal with these steps in detail.

### 4.2.1 Blurring

The construction of a continuous linear scale space follows a blurring strategy, which is essentially a repeated low-pass filtering using a Gaussian kernel [131, 54]:

\[
L(\vec{x}, \sigma_0 \oplus \sigma) = L_0(\vec{x}, \sigma_0) \ast G(\vec{x}, \sigma),
\]

where \( L_0(\vec{x}, \sigma_0) \) is the luminance or intensity of the original image, \( G(\vec{x}, \sigma) \) is the Gaussian with width \( \sigma \) of corresponding dimension, and \( L(\vec{x}, \sigma_0 \oplus \sigma) \) is the \( \sigma \)-blurred replica of the input image. Note that the additive operator “\( \oplus \)” does not correspond to ordinary addition, but follows the semi-group property: \( \sigma_0 \oplus \sigma = \sqrt{\sigma_0^2 + \sigma^2} \).

A discrete scale space is best constructed by convolution of the original image with sampled Gaussian kernels of increasing width. The alternative—building the scale space by convolving level \( i \) with a Gaussian to obtain level \( n+1 \) for \( n = 0, \ldots, L-1 \)—has the disadvantage that a concatenation of sampled Gaussians is not a discrete scale space transformation [66].

As for scale space sampling, an equidistant sampling of the absolute scale \( \sigma \) would violate the important property of scale invariance [33]. Instead, the sampling should follow a linear and dimensionless scale parameter \( \delta \tau \), which is related to \( \sigma \) by:

\[
\sigma_n = \varepsilon e^{\tau_0 + n \delta \tau}, n \in \mathbb{N}.
\]

In this equation \( \varepsilon \) is taken to be the smallest linear grid measure of the imaging device. A convenient choice for \( \tau_0 \) is 0, which implies that the inner scale \( \sigma_0 \) of the initial image is taken to be equal to the linear grid measure \( \varepsilon \).
4.2.2 Linking

In the linking step, voxels may be connected to voxels in the next-higher level (child-parent linking). We discriminate between active and passive voxels. Passive voxels do not participate in the linking process. At the ground level, passive voxels are voxels without parents; they may be introduced to decrease the number of voxels to be processed, notably to eliminate uninteresting background voxels. At the higher levels passive voxels are generated automatically: all voxels that have not been chosen as a parent of at least one child are considered passive.

In the conventional hyperstack [116, 118, 25], every child is linked to exactly one parent. In the probabilistic hyperstack [121, 123, 124], a child voxel can have more than one parent (see section 4.3.1).

Research has shown that a general and robust linking scheme should consist of three components [58]: (i) the intensity difference between a child and its parent, (ii) the ground volume of a parent, and (iii) the mean ground volume intensity. See section 2.2.2 for details.

4.2.3 Root labeling

Voxels are labeled as roots after all child-parent linkages have been established. The main reason for avoiding root labeling during the linking process is that we want to have control over number of roots at the segmentation stage. Reducing the number of roots (which implies increasing the number of linkages) after the completion of the hyperstack is not only a laborious task, but also a dubious effort: owing to the ground volume component in the affection formula linkages can not be added without recalculating all the other linkages in that level and higher. Ignoring this fact may very well lead to ‘false’ links, and hence to a lower accuracy of the segmentations in general. Consequently, every voxel in the hyperstack is linked to a parent, except for the voxels in the top level. After the top level has been appended, the roots can be defined.

For a detailed description of the conventional root labeling process, we refer to section 2.2.3. The process of root labeling will become much more complex in the case of probabilistic linking (see section 4.3.3).

4.2.4 Downward projecting

In the last step, downward projection, intensity values that are characteristic of the roots are projected downwards to the ground level. This requires—besides the choice of a lowest root level—the definition of a level from which to start the projection. All parents in this segmentation level are considered roots. A higher segmentation level implies—owing to the fact that the number of parents decreases at larger scales—a smaller lower bound for the minimum number of segments to be found.
4.3 The probabilistic hyperstack

The number of segments and their different sizes are thus influenced by the choice of the lowest root level and the segmentation level, and by the actual root labeling.

In the conventional hyperstack there are three possibilities for calculating a segment value (i.e., the actual intensity value given to all the voxels of one segment): (i) the root intensity, (ii) the mean intensity of the ground voxels, and (iii) a unique (pseudo-)value; for the probabilistic hyperstack a fourth possibility (the probability value) exists, which will be discussed in section 4.5.

Root values are blurred intensities and thus will often produce low-contrast segmentations. (Normally, the global intensity extrema of an image converge approximately linearly towards the average intensity of the largest scale, provided the scale space is sampled according to equation (4.2).) The mean intensities of sets of ground voxels are less dependent on the scales at which the roots reside, which results in a higher contrast in the thus produced segmentations (at the cost of a little more computing time).

In some cases it is desirable to know the contours of segments, e.g., in the case where two adjacent segments with the same intensity (after downward projection) need to be distinguished. In such cases, it is not sufficient to use root- or mean intensity values for the down projection, although they give an acceptable output for visual examination of the results. The distinction of segment values may be accomplished by giving each segment a unique number. The use of pseudo-colored overlays can help visualizing the different segments.

4.3 The probabilistic hyperstack

After having discussed the conventional hyperstack, we now turn to probabilistic hyperstacks in which every child voxel is allowed to connect to more than one parent.

4.3.1 Probabilistic linking

Instead of forcing a binary decision to which parent a child should be linked, we simply link a child to all parents that are ‘good enough’, according to some objective criterion. This minimizes the chance that a crucial link is missed (which may happen if a worse parent is preferred because of noise); higher up in the hyperstack the ‘mistake’ will be corrected automatically. This is an important feature of probabilistic linking. Note that—since the noise will disappear if the scale is increased—the chance that a crucial link is missed owing to noise will be negligible in the higher levels. We will use this observation when discussing the complexity of the hyperstack in detail (see section 4.4).

Once all interesting linkages have been established, the corresponding probabilities are found simply by normalization of the afflection values, since the sum of the linkage probabilities from a child to all of its parents must equal one.
Fig. 4.2. Example of multi-parent linking in a hyperstack.

Fig. 4.2 shows an example of multi-parent linking. Here, and in the sequel of this chapter, we denote by $P(V_{i,l+1} | V_{i,l})$ the conditional probability that voxel $i$ at level $l + 1$ is the parent of child voxel $i$ at level $l$. (Note that a voxel is indexed both by the level index ($l$) and the grid index ($i$). Consequently, voxel $i$ at level $l$ is not necessarily related to voxel $i$ at level $l + 1$.) Child voxel $V_{1_0}$—meaning: the child with index 1 at level 0—is linked to two parents at level 1: $V_{1_1}$ and $V_{2_1}$. The corresponding child-parent probabilities are $P(V_{1_1} | V_{1_0}) = 0.55$, and $P(V_{2_1} | V_{1_0}) = 0.45$. If a child links just to one parent, the child-parent probability takes the value 1.

In appendix 4.A we discuss the consequences for the data structure of extending the hyperstack from single-parent to multi-parent linking, and in appendix 4.B we deal with the implementation of the linkage tree by means of so-called link containers.

### 4.3.2 The ground volume under multi-parent linking

In conventional hyperstacks, the ground volume of a voxel is the number of ground voxels with a route to that voxel. The same definition is not appropriate for probabilistic hyperstacks. Instead, the voxels encompassed in the ground volume are weighted by the probabilities of the linkage paths. For example, the ground volumes of parents $V_{1_2}$, $V_{2_2}$, and $V_{3_2}$ in Fig. 4.2 have values 0.33, 0.8435 and 0.8265, respectively.

### 4.3.3 Root labeling under multi-parent linking

Each of the two paradigms for root labeling—discussed in section 2.2.3 for conventional hyperstacks—must be adapted to multi-parent linking:

- The threshold value on the adultness to determine which voxels have to be labeled as roots must be applied to the strongest link of every child.
A specified number of roots is no longer necessarily identical to the number of segments after downward projection. For instance, if only the highest probability path of a linkage tree is used to segment the image, several roots may end up not having a ground volume. Thus, only an upper bound for the number of segments can be specified with this method.

The eventual root probabilities (obtained as a result of the root labeling) represent the chances for a ground voxel to belong to various segments. In an equivalent interpretation these probabilities represent the fractions of segments that are contained in a ground voxel. The result is comparable to the output of fuzzy image subsets (see [94]) in which a degree of membership is associated with every voxel.

The root probabilities are computed by following all the linkages that connect a ground voxel to different roots. The child-parent probabilities are multiplied for each path, and the ‘path probabilities’ are added to find the final root probability, denoted by \( P(V_i \mid V_{i_0}) \), where \( l \) is the level at which the root is defined.

The root probabilities can be calculated from the recursive relation

\[
P(V_i \mid V_{i_0}) = \sum_{i_{l-1}=1}^{N_{i_{l-1}}} P(V_{i_l} \mid V_{i_{l-1}}) \cdot P(V_{i_{l-1}} \mid V_{i_0}),
\]

with \( 2 \leq l \leq L - 1 \), \( L \geq 3 \), and \( 1 \leq i_k \leq N_k \) for all \( 0 \leq k \leq L - 1 \); \( N_k \) is the number of active voxels at level \( k \). Equation (4.3) can be written as

\[
P(V_i \mid V_{i_0}) = \sum_{i_1=1}^{N_1} \sum_{i_2=1}^{N_2} \cdots \sum_{i_{l-1}=1}^{N_{l-1}} \left\{ \prod_{j=1}^{l} P(V_i \mid V_{i_{j-1}}) \right\} .
\]

Note that the following expression is also valid for all \( 1 \leq i_{l-1} \leq N_{l-1} \) (normalization property):

\[
\sum_{i_l=1}^{N_l} P(V_i \mid V_{i_{l-1}}) = 1
\]

The root probabilities between voxels which are two levels apart can be found from equation (4.4). Using the affections as given in Fig. 4.2, we find

\[
\begin{align*}
P(V_{1_2} \mid V_{1_0}) &= P(V_{1_2} \mid V_{1_1}) \cdot P(V_{1_1} \mid V_{1_0}) = 0.33 \\
P(V_{2_2} \mid V_{1_0}) &= P(V_{2_2} \mid V_{1_1}) \cdot P(V_{1_1} \mid V_{1_0}) + P(V_{2_2} \mid V_{2_1}) \cdot P(V_{2_1} \mid V_{1_0}) = 0.4135 \\
P(V_{3_2} \mid V_{1_0}) &= P(V_{3_2} \mid V_{2_1}) \cdot P(V_{2_1} \mid V_{1_0}) = 0.2565
\end{align*}
\]

From Fig. 4.2 it follows that segmentations obtained using the strongest linkage path in multi-parent linking may differ from segmentations obtained with single-parent linking. In the latter, only the parent with the locally highest affection is included in the data structure, which would make \( V_{1_2} \) the grandparent of \( V_{1_0} \). Multi-parent linking identifies two voxels, \( V_{1_2} \) and \( V_{2_2} \), as possible grandparents. If only the strongest linkage path is considered, voxel \( V_{2_2} \) is the most probable grand parent of \( V_{1_0} \).
The output of a probabilistic hyperstack is a list of normalized probabilities per (ground) voxel. These probabilities represent the chances that a voxel belongs to the listed segments. To obtain actual segmentations from lists of probabilities, as provided by probabilistic hyperstacks, we have several possibilities. In section 4.5 some methods are discussed and applied to actual images, while also the most important advantages of probabilistic over conventional segmentations will be summarized.

### 4.3.4 Probability maps

Before actually applying a probabilistic hyperstack, we first want to understand how linkages evolve in scale space. The number of linkages is enormous, which makes it hard to follow and evaluate each of them. Therefore, we developed a way to visualize the progression of weak and strong linkages.

The idea is to produce so-called *probability maps* by displaying for every ground voxel the largest probability that it belongs to a root in a specific level. This simple algorithm first chooses a level $l$ to create a map for, then turns all the active voxels in level $l$ into roots, and finally calculates the highest root probabilities for every ground voxel by scanning the entire hyperstack. A linear remapping of these probabilities onto a range of regular image intensities, produces the desired result: dark voxels represent areas of low probability, whereas brighter voxels correspond to high probabilities.

It might be expected that edge voxels are harder to link than voxels in near-homogeneous volumes, for two reasons:

- **Partial volume effect.** An edge voxel will have, apart from possible noise components, a value proportional to the volumes of the different object types that are represented in that voxel. Thus, statistically the value of an edge voxel will lie in between the values of the surrounding object values. Since the intensity difference in equation (2.12) does not explicitly prefer smaller or larger values, there will not be a preference for either of the objects.

- **Blurring strategy.** The blurred value of a voxel is determined by a weighted sum of the values of the voxel at hand and the values of its neighbors. Thus, edge voxels will be subjected to larger intensity changes in scale space than non-edge voxels.

Hence, it is plausible that voxels near object edges will generally have links to more parent voxels than voxels in homogeneous areas.

In Fig. 4.3 a series of probability maps in scale space is shown for the HEAD image, a sagittal MR image of the brain. One can clearly see the distribution of the largest probabilities evolving in the hyperstack. The edges of the objects of different sizes are best visible at the scale at which they can be represented by a single root. Logically, there is a high correspondence between these levels and the levels found in edge detection algorithms based on scale space [17]. For instance, the edges of the
Fig. 4.3. Probability maps of a hyperstack based on 15 blurred levels of the two-dimensional HEAD image. Shown are: the original MR image (with dimension sizes 256 × 256), five probability maps (that correspond with level 4, 6, 10, 14 and 15 of the hyperstack, respectively), and finally two coarse segmentations (containing a predefined number of 5 and 2 segments, respectively). Note the similarity between the last two probability maps and the two segmented images (see text).
ventricle are noticeable at the middle levels (cf. $\sigma = 32$), but merge with other objects at the higher levels.

At one but the highest level shown ($\sigma = 128$) the probability map contains five separate fragments: one for the head object and four for the background. The reason for the side contours to be on the top, at the bottom, at the left and at the very right of the image is that those pixels have great difficulty choosing between two near background parts. At a higher level still ($\sigma = 181$), the background merges into a single segment. A segmentation made with a predefined number of segments that matches the number of segments in a probability map, bears a close resemblance with this map, as the lower frames of Fig. 4.3 show convincingly.

![Fig. 4.4. Volume renderings of 3D probability maps of the artificial ellipsoid image. Shown are a rendering of the original image (top picture), followed by 6 renderings of inverted probability maps (for an explanation see text).](image)

In 3D a probability map is generated in the same way as for the 2D case. Visualization of the result in a slice-by-slice manner, however, is not very useful for recognizing edge-like structures. To produce a suitable 3D presentation we invert the
4.4 Computational complexity

The complexity of probabilistic hyperstacks is mainly determined by the total number of linkages; if no precautions are taken, this number grows explosively. Thus, in order to keep the number of linkages limited, we need to introduce some constraints.

We have used two limits while building a probabilistic hyperstack:

1. an upper bound for the number of parent linkages per child
2. a lower bound for the affection, below which no linkages are established (provided that at least one link per active voxel is present)

Since either constraint has great influence on the actual number of linkages formed, we will explain them in more detail below. Note that during the linking process, care must be taken that at least one link per child is established, so as to avoid premature creation of roots. Section 4.3 presents some results on the number of linkages established with and without the constraints.

Another factor that influences the computational complexity of a hyperstack, though to a lesser extent, is scale space sampling. Using only a few levels will also keep the amount of linkages low. The minimum scale space sampling rate to avoid deterioration of the quality of the segmentation will be application dependent. Research is due to optimize the sampling rate for classes of applications.

In practice, it turns out that it is not necessary to blur until a scale is reached where only one intensity is left (an average image intensity value). The number of levels used will generally lie between 10 and 20.

4.4.1 A maximum number of parents per child

Owing to noise present at the smaller scales, the maximum number of parent linkages per child should be relatively high when linking the lowest levels, and may decrease when linking levels at larger scales. A similar argument holds for the partial volume voxels in the lower levels (see section 4.3.4 about the probability maps). Thus, we need to allow the lowest levels more links than those higher up in scale space.
Chapter 4. Probabilistic image segmentation

If we try to determine the maximum number of parents for every child voxel at the ground level without a priori information on the image to be segmented, the number of dimensions plus one seems a reasonable choice. This choice is motivated by the partial volume notion and is based on the fact that it is unlikely that on a two-dimensional map four different areas join together in one point, whereas 3-junctions (i.e., three joining areas) are much more common. Similar considerations lead to a maximum of four in the three-dimensional case. The diminishing influence of both the noise component and the partial volume effect at increasing scale make it acceptable to decrease the maximum number of parents per child with one at every scale step. This decrease is effectuated after level 1 has linked to level 2.

4.4.2 A lower bound for the affection

The linkage strength increases when linking at larger scales, for ground volumes are steadily growing and the intensity differences will diminish. Consequently, the lower bound for the affection should increase accordingly.

When searching for a suitable lower bound for the affection of linkages, it is attractive to define a uniform lower bound per level. A simple calculation—for instance based on the affection range of the first iteration—can then serve as initial estimation for the lower bound. Moreover, the need to recalculate the limit per child, or even per iteration, is absent.

Implementation of this paradigm, however, showed an annoying side-effect: owing to the fact that between any two levels there is a rather high variance in affection values (especially at smaller scales), voxels in areas with a relatively low affection range (edges) are assigned one single link, the minimum amount. But probabilistic linking has been invented to give better segmentation results precisely in those areas! Consequently, we dropped this ‘solution’.

Much better results are obtained by defining an adaptive lower bound of the affection, despite the amount of work involved. The idea is to define a parameter, called the minimum relative affection, that specifies how much the affection of a child-parent link may differ from the highest affection present for that child. In order to avoid absolute differences here, we choose to set a relative value (e.g., 95%). Intuitively, this leads to the desirable effect that voxels with a strong preference for a specific parent do not need any other links, whereas children in dubio are assigned more than one parent.

4.4.3 Results on constrained linking

In order to evaluate the effects of constraining the number of linkages, we built five types of hyperstacks with different parameter settings, and compared the number of linkages evolving in the first 14 levels for:

a. a traditional hyperstack with single-parent linking
b. a multi-parent hyperstack with a minimum relative affection value of 95% and initially a maximum of 3 parents per child

c. a multi-parent hyperstack with a minimum relative affection value of 0% and initially a maximum of 3 parents per child

d. a multi-parent hyperstack with a minimum relative affection value of 95% and continuously a maximum of 3 parents per child

e. a multi-parent hyperstack with a minimum relative affection value of 0% and continuously a maximum of 3 parents per child

Fig. 4.5. Scale dependence of the number of links for five different types of hyperstacks based on the HEAD image. For an explanation of the different types (a) up to (e) see text.

The results for the HEAD image of Fig. 4.3 are presented in Fig. 4.5. The hyperstack has been created according to equation (4.2) with a $\delta \tau$ value of $\frac{1}{2} \ln 2$; this results in a hyperstack of 15 levels.

From Fig. 4.5, it follows that the constraints we impose on the linking procedure are effective. Setting the minimum relative affection to 95% (d) limits the number of linkages in the lower levels in the hyperstack. Using no lower bound (e) results in a significantly higher number of linkages.

Applying the constraint that limits the maximum number of parents per child at increasing scale (c) has even more effect (again compared to (e)): irrespective of the
minimum relative affection used, the hyperstacks quickly converge to single-parent variants (a). Finally, applying both constraints is most useful (b).

4.5 Results

The presentation of the results—lists of tissue probabilities—is hampered by the unavailability of suitable volume rendering software. Yet, we can compare images segmented by multi-parent hyperstacks with conventional segmentations by selecting one segmented image from an ensemble of possible realizations. The natural option is to use the highest object probability of each list. In section 4.3.1 we have indicated why segmentations thus obtained may be expected to outperform segmented images from single-parent hyperstacks.

![Fig. 4.6. Two-dimensional segmentation of the sagittal head image. Contours have been superimposed (in bright white) on the original MR image. Single-parent segmentation (left), multi-parent segmentation (right).](image)

Fig. 4.6 shows a comparison of a segmentation using the single-parent hyperstack with a multi-parent segmentation, in which each voxel is represented by the highest object probability of its list. The arrows emphasize the main differences between the two segmentations. The multi-parent hyperstack clearly performs better than the single-parent hyperstack.

The probabilistic hyperstack used contained 17 levels (with constrained linking and a minimum relative affection of 0.90), while the lowest root level was set to 6. The contours have been found by simple thresholding of the segments formed by downward
projection of mean segment values. We emphasize that *no* additional editing has been performed on the segmentations.

![Segmentation of the cerebellum of the HEAD image. Original image (left), single-parent segmentation (middle) and a probabilistic result (right).](image)

A second way to visualize probabilistic results is to focus on one object (organ or tissue type). The summed probabilities of the paths from a ground voxel to a root are indicative of the probability that this ground voxel belongs to the segment defined by the root. (Note that this value is not equal for all the members of one segment, in contrast with other downward projecting techniques.) The result is an image whose intensities are proportional to the amount of tissue contained in each voxel (see Fig. 4.7). Volumetric compositing methods [27, 63] can be applied to visualize this type of output (see also [87, 9, 139, 138]).

![Renderings of the stylized ELLIPSOID image: original image (left), conventional hyperstack segmentation (middle) and probabilistic hyperstack segmentation (right).](image)

In Fig. 4.8 the same technique has been applied to the ELLIPSOID image. The (noise-free) input image was used to generate the first rendering, in order to show the desired result. The object intensity value is 2000, with a background value of
Chapter 4. Probabilistic image segmentation

The second rendering is based on segmentation by a single-parent hyperstack (after having added Gaussian noise with a standard deviation of 10% of the object intensity), and the third rendering is based on a multi-parent hyperstack. Note the notched edges in the middle image, owing to the single-parent segmentation. The probabilistic linking softens this effect (third image).

Fig. 4.9. Renderings of (a) a single-parent and (b) a probabilistic segmentation of the ventricles image, a 3D MR image of the brain. Including additional segments is very simple (c).

Fig. 4.9 shows a series of results of techniques to segment the ventricles from a three-dimensional MR image of the brain (the VENTRICLES image). On the top row, renderings from different viewpoints are shown based on conventional (single-parent) hyperstack segmentations. On the second row, the probabilistic counterparts are shown. The third row also represents a probabilistic hyperstack, but now extended with two small additional segments in the middle of the ventricle. To obtain this result, we only had to increase the desired number of segments with two. Note that the single-parent segmentation has several serious shortcomings.
4.6 Conclusions and discussion

We have presented a method to segment images in a probabilistic way using multiscale based hyperstacks. We showed the surplus value of multi-parent linking over single-parent linking and described a data structure capable of handling such complex child-parent connections.

As regards the complexity of probabilistic hyperstacks, we indicated how the number of linkages involved easily grows prohibitively. Two constraints have been introduced to keep this number limited. Their adequacy has been demonstrated experimentally.

The display of probabilistic images calls for proper visualization software: it seems most appropriate to let the calculation of the opacities of the voxels depend on the knowledge of the partial volume voxels. Until this becomes available we can only visualize two-dimensional segmentations, or simulate intelligent renderers by ad hoc postprocessing. Images produced this way do allow us to sufficiently evaluate our segmentation results.

We are currently investigating what other strategies can be pursued to calculate the probability for each link. Interesting features are, inter alia, the number of parent/child linkages per voxel, the scale, and the used blurring strategy. It seems promising to extend the affection formula with these features.

We also expect probabilistic segmentations to have advantages over conventional segmentation schemes with respect to quantitative measurements. The accuracy of calculated distances and volumes heavily depends on the accuracy of the segmentation. Probabilistic segmentation introduces voxels that are only partly contained in segments, which increases the accuracy of quantitative measures significantly [123, 124].

Experiments to quantify and verify these expectations are currently in progress.

4.A Single-parent data structure

In this appendix we will outline the design of a suitable data structure.

The data structure to store single-parent linked hyperstacks [116, 118, 25] is not suited to be extended to multi-parent linking (see Fig. 4.10). The single-parent data structure is a variant of doubly linked lists, with the modification that one parent having multiple children does not have separate child references, but each child has a sibling reference instead. A sibling linkage points to the next child belonging to that parent, thus creating a chain of children. Together with every parent reference per child, a triangular data structure is formed.

The main problem in applying this data structure to multi-parent linking is that it is impossible to discriminate between multiple children of one parent and multiple parents of one child. Introducing a fourth (sibling-like) type of reference to connect all parents of a child would create ambiguity, because parents can be shared by different children. Besides, the implementation would become unacceptably expensive.
4.B Multi-parent data structure by means of link containers

In this appendix we discuss the implementation of the multi-parent data structure by means of link containers.

The solution to the ambiguity problem of appendix 4.A is effectively a singly linked list, implemented with link containers (see Fig. 4.11). With the help of the link containers, only parent linkages are stored, keeping all parents that belong to one child together. Each voxel has a unique reference to one link in the link container—indexing the first parent—while all remaining parents are accessed by subsequent fields in the container. This makes multi-parent linking possible without introducing ambiguities.

The list of probabilities per ground voxel—denoting the chances that a voxel belongs to different segments—are implemented similarly by means of a root container (not shown in the figure).

Using link containers, the downward projecting of root values can easily be implemented as a bottom-up process. Since it can be shown that this segmentation step is equally expensive as the top-down scanning process used in the single-parent data structure, the halving of the number of linkages—owing to the use of singly- instead of doubly linked lists—will result in an appreciable reduction of computing time for the segmentation step.
Fig. 4.11. Schematic of the notion of link containers (right picture). The arrows in the hyperstack (left picture) are implemented as indices, so no pointers are involved.