Medical Image Analysis 5 (2001) 127–142

MEDICAL IMAGE ANALYSIS

www.elsevier.com/locate/media

Interaction in the segmentation of medical images: A survey

S.D. Olabarriaga[a,*], A.W.M. Smeulders[b]

[a]Informatics Institute, Federal University of Rio Grande do Sul, Caixa Postal 15064, 91501-970 Porto Alegre, RS, Brazil

[b]Informatics Institute, University of Amsterdam, Kruislaan 403, 1098 SJ Amsterdam, The Netherlands

Received 30 August 1999; received in revised form 1 June 2000; accepted 15 July 2000

Abstract

Segmentation of the object of interest is a difficult step in the analysis of digital images. Fully automatic methods sometimes fail, producing incorrect results and requiring the intervention of a human operator. This is often true in medical applications, where image segmentation is particularly difficult due to restrictions imposed by image acquisition, pathology and biological variation. In this paper we present an early review of the largely unknown territory of human–computer interaction in image segmentation. The purpose is to identify patterns in the use of interaction and to develop qualitative criteria to evaluate interactive segmentation methods. We discuss existing interactive methods with respect to the following aspects: the type of information provided by the user, how this information affects the computational part, and the purpose of interaction in the segmentation process. The discussion is based on the potential impact of each strategy on the accuracy, repeatability and interaction efficiency. Among others, these are important aspects to characterise and understand the implications of interaction to the results generated by an interactive segmentation method. This survey is focused on medical imaging, however similar patterns are expected to hold for other applications as well.

Keywords: Image segmentation; Medical images; Interactive segmentation; Human–computer interaction

1. Introduction

Segmentation is an intermediate step in the analysis of images where the object of interest is isolated from the background. The ultimate goal of segmentation is to identify which part of the data array makes up an object in the real world. Segmentation supports tasks such as measurement, visualisation, registration, reconstruction and content-based search, each of them with specific needs. For example, the demand for accuracy is much higher for measurement than for visualisation, while efficiency is more important for search in a large database than for surgery planning and simulation. In all applications, automatic processing is desirable, but sometimes unattainable due to limitations imposed by image acquisition, abnormalities in the scene, or both.

As a consequence, the intervention of a human operator is often needed to initialise the method, to check the accuracy of the result produced automatically, or even to correct the segmentation result manually. Interaction is usually adopted in applications with a high demand for accurate results, and where the volume of images is reasonable, allowing for human manipulation. This is the case of applications such as data input for geographical information systems (Gülch et al., 1998), formulation of query for content-based image retrieval (Gevers et al., 1998), and especially several medical applications in clinical practice and research (e.g., Shyu et al., 1999; Treece et al., 1999; Bullitt et al., 1999). In every day practice much interaction is employed for image segmentation in a large number of applications.

Contradicting intuition, however, the description of segmentation methods in the scientific literature usually emphasises the computational part, marginally discussing the interactive part, if at all. Since the methods use different strategies to combine the expertise of humans and computers, the outcome of such methods depends on the interaction strategy as much as it depends on computation. Consequently, a proper assessment of interactive seg-

*Corresponding author.
E-mail address: silvia@inf.ufrgs.br (S.D. Olabarriaga).

1361-8415/01/$ – see front matter
PII: S1361-8415(00)00041-4

mentation methods requires the computational and interactive parts to be equally understood.

In this paper we make an early attempt to enter the largely unknown territory of interaction in image segmentation, developing ideas introduced by Olabarriaga and Smeulders (1997) and Smeulders et al. (1997). Our goal is to better understand the implications of interaction for the design of interactive segmentation methods and how it can affect the segmentation results generated by these methods. With the review presented here we aim at identifying patters in the use of interaction, as well as developing qualitative criteria for the design and evaluation of segmentation methods that involve user intervention.

We start in Section 2 with a general view of an interactive segmentation process, introducing the main components and aspects considered in the subsequent evaluation. Next, we classify and discuss reviewed methods with respect to the following aspects: the type of information provided by the user (Section 3), how this information is translated into parameters for the computational part (Section 4) and what is the role played by the user in the segmentation process (Section 5). A final discussion and some conclusions are presented in Section 6. The review and discussion presented here are focused on medical applications, but similar patterns and trends are expected to hold for other application domains as well.

2. Interaction in image segmentation

A sketch of a general interactive segmentation method is illustrated in Fig. 1, where the main components are the user, the computational part, the interactive part and the user interface.

The *computational part* corresponds to one or more pieces of program capable of generating a delineation of the object of interest given some parameters. The parameter values can be determined from prior knowledge about the segmentation problem, from information provided by the user, or both. The type of algorithm used to find the object delineation is not directly relevant at this point. Whereas we acknowledge the influence of the underlying algorithm in the segmentation result, we leave the algorithmic part out of the discussion, as we conceive it as one or more black boxes. The computational component has been emphasised in the reviewed literature, be it region- or boundary-based, hard- or soft-bounded, model-based, based on morphological operators, etc. (see Udupa and Herman, 2000).

The *interactive part* is responsible for mediating information between the user and the computational part. Specifically, it translates the outcome produced by the computational part into visual feedback to the user and the data input by the user into parameters for the program. The actual communication between the computer and the user is done via the output and input devices controlled by the *user interface*. The user analyses the visual information displayed on the screen and reacts accordingly, providing feedback for the computation.

Note that this survey does not focus at the user interface level, where the choices for the implementation of a given human–machine dialog are investigated. At such level, one would discuss whether the user should input a threshold value via a slider or by directly clicking inside the object, or which would be the most intuitive and effective way to display the object delineation for user evaluation. Obviously, the user interface design is very important for the success of any interactive method. We believe that this topic is better covered by others, such as Schneiderman (1997) and Keller and Keller (1993). Instead, we are interested in interaction at a higher level, discussing different alternatives for the flow of information and control in the dialog between the user and the segmentation method, as well as their implications to the segmentation result.

To better compare interaction strategies adopted by existing methods, we analyse three aspects of this dialog:
1. the type of data input by the user during the segmentation process (Section 3);
2. how the input data are interpreted to feedback the computational part (Section 4); and
3. what is the purpose of user intervention in the process (Section 5).

Although this list is not exhaustive, it covers a large spectrum of aspects that facilitate the study of capabilities of interactive segmentation methods.

2.1. Evaluation

The objective evaluation of segmentation methods –

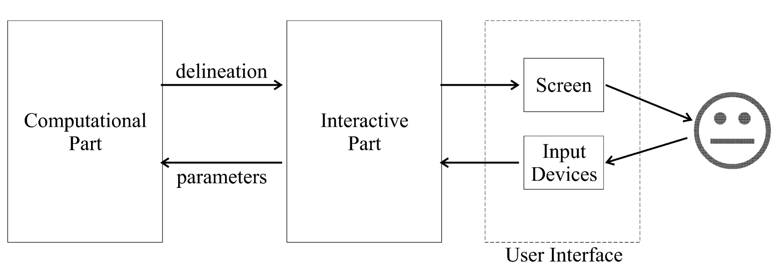

Fig. 1. Main components of an interactive segmentation method.

interactive or not – is a difficult issue. The evaluation depends on the task, with the consequence that the method's performance may be considered reasonable for one application and not acceptable for another. In the literature, however, a number of criteria have been adopted consistently to demonstrate the capabilities of interactive segmentation methods: accuracy, repeatability and efficiency (Udupa and Herman, 2000). Although these refer to the *results* generated by segmentation methods, they offer a reasonable starting point to assess the capabilities of interactive segmentation methods.

2.1.1. Accuracy

The most common evaluation criterion is accuracy, indicating the degree to which the delineation of the object corresponds to the truth. Accuracy can be assessed subjectively or objectively. Subjective evaluation of accuracy is done by human experts, who rank the result generated by the computational method when operated by others (e.g., Udupa et al., 1997; Bzostek et al., 1998) or by themselves (Maes, 1998). For objective evaluation, the result generated by the segmentation method is compared against the ground truth using different distance measures. Examples of measures used for this purpose are the area of difference and the distance between results (Yasnoff et al., 1977; Chalana and Kim, 1996), the cost to manually correct the result (de Graaf et al., 1992), and spatial correspondence (Bello and Colchester, 1998).

In many applications the ground truth for real images is not known. In such cases a rough estimation of accuracy can be obtained with synthetic images, for which the truth can be determined precisely (e.g., Cagnoni et al., 1998; Mortensen and Barrett, 1998). Synthetic images, however, are capable of reproducing limited operating conditions, even when a sophisticated perturbation model is used to generate the test set, as proposed by de Boer and Smeulders (1998). The evaluation of accuracy under realistic operating conditions therefore requires real images. The ground truth adopted then is a 'golden standard' usually generated by human experts using manual segmentation tools. Due to manual processing, the golden standard may incorporate variation and subjectivity, which must be taken into account in the global evaluation of accuracy as suggested by Collins et al. (1999).

Note that accuracy as a criterion makes more sense in the evaluation of segmentation results generated by automatic processing. In interactive methods, user participation is included in the process exactly to improve accuracy to the point where the result obtained by the method is 'always' satisfactory. The only situation where this is not true occurs when user control is limited by the interaction strategy, by the underlying computational method, or both. Consequently, a method is potentially accurate when it provides full control to the user to generate any desired result.

2.1.2. Repeatability

Repeatability evaluates to which extent the same result would be produced over different segmentation sessions when the user has the same intention. In this case, the same image and object are segmented several times by one human operator and the results are compared. The same procedure is followed to assess the inter-operator repeatability. The differences indicate the intra-operator or inter-operator variability of results – see examples of such studies in Udupa et al. (1997); Mortensen and Barrett (1998); Falcão et al. (1998); Gill et al. (1999) and Olabarriaga (1999).

Note that variability in the results can be due to two factors: difference in the operation of the segmentation tool (the user clicks the mouse at different image positions each time) or difference in judgement (the user considers the delineation to be located at different positions each time). A method potentially generates repeatable results when it takes precautions to minimise the effect of the first type of variation. Nothing can be done about the second type, though. Results obtained by interactive segmentation methods are inherently subjective, and this must be taken into account when evaluating their repeatability and accuracy.

2.1.3. Efficiency

Last but not least, it is important to evaluate efficiency, especially in the case of interactive methods. The total elapsed time could be used as an indicator of the method's overall efficiency (e.g., Mortensen and Barrett, 1998; Falcão et al., 1998), but it depends too much on the task. Moreover, this indicator hides the individual contributions of the computational and the interactive parts. For these reasons, we try to separate these parts and concentrate on the development of criteria to evaluate interaction efficiency per se.

Efficiency of the computational part is measured in terms of the time needed by the computer to generate the result. Computation should be fast enough to allow for interaction in real-time; objective evaluation in this case is straightforward and outside the scope of this paper.

With respect to the interactive part, efficiency is inversely proportional to the effort required from the user to accomplish the segmentation task. This effort is determined mostly by the amount and the nature of user interventions. The amount of interaction depends on the autonomy of the computational part, and it is often estimated in terms of the number of mouse clicks (e.g., Vehkomäki et al., 1997; Bzostek et al., 1998). As for the nature of interaction, it is necessary to evaluate the complexity of the task performed by the user. Task complexity involves several issues, among them the demand posed on mouse operation, the type of knowledge needed to input data during interaction, and the predictability of the method's behaviour in response to user input.

As a first example, consider the following drawing tasks: (1) draw a line around the object; (2) roughly sketch the object; and (3) draw the object's boundary. The effort to control the mouse increases with the need to draw more carefully, potentially leading to slower operation and higher levels of fatigue.

As a second example, consider two input tasks: (1) the user types the value of a parameter for the segmentation method; and (2) the user points the object in the image and, from this information, the program determines the appropriate parameter value automatically. In principle, the first task requires knowledge about the workings of the algorithm, forcing the user to think in two disjoint domains: the problem domain and the domain of the segmentation tool. As a consequence, slower operation is expected in the first case. Although the notion of intuition depends on the human operator's background and the task to be performed, one is inclined to say that the second task is more 'intuitive'.

As a final example, consider the situation where the user must interactively tune parameters for the segmentation method, progressively refining the result towards the desired delineation. For efficient interaction, the user should be able to predict the impact of his/her actions in the generated result. This can be achieved in different ways: by adopting controllable computational methods that can be activated in a straightforward manner (e.g., Griffin et al., 1994; Gleicher, 1995; Neuenschwander et al., 1997; Falcão et al., 1998; Mortensen and Barrett, 1998); and by providing real-time visual feedback, enabling the user to try out different alternatives in an efficient manner (e.g., Hastreiter and Ertl, 1998).

In conclusion, the evaluation of efficiency of interactive methods is mostly subjective, and we feel that measuring elapsed time will not be the definitive answer here. In general terms, it seems reasonable to say that an interactive method is potentially efficient when the computational part is fast, highly autonomous and predictable, and when user interventions are few, quick and simple. At any rate, the impact of complex user interventions is likely to be reduced or eliminated over time as the user learns to operate the segmentation tool.

3. Types of interaction input

Three main types of input provided by the user during the interactive segmentation process were identified in the reviewed methods: setting parameter values, pictorial input directly on the image grid, and choosing from pre-defined options in a menu.

3.1. Setting parameter values

This is the case of real parameter values $v \in [v_1, v_2]$, $[v_1, v_2] \subset \mathbb{R}$ with continuous impact on the computational method. Some examples of such parameters are: the threshold level for binarization (Lifshitz and Pizer, 1990; Cabral et al., 1993); the balance of weights in the cost function of a deformable model (Buck et al., 1995); the scale used to compute image derivatives and locate image structure (Lifshitz and Pizer, 1990); and the desired quality level of the segmentation result, given a quality criterion defined by an objective function (Elliot et al., 1992).

In a discrete range, values $v \in [v_1, v_2] \subset \mathbb{Z}$ refer to parameters that determine ordered 'levels' for the computational method, typically with a non-continuous impact on the segmentation result. Examples are the maximum number of iterations for progressive refinement via alternating region merging and splitting (Cabral et al., 1993) and the maximum size of the segmented region, given by the number of pixels (Sivewright and Elliot, 1994).

In most examples above, the value is input with a slider, a dial or a similar interactive technique, and the result obtained with the new parameter configuration is displayed on the screen for user evaluation.

3.2. Pictorial input on the image grid

Positions in the image grid $[\vec{x}, \vec{y}] \subset \mathbb{R}^n$ refer to spatial parameters for the computational method such as points, lines or regions.

In some methods, the spatial parameter roughly indicates the focus of attention, such as a rectangle corresponding to the region of interest in the image (Lifshitz and Pizer, 1990); image positions that belong (or do not belong) to the object (Higgins, 1994a; Griffin et al., 1994; Maes, 1998); a circular target area for the method, given by the centre point and radius (Bzostek et al., 1998); and lines indicating barriers that locally confine the segmented result to a region of any shape (Udupa, 1982; Sivewright and Elliot, 1994; Tieck et al., 1998).

In other situations, the spatial parameter corresponds to an initial delineation of the object of interest used to bootstrap the computational method. The initialisation may correspond to a rough outline of the object, which is the case of deformable models in general (McInerney and Terzopoulos, 1996). In these methods an initial curve or surface is optimised on the basis of a cost function that balances shape and image properties. Bootstrapping is done in basically two ways: (1) the user draws the initial curve freely (e.g., Kass et al., 1987); and (2) the user adjusts pre-defined templates to the object in the image (e.g., Sequeira and Pinson, 1990; Brinkley, 1993; Hinshaw et al., 1995; Buck et al., 1995; Neumann and Lorenz, 1999; Olabarriaga et al., 1999). Another example is presented by Vehkomäki et al. (1997), where a starting point and a few control points near the object's boundary are provided interactively. These points are used by the method to start the search for the optimal path in a graph representing edges in the image.

The method can be initialised also with samples of the object of interest. This is the case of interactive region-growing methods, where the user provides a seed corresponding to a pixel inside the object (e.g., Fontana et al., 1993; Cabral et al., 1993; Adams and Bischof, 1994; Sivewright and Elliot, 1994). In a similar fashion, a point inside the object is used by Gill et al. (1999) to initiate an inflating three-dimensional balloon, and by Höhne and Hanson (1992) the indicated position is used as a basis for connected component analysis. Another example is the method by Udupa et al. (1997), where the samples indicated by the user determine the image features of different types of objects to segment.

Finally, the spatial parameter may correspond to special points, such as image positions that attract or repel the contour (Kass et al., 1987).

Pictorial input is provided directly on the grey image, using the mouse or some other pointing device. The point, line, curve or region indicated by the user is shown in real-time using colour or another type of visual highlight. The level of detail used for visualisation and data input depend on the type of information provided by the user: a coarse resolution suffices to roughly specify the focus of attention, while a higher resolution is needed to precisely delineate the object.

3.3. Choosing from a menu

In this case, the user chooses an option from a pre-defined menu $o \in \{o_1, o_2, \ldots, o_n\}$, where entries o_i refer to parameter values indicating unordered categories for the computational method. Examples are commands to accept or reject the result generated by the program (Matsuyama, 1989; Udupa et al., 1997); to choose the type of global geometry model (or template) to use for the object of interest (Buck et al., 1995; Hinshaw et al., 1995); to choose among object properties such as 'it contains no holes' and 'it consists of one connected component' (Higgins, 1994a), and to indicate the situation that caused a failure in the computational part (Olabarriaga et al., 1999).

Different interaction techniques are adopted to implement the choice from pre-defined options, such as buttons, forms and iconic menus.

A variation of this input mode consists of offering the user a number of pre-computed segmentation results and the user selects the correct one. In this case, the user also selects among pre-defined options, but these indicate directly the result, and not parameter values. This type of interaction strategy is further explored in Section 5.4.

3.4. Discussion on user input

We distinguished three main types of user input: setting parameter values in a continuous or discrete interval; pictorial input, by directly pointing positions on the image grid; and menu-driven, by selecting options from a pre-defined repertoire (menu).

Setting parameters may be simple to implement, but it requires an insight of the user in the functioning of the computational part, potentially leading to inefficient interaction. This drawback may be lifted when the human operator receives specific training, or when the effect of interaction is immediately visualised on the screen. When real-time feedback of the resulting delineation is available, the user does not need to understand how the method works; he/she must only turn a knob until the desired result appears on the screen. An example is presented by Hastreiter and Ertl (1998): the user specifies the level of similarity for volume growing by instant visual inspection of the resulting segmentation in 2-D and 3-D visualization windows.

Pictorial input is simpler for the user, but it is typically time consuming. When interaction requires roughly indicating a region, a line or a point of interest on the image, a superficial glance at the screen and a few mouse clicks may suffice. If, in contrast, the interaction aims at precisely bootstrapping the computational method, more scrutiny during interaction may be required. The live lane method (Falcão et al., 1998) provides a good example where the balance between rough and precise operation is optimal: the user draws roughly and quickly where the edges of the object are well-defined, and therefore can be captured by the program automatically, but he/she slows down to draw more carefully otherwise.

Menu-driven input is most efficient, since it limits the choice of the user to selections, eliminating the need for hands-on manipulation of parameter values or the image on the screen. Moreover, the computation guides the user in the decision about the actions to take, possibly leading to more efficient interaction. On the other hand, this interaction strategy is likely to require sophisticated computational methods to support each of the options. The method described by Higgins (1994a), for example, uses heuristics to determine the method's configuration based on yes/no answers provided by the user about object properties. Input by menu is particularly suited for future optimisation by adding a computational criterion that is capable of making the choice based on knowledge acquired from past experience (e.g., during interaction with the user).

In all cases, for efficient interaction the program must provide visual feedback to the user in two moments: before the intervention, to help the user plan the next action; and after the intervention, to inform the user about the impact of his/her actions in the resulting delineation.

Additionally, precaution must be taken to prevent accidental variations in the user input from dominating the resulting delineation. This is more critical for the first two types of input data (parameter values and image positions). For the third type (menu), variation in the input data is greatly reduced by constraining the degrees of freedom, possibly leading to more repeatable results.

Finally, accuracy depends only indirectly on the type of input provided by the user during interaction, since the

Table 1
Summary of the types of interaction input, their consequences to accuracy, efficiency and repeatability, and possible ways to overcome drawbacks

Type	Accuracy	Repeatability	Efficiency
Parameter setting	−/+	−/+ Needs robustness against variations	− Needs immediate feedback
Pictorial input	+/++	− Needs robustness against variations	−/+ Needs adaptive scrutiny level
Menu	− Needs complete set of options	+/++	+ Needs sophisticated computation

program can use the input data in different ways (see Sections 4 and 5). In fact, accuracy depends mostly on the computational part and its capability to generate the delineation as commanded by the user. As a general rule, however, one could say that accuracy is favoured when data input occurs as close as possible to the object delineation, preferably in the image domain. Menu-driven input potentially represents the worst case, unless the set of pre-defined options is capable of covering all desired delineations.

Table 1 presents a summary of the discussion above for typical situations found in the majority of the reviewed methods.

4. Computational consequence of user input

The data provided by the user can be used to configure parameters for the computational part directly or indirectly. In the first case, input data have direct impact on the resulting delineation, usually leading to low-level interaction where the user must have at least some basic knowledge about the method to operate the system. In the other case, user input is interpreted by the method for basically two purposes: to implement high-level interaction, where the operation is simplified; and to reduce user interventions, where the method 'learns' from interaction.

4.1. High-level interaction for operation simplicity

In this case, the technical aspects of the segmentation method are hidden from the user, elevating interaction to a higher level of abstraction. The communication level is usually defined closer to the mental model of the user, facilitating operation and possibly leading to interaction efficiency.

In such methods, input data are converted from the high level of abstraction recognized by the user into the low-level parameters recognized by the computational part. We treat this conversion as a mapping function defined as follows:

$$\vec{p} = f(\vec{u}), \tag{1}$$

where $\vec{u}$ is a vector of data corresponding to a setting v, a pictorial input $[\vec{x},\vec{y}]$ or an option o provided by the user. The vector $\vec{p}$ contains parameters for the computational part, and f is the function that interprets user input, mapping $\vec{u}$ onto $\vec{p}$. Note that these functions can be implemented with different techniques, but it is outside the scope here to discuss them in depth.

We illustrate different types of interpreter functions f in the reviewed methods:

- $\vec{p} = f([\vec{x}, \vec{y}])$. The spatial coordinates $[\vec{x}, \vec{y}]$ indicated by the user serve as a rough approximation of the object in the image. These positions are used as a criterion to choose the most appropriate delineation among reasonable options considered by the computational part, but they are not included in the segmentation result directly. Segmentation methods that adopt this strategy are the magic crayon (Beard et al., 1994), the live wire and live lane (Falcão et al., 1998), intelligent scissors (Mortensen and Barrett, 1998), and the active paintbrush (Maes, 1998). Another example is found in Higgins (1994a), where the user provides 'iconic cues' corresponding to pictorial information about the properties of regions in the image such as 'it is totally inside the object of interest,' 'it contains the object', and 'it is completely outside the object'. This information is used to constrain the location of the segmentation result. Finally, the maximum size of the object of interest is estimated by Higgins (1994b) from a rough sample drawn by the user directly on the image.
- $\vec{p} = f(g(x, y))$, where $g(x, y)$ is the image intensity function. The pixels indicated by the user serve to determine the image properties of the object at hand. In the simplest case, $f(\cdot)$ is the identity function, and the parameter value corresponds to the intensity level at the pixel location. This is the case in the method by Worth et al. (1997), where the threshold level is indicated by pointing one of the iso-intensity contours overlaid on the image. More complex functions compute the statistics of pixel values in the region of interest based on a sample region indicated by the user. This is the case of the methods described by Cabral et al. (1993), Adams and Bischof (1994), Higgins (1994b), Griffin et al. (1994), Udupa et al. (1997), Maes (1998) and Carvalho et al. (1999). Finally, some methods determine the type of image intensity profile at the boundary positions pointed by the user, such as in Falcão et al. (1998) and Neugebauer (1995).
- $\vec{p} = f\left(\left[(\partial/\partial^n t)\vec{x}(t),\ (\partial/\partial^n t)\vec{y}(t)\right]\right)$, $n \geqslant 0$, where t stands for time. In this case, the mapping function also takes into account the sequence of user input. In static interpretation ($n = 0$), values are considered indepen-

dently from the sequence in which the user informs them. This is the case of the large majority of interactive methods, where the interaction data correspond directly to the current mouse position on the image grid. In a few methods, however, the interpreter function also takes into account the dynamics of interaction, based on higher order differentials of the user input taken over time. A first example is found in Udupa (1982), where the direction of the mouse stroke is used to indicate the inside and outside of objects. The boundary is therefore oriented, assuming that the interior is always at the left of the curve drawn by the user. In this case, the mapping function is defined as

$$\vec{p} = f((\partial/\partial t)x(t), (\partial/\partial t)y(t)).$$

A second example is found in the live lane method (Falcão et al., 1998), where the mouse speed is used as an indication of local image quality. The method assumes that users draw quickly when the boundary is clearly visible, but they move the cursor slowly for careful delineation when the visual evidence of the boundary is weak. This information is used to dynamically calibrate weights in the cost function defining the boundary features, adapting it locally to the diverse imaging conditions. The interpreter function in this case is defined as

$$\vec{p} = f(\|(\partial/\partial t)x(t), (\partial/\partial t)y(t)\|).$$

Finally, mouse acceleration is used by Falcão et al. (1998) as an indication of the boundary between regions with low and high image quality, allowing for a smooth transition of parameters between the two. The interpreter function in this case is defined as $\vec{p} = f(\|(\partial/\partial^2 t)x(t), (\partial/\partial^2 t)y(t)\|)$.

In all cases, real-time visual feedback about the effect of interaction in the segmentation result is displayed to the user.

4.2. Learning for reduction of user interventions

In this case, user input is used to update the knowledge contained in the method. As much as possible, the information obtained with interaction is reused for a similar segmentation task in the same or in another image; we refer to this as 'learning'. The goal is to improve the performance of the computational part and possibly reduce the need for future user intervention, leading to interaction efficiency. This interaction strategy is found in existing methods in the forms described below.

In the live wire (Falcão et al., 1998) and intelligent scissors (Mortensen and Barrett, 1998) methods, the image properties of stable boundary parts are used to dynamically adjust parameters for the computational method. In this case, the parameters refer to the cost function defining the expected boundary features, such as the type of image intensity profile and statistics of pixel intensity values.

In the method described by Elliot et al. (1992), the segmentation result obtained with user interaction is compared to the result obtained when the default parameter settings are used. The difference between the two is used to calibrate the parameters for the computational part, which are used as default values in future segmentation sessions. In this method, the parameter values express statistical knowledge about the expected image intensity pattern inside the object of interest.

In slice-by-slice segmentation of 3-D images, the information obtained with interaction in one slice can be propagated to the next in different ways. In the method by Sijbers et al. (1996), all the pixels inside the resulting object are propagated as seeds for region growing in the next slice. In the active paintbrush (Maes, 1998), selected points inside and outside the resulting object are propagated as 'hint' that indicate regions in the next slice where the object should (or should not) be located. The interactive method described by Cagnoni et al. (1998) uses a set of reference contours drawn by the user to find the optimal parameters for an elastic-contour model. The optimised parameters are used in all the other slices in the same or in another dataset. In Xiaohan and Yla-Jaaski (1995), the resulting boundary itself is propagated as the initial contour for deformation in the next slice. And finally, in the method by Wink et al. (1997) the contour in the next slice is estimated on the basis of local similarity measures of the image intensity pattern at the resulting boundary.

These methods use different techniques to adapt the program to the information provided by the user, such as dynamic programming (Falcão et al., 1998), statistical analysis (Elliot et al., 1992) and genetic algorithms (Cagnoni et al., 1998).

4.3. Discussion about interpretation of user input

With respect to the computational consequence of interaction, we discriminate between direct and indirect usage of user input to derive information to configure the computational method. Interpretation of user input is added to a method with two purposes: to achieve interaction simplicity or to reduce the amount of user interventions. In both cases, the final goal is to achieve interaction efficiency by adding 'intelligence' to the method, in the sense of 'the ability to learn, understand or cope with a new situation' (Neufeldt and Sparks, 1995).

For the first purpose of intelligence (interaction simplicity), the user only sees the grey image on the screen and draws the object with an intuitive graphical tool. In this case, he/she is possibly unaware of the technical aspects of the mechanism that makes the tool have an active behaviour. The choices here are to interpret spatial coordinates, image intensity values and the dynamic aspects of

Table 2
Summary of the purposes of intelligence, their consequences to the accuracy, efficiency and repeatability, and possible ways to overcome drawbacks

Purpose of intelligence	Accuracy	Repeatability	Efficiency
Operation simplicity	−/+ Needs predictable behaviour	−/+ Needs robustness against variations	+
Reduce interventions	−/+ Needs predictable behaviour	+	−/+ Needs sophisticated learning mechanism

interaction. In all cases, the computational part should be robust with respect to accidental variations in user input to guarantee repeatable results. This demand increases proportionally with the need for precise or detailed interaction, respectively, when spatial coordinates, intensity values and dynamic aspects are used to derive parameters for the computational method. In particular, taking the dynamics of interaction into account poses the highest demands on the computational part; on the other hand, this strategy provides the largest impact of interaction on the computation.

For the second purpose of intelligence (reduce the amount of user interventions), an analysis of the pattern of the interaction, together with the image data and the final result, is used to revise the prior knowledge built into the segmentation method. This strategy may lead to long-term interaction efficiency and repeatable results due to the reduction of situations where the user must take action. The proviso here is that the computational method is capable of identifying the circumstances under which the revision should be applied. In the reviewed methods, for example, this strategy is found mostly in slice-by-slice segmentation methods, where the propagation of knowledge is facilitated by spatial coherence. Even in this case, however, the result may be incorrect, requiring human supervision (e.g., Cagnoni et al., 1998).

A final observation applies to both types of intelligence: the interpretation of interaction should be predictable, such that the user can anticipate the impact of his/her interventions in the segmentation result. Predictability is important not only to guarantee interaction efficiency, but also to enable the user to control the process, leading to accurate results.

Table 2 presents a summary of the discussion above for typical situations found in the majority of the reviewed methods.

5. Role of the user in the segmentation process

As a general rule, the goal of interactive segmentation methods is to combine a human operator and a computer to obtain an accurate delineation of the object of interest in an efficient manner. The underlying assumption is that the user has a clear vision of the 'correct' delineation of the object in the image, so his/her role in the segmentation process is mainly to guide the computational part to find it efficiently.

Consider the general segmentation process illustrated in Fig. 2, where the computational part generates segmentation results based on the image data and parameters configured during an initialisation procedure. In the interactive scenarios identified in the reviewed segmentation methods, the role of the user is to *judge* whether the result obtained by the computational part is correct, to *correct* results generated by the computational part, to *set parameters* for the computational part, to *compose* the object delineation from primitive results generated by the computational part, and to *build* a dedicated computational method with low-level operations. In many methods, the user plays more than one role.

5.1. Judging the result

In all the reviewed segmentation methods, the user has the ultimate word about the correctness of a result generated by the computational part. If the delineation is not satisfactory, the user rejects it and a correction strategy is activated (see illustration in Fig. 3).

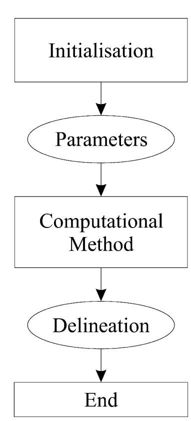

Fig. 2. Components of a generic segmentation process without interaction.

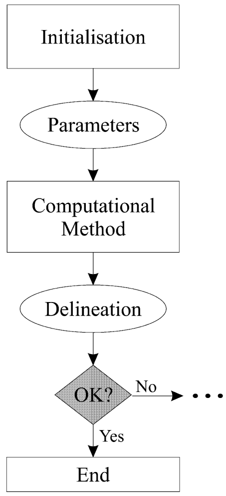

Fig. 3. Components of an interactive segmentation process where the user judges the delineation generated by the computational part. User actions affect the components in grey (see also Fig. 2).

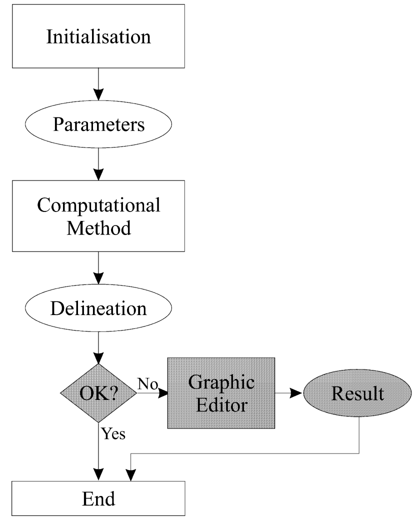

Fig. 4. Components of an interactive segmentation process where the user manually corrects the delineation generated by computation. User actions affect the components in grey (see also Fig. 2).

For example, an interactive procedure could be activated, by means of which the user edits the delineation directly (see Section 5.2) or provides extra information to reconfigure the computational part (see Section 5.3). Another approach is to use the 'accept/reject' answer to provide feedback for an expert system capable of autonomously reconfiguring parameters for the computational part (see Matsuyama, 1989).

All methods display the resulting delineation on the screen, enabling the user to instantly evaluate its location on the image grid. Different visualization schemes are adopted, such as to show the contour of the object or its interior in colour for better discrimination from the grey image in the background.

5.2. Correcting the result

If the result generated by the computational part is wrong, the user can correct it directly using a graphic editor (see Fig. 4). This practice is mentioned in the reviewed literature as the remedy for situations where the computational method fails (e.g., Krivanek and Sonka, 1998).

Examples of methods that use this strategy are described by Höhne and Hanson (1992), Elliot et al. (1992), Gauch (1999) and Cagnoni et al. (1998). For region-based methods, editing tools such as 'add pixel to object' and 'remove pixel from object' are used (Gauch, 1999). Several pixels can be corrected at once with tools based on morphological operations like erosion and dilation (Höhne and Hanson, 1992). For boundary-based methods, direct curve manipulation tools are used (e.g., Elliot et al., 1992; Cagnoni et al., 1998).

A higher level of editing tool is found in Bullitt et al. (1999), where the user can modify a tree of vessels generated automatically, deleting or linking vessel segments and associated sub-trees in one operation.

5.3. Setting parameters

In the large majority of the reviewed methods, interaction is used to set parameters for the computational method. We distinguish two situations for user intervention: to start the method (initialisation) or to provide information that is used to dynamically reconfigure parameter values (steering).

5.3.1. Initialisation

The user sets the initial parameter values and the computational part is executed, generating results that are displayed on the screen for user evaluation. If the resulting delineation is not satisfactory, the user adjusts parameter values and the computation is repeated (see Fig. 5(a)). The parameters usually affect computation globally, such that the new parameters can modify the entire delineation, and not only the area that was wrong.

Examples of methods in this category are thresholding, where the user provides a threshold level (e.g., Höhne and Hanson, 1992; Worth et al., 1997); region-growing, where the user indicates a seed position (e.g., Cabral et al., 1993; Adams and Bischof, 1994; Sivewright and Elliot, 1994); and deformable models (McInerney and Terzopoulos, 1996), where the user draws an initial contour to be deformed by the model and configures model parameters such as the balance between internal and external model constraints.

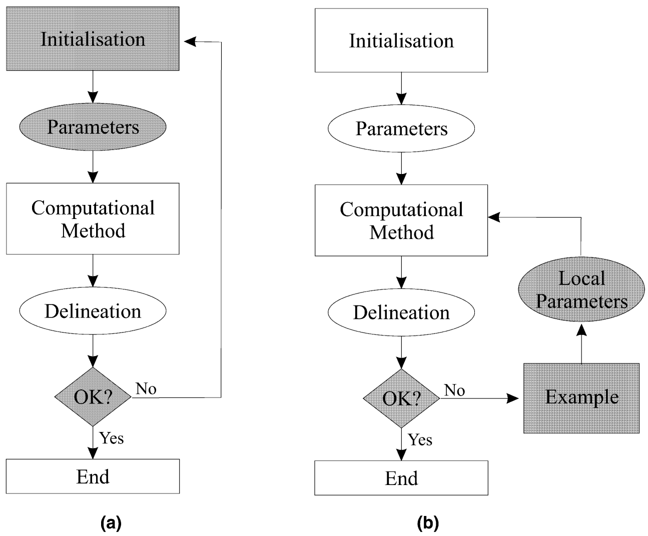

Fig. 5. Components of an interactive segmentation process where the user sets parameters for the computational part. User actions affect the components in grey (see also Fig. 2). (a) Initialisation. (b) Steering.

All methods provide visual feedback of the delineation obtained with the new parameter values, which are input using the strategies discussed in Section 3.

5.3.2. Steering

The user dynamically provides local information indicating the desired outcome, guiding the computational part in a process of progressive refinement of the segmentation result. The information indicated by the user is used to locally adjust the value of parameters for the computational method, while keeping unchanged the regions where the delineation is correct (Fig. 5(b)).

In the 'live wire' (Falcão et al., 1998) and 'intelligent scissors' (Barret and Mortensen, 1996) methods, the user indicates one point in the boundary and drags the mouse; the best path between the initial point and the current mouse position is determined by optimising a cost function. The resulting line is displayed in real-time, so that the user can dynamically evaluate the result and move the mouse to obtain a better delineation. The contour is accepted when the user clicks the mouse button, and this position becomes the starting point for a new boundary segment. The cost function is based on a weighted combination of image-derived features, with parameter values initialised from prior knowledge and dynamically updated based on the contour parts accepted by the user.

In line tracking methods, the user indicates a few points and the computational method finds the line connecting them based on a model describing how the line is expected to appear in the image. If the resulting line is not correct, the user can indicate additional points, which are used to modify or complement the model constraints. Examples of methods that operate in this fashion are found in Neugebauer (1995), where the path of minimal cost is searched simultaneously from given starting and ending positions, and in Houtsmuller et al. (1993), where the medial axis of a chromosome is tracked starting from a given direction and position. This method follows the line corresponding to the maximum of image intensity until the given ending position is reached. Another example is the ziplock snake (Neuenschwander et al., 1997), a method based on deformable models (McInerney and Terzopoulos, 1996). In this case, two open curves grow toward each other starting from given positions, based on image and shape constraints.

Finally, in Griffin et al. (1994) the user can interactively adjust the hierarchy of segmented regions by splitting regions that belong to different objects. The user indicates points inside and outside the object of interest, and this information is used to locally constrain region merging, generating a modified hierarchy.

In all methods that adopt steering, pictorial input is used and the resulting delineation is display to the user in real-time.

5.4. Composing the result

The underlying motivation in this case is to reduce the amount of user intervention by ''having the computer derive the syntactically defined regions and let these serve as a means of communication between the human and the computer, such that the user interactively and quickly can specify the semantically correct regions from the syntactically defined ones'' (Pizer et al., 1990).

The image is initially segmented with a computational method, generating *primitive* results that correspond to a large number of regions or boundaries in the image (over-segmentation). The assumption here is that the object of interest is composed of a subset of the primitives generated automatically. Interaction therefore consists of selecting the primitives that compose the delineation of the object of interest (see Fig. 6).

In the simplest case, the user only points the correct delineation among several generated automatically, such as in the methods by Krivanek and Sonka (1998) and Udupa et al. (1997). In this situation one mouse click suffices. In most methods, however, the user must select and combine primitives into a single result, progressively refining the object delineation. These methods differ with respect to two aspects: the composing tools used to manipulate and combine primitives into a single object and the mechanism to select primitives.

In terms of composing facilities, the reviewed methods are alike in the sense that all pixels in the selected primitive can be added or removed from the intermediate delineation at once. In most methods, the user cannot modify the primitives, thus accuracy ultimately depends on the success of pre-segmentation. In other words, the process fails if the desired delineation cannot be expressed as a combination of the available primitives. In response to

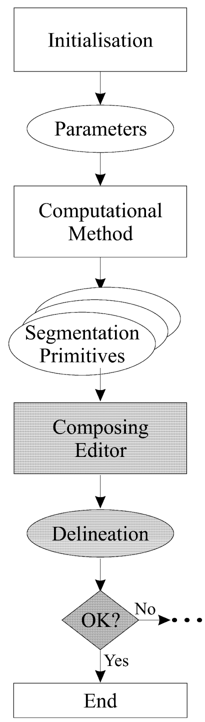

Fig. 6. Components of an interactive segmentation process where the user composes the result with primitives generated by the computational part. User actions affect the components in grey (see also Fig. 2).

this problem, some methods allow the user to edit the primitive segmentation directly, using interactive tools such as 'merge' two regions and 'split' a region with a given edge (e.g., Elliot et al., 1992; Tieck et al., 1998). A more elegant solution is described by Griffin et al. (1994), where the user can modify the hierarchy on the basis of interactive steering (see also Section 5.3).

In terms of selection of primitives, the methods can be roughly of two types, depending on how the primitives are organised: horizontally or hierarchically.

5.4.1. Horizontal organisation of segmentation primitives

In this case, there is no ordering in the results generated by the computational part, thus all primitives are eligible for the final delineation with equal priority. This is the case of the methods described by Tieck et al. (1998) and Sijbers et al. (1996), where the user selects regions from an over-segmented image to compose the delineation. In a similar fashion, Vehkomäki et al. (1997) provides an example where edge grouping is controlled interactively. The user selects a few control points near the object, and these are used to start the search for the optimal path linking pre-computed edge fragments.

5.4.2. Hierarchy of segmentation primitives

In this case, the computational part groups primitives progressively into a hierarchy ranging from coarse to fine segmentation. The goal is to minimise user intervention by offering pre-combined options that are likely to correspond to the desired delineation. When grouping is successful, large portions of the object can be selected and combined with a single mouse click.

A hierarchy of primitives is usually represented by a tree, where the leaves correspond to the finer segmentation level, e.g., the image at full pixel resolution, and the root corresponds to the coarser level, e.g., the complete image. Each tree level corresponds to segmentation results obtained by monotonically varying the values of one parameter. In some methods, the parameter refers to the resolution or scale (Lifshitz and Pizer, 1990; Eberly and Pizer, 1994; Beard et al., 1994; Gauch, 1999). In other methods, the parameter corresponds to the confidence level in the segmentation result measured by an objective function (Elliot et al., 1992; Fontana et al., 1993; Maes, 1998). Different strategies can be used to link the levels in the tree. The following examples illustrate how interaction is carried out in the reviewed methods to select segmentation primitives from such a hierarchy.

In Lifshitz and Pizer (1990), the hierarchy of primitives obtained at different resolutions is displayed as a tree in 3-D vector representation. The user can rotate the tree for better visualisation. In the first type of selection tool, the user picks any branch of the tree with a light pen, and all pixels contained in the corresponding sub-tree are selected. In the second type of tool, the user manipulates two sliders that specify lower and upper limits for the scale parameter, corresponding to upper and lower levels in the tree. All objects detected within this range are selected.

In some methods, the hierarchy itself is shown to the user only indirectly. This is the case of Elliot et al. (1992), where the operator inspects the primitive regions obtained at different confidence levels by manipulating a slider. The regions delineated at each level are displayed on the screen in real-time, and the user can select any of them with the mouse at any moment. A similar approach is adopted by Fontana et al. (1993), where all primitive contours are simultaneously drawn on the grey image, using colours to indicate different confidence levels. The user selects a contour by directly picking it with the mouse.

In other methods the hierarchy is hidden completely from the user. This is the case of the methods by Beard et al. (1994); Sijbers et al. (1996) and Maes (1998), where the user simply drags the mouse over the image where the object of interest is located. All regions in the low-level sub-tree containing the pointed pixels are automatically selected. Another selection tool implemented by Maes (1998) provides a higher level of interaction: the operator only clicks at points inside and outside the region of interest. All pixels contained in the largest sub-tree satisfying the given constraints are selected, i.e., the sub-tree that contains the 'interior' pixels and does not contain any of the 'exterior' pixels.

Pictorial input is used in most methods, based on two types of visual feedback that are displayed separately: the original image with the segmentation primitives, and the intermediate result under construction.

5.5. Building a dedicated segmentation process

Here the purpose of interaction is to define a sequence of low-level image processing operations necessary to obtain the desired delineation. In this case, the user interactively builds a dedicated computational method for the problem at hand, choosing the appropriate operators and the corresponding parameters (see Fig. 7).

Data-flow systems such as KHOROS (Kostantinides and Rasure, 1994) and AVS (Upson et al., 1989) are examples where this strategy is adopted. In such systems, the construction of a data-flow is straightforward when a visual language is used. On the other hand, the choice of low-level image processing operations and the corresponding parameters remains as a difficult problem that requires much knowledge about the available image processing operators.

The system by Higgins et al. (1994a) seems easier to operate. In this case, the user provides ‘hints’ or examples of the desired delineation directly on the grey image. Based on this information, the system automatically chooses the adequate sequence of operators and parameters necessary to accomplish the goal defined by the user. The construction of the segmentation process can be based on simple heuristics encoded in a look-up table (e.g., Higgins et al., 1994a) or on explicit knowledge – e.g., expert systems reviewed by Matsuyama (1989) and Crubezy et al. (1997).

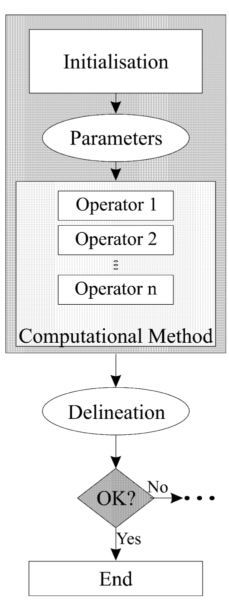

Fig. 7. Components of an interactive segmentation process where the user builds a dedicated segmentation process. The components in grey are generated or configured on the basis of interaction (see also Fig. 2).

5.6. Discussion about the role of the user

We identified five roles for the user in the segmentation process: to accept or reject the delineation generated by computation, to correct the outcome of the computational part, to initialise or to steer parameters for the computational part, to select among results generated automatically, and to define targets for the generation of a computational method.

The role of judging the delineation generated by the computational part only makes sense as a complementary interaction strategy. The purpose in this case is to guarantee accurate results by keeping the user in control of the final delineation generated by the process, as recommended by Stiehl (1990), COVIRA (1995) and Gerritsen et al. (1995).

For the role of editing the result, the purpose of interaction is to guarantee accuracy even when the computational part is not capable of generating the correct delineation. This strategy is adopted in large scale, but it has drawbacks. In the first place, it may lead to inefficient and non-repeatable interaction when corrections are indicated freely at the pixel level. In the second place, the resulting delineation may have non-uniform properties, since two processes of completely different nature generate it, one automatic and the other manual. Consequently, it makes little sense to design a very precise computational method that is followed by complete freedom in making corrections.

In the case of parameter initialisation, the purpose of interaction is to bootstrap the computational part in an efficient way. This strategy requires immediate visual feedback of results obtained with the given parameter values to allow for quick user reaction when the result is not satisfactory. Note, however, that the new parameter values can modify the entire delineation, and not only the area that was wrong, possibly leading to inefficient interaction and inaccurate results.

For the role of steering, the purpose is to keep the user in control of the entire segmentation process, such that manual editing at the end is not necessary. This control has an indirect effect on the delineation, with the consequence that results are repeatable, except in the extreme case where user input is detailed to the pixel level. Moreover, computation and interaction are integrated into one process, leading to segmentation results with uniform properties. And finally, it is possible to update the knowledge about the segmentation problem based on the modifications introduced by interaction, reducing the need for future user participation by computational learning.

For the role of selecting among results, the purpose is to let the computer find a number of possible delineations and let the user decide on the most reasonable ones. Interaction in this case is expected to be efficient, since the user can compose a delineation by clicking just a few pre-computed objects on the screen. As a consequence, there is a good chance of repeatable results because the final delineation is mostly generated by computation. Moreover, the segmentation result has uniform properties, since it is generated by one process. On the other hand, accuracy depends on the success of the primitive segmentation.

The selection and combination of primitives can be inefficient due to the large number of components to manipulate. This drawback is lifted in methods where the primitives are grouped into a hierarchy. In such cases, higher demands are posed on the visual tools, since an effective visualisation scheme is needed to reveal the available primitives at different levels of the hierarchy. And finally, this strategy usually requires from the user some knowledge about the workings of the algorithm, with the exception of methods where the hierarchy is completely hidden from the user (e.g., Maes, 1998).

For the role of building the computational method, the purpose is to let the user establish targets that guide the construction of an appropriate combination of operations to accomplish a given task. The major disadvantage here is that there is no guarantee that the computational method will actually be capable of finding an accurate delineation. Further interaction for correction may be needed anyway. As an additional drawback, the sequence of operations is composed on the basis of a given image, thus the operators and parameters do not necessarily hold for other images in the same application.

Table 3 presents a summary of the discussion above for typical situations found in the reviewed methods.

Table 3
Summary of the user roles in the segmentation process, their consequences to the accuracy, repeatability and efficiency, and possible ways to overcome drawbacks (see also Fig. 2)

Type	Accuracy	Repeatability	Efficiency
Judge	+ / + +	+	− / + Needs correction strategy
Correct	+ / + +	−	− / + Needs high level tools
Initialise	− / + Needs controllable computation	+	− / + Needs immediate feedback
Steer	+ / + +	+	+
Compose (horizontal)	− / + Needs editing of primitives	+	− / + Needs efficient selection tools
Compose (hierarchical)	− / + Needs editing of primitives or hierarchy	+	+ / + +
Build	− / +	+	−

6. Discussion and conclusions

In this survey we analysed the capabilities of interactive methods based on the following aspects: the type of user input (Section 3), the consequence of interaction in terms of parameters for the computational method (Section 4), and the purpose of interaction and the role played by the user (Section 5).

Concerning the type of user input, we have made a distinction between parameter setting, pictorial specification and choosing options from a menu. With respect to the computational consequence of interaction, we discriminate between direct and indirect usage of user input to derive information to configure the computational method. Concerning the purpose of interaction, we discriminate six roles for the user in the segmentation process: judge, correct, initialise, steer, compose and build. For each of these interaction strategies, we discussed the consequences for the accuracy and repeatability of results and for the efficiency of interaction. This discussion leads to the following conclusions.

A necessary condition for *accuracy* is that the outcome of interaction plus computation is complete, admitting all desired delineations. Completeness of interaction is achieved when user intervention can affect the parameters for the computational method in a non-limiting fashion. Completeness of computation is more difficult to achieve because it requires a flexible computational model that allows for local adjustment of parameter values (e.g., Olabarriaga et al., submitted). Moreover, the target of segmentation might be unreachable within the scope of the interactive method due to problems in the image such as low contrast, noise, overlapping or touching objects, non-uniform image acquisition, partial volume voxels and abnormal shape. In such cases, interactive steering and manual correction may be the solution.

Repeatable results are obtained when the final delineation is generated mostly by the computational part. This situation can happen only when the user input is not taken directly as part of the result, but it is used instead to configure parameters for the computational method or to select the best solution among computed results.

Efficient interaction is achieved when the operation of the system is simple and kept at a minimum. Simple operation is obtained when pictorial input is used, or when interaction is carried out at a higher abstract level than the underlying computational method. Interaction is kept at a minimum when the user only has to choose among options or when the system can learn from interaction in the long term.

Based on these conclusions, the following strategies seem promising for the design of efficient interactive segmentation methods that generate accurate and repeatable results:

- design an integrated process for interaction and computation;

- use pictorial input to the computational process;
- minimise the amount of interaction by presenting options for user selection;
- involve the user in the initialisation of the segmentation process to provide information that can bootstrap or lead the method to the desired segmentation result more quickly;
- properly visualise the working of the computational part to enable an effective user's response;
- keep the user in the control during the whole process to generate accurate results;
- emphasise computation after each interaction to generate repeatable results;
- add intelligent behaviour to elevate the abstraction level of interaction; and
- add intelligence to learn from interaction and reduce the need future interventions;

In conclusion, interactive segmentation should not be confused with *manual* processing, which has a bad reputation due to subjective results and inefficient operation. On the contrary, this survey reveals that several interactive segmentation methods aim at accurate and repeatable results and efficient operation, a goal that is achieved in many of the examples presented here. As a consequence, interactive methods constitute a reasonable alternative for complex segmentation problems where fully automatic computation is not possible with the digital imaging tools at hand. This is the case of many medical applications, which could benefit from an interactive strategy instead of adopting a purely manual process, such as found in many practical problems (e.g., Treece et al., 1999).

Acknowledgements

Supported by the Brazilian Council of Research and Development (CNPq). We thank the reviewers for their valuable comments and suggestions.

References

Adams, R., Bischof, L., 1994. Seeded region growing. IEEE Trans. Pattern Anal. Machine Intelligence 16 (6), 641–647.

Barret, W.A., Mortensen, E.N., 1996. Fast, accurate, and reproducible live-wire boundary extraction. In: Höhne, K.H., Kikinis, R. (Eds.), Proc. Conf. on Visualization in Biomedical Computing. Lecture Notes in Computer Science, Vol. 1131. Springer, Berlin, pp. 183–192.

Beard, D.V., Eberly, D., Hemminger, B., Pizer, S.M., 1994. Interacting with image hierarchies for fast and accurate object segmentation. In: Proceedings of the SPIE, Vol. 2167 of Proc. SPIE Conf. on Medical Imaging. SPIE, Bellingham, WA, pp. 10–17.

Bello, F., Colchester, A.C.F., 1998. Measuring global and local spatial correspondence using information theory. In: Wells, W.M., Colchester, A., Delp, S. (Eds.). Proc. Conf. on Medical Image Computing and Computer-Assisted Intervention (MICCAI'98), Vol. 1496 of Lecture Notes in Computer Science. Springer, Berlin, pp. 964–973.

Brinkley, J.F., 1993. A flexible, generic model for anatomic shape: application to interactive two-dimensional medical image segmentation and matching. Comput. Biomed. Res. 26, 121–142.

Buck, T.A., Ehricke, H.H., Strasser, W., Thurfjel, L., 1995. 3D segmentation of medical structures by integration of ray-casting with anatomic knowledge. Computers and Graphics 19 (3), 441–449.

Bullitt, E., Ayward, S., Liu, A., Stone, J., Mukherji, S.K., Coffey, C., Gerig, G., Pizer, S.M., 1999. 3D graph description of the intercerebral vasculare from segmented MRA and tests of accuracy by comparison with X-ray angiograms. In: Kuba, A., Sámal, M., Todd-Pokropek, A. (Eds.). Information Processing in Medical Imaging: Proc. Int. Conf. (IPMI'99), Vol. 1613 of Lecture Notes in Computer Science. Springer, Berlin, pp. 308–321.

Bzostek, A., Ionescu, G., Carrat, L., Barbe, C., Chavanon, O., Troccaz, J., 1998. Isolating moving anatomy in ultrasound without anatomical knowledge: Application to computer-assisted pericardial punctures. In: Wells, W.M., Colchester, A., Delp, S. (Eds.). Proc. Conf. on Medical Image Computing and Computer-Assisted Intervention (MICCAI'98), Vol. 1496 of Lecture Notes in Computer Science. Springer, Berlin, pp. 1041–1048.

Cabral, J.E., White, K.S., Kim, Y., Effmann, E.L., 1993. Interactive segmentation of brain tumors in MR images using 3D region-growing. In: Proc. SPIE Conf. on Medical Imaging, Vol. 1898 of Proceedings of the SPIE. SPIE, Bellingham, WA, pp. 171–181.

Cagnoni, S., Dobrzeniecki, A.B., Poli, R., Yanch, J.C., 1998. Genetic algorithm-based interactive segmentation of 3D medical images. Image and Vision Computing 17, 881–895.

Carvalho, B.M., Gau, C.J., Herman, G.T., Kong, T.Y., 1999. Algorithms for fuzzy segmentation. Pattern Analysis and Applications 2, 73–81.

Chalana, V., Kim, Y., 1996. A methodology for evaluation of image segmentation algorithms on medical images. In: Proc. SPIE Conf. on Medical Imaging, Vol. 2710 of Proceedings of the SPIE. SPIE, Bellingham, WA, pp. 178–189.

Collins, D.L., Zijdenbos, A.P., Baaré, W.F.C., Evans, A., 1999. ANIMAL+INSECT: Improved cortical structure segmentation. In: Kuba, A., Sámal, M., Todd-Pokropek, A. (Eds.). Information Processing in Medical Imaging: Proc. Int. Conf. (IPMI'99), Vol. 1613 of Lecture Notes in Computer Science. Springer, Berlin, pp. 210–223.

COVIRA. Multimodality medical image analysis for diagnosis and treatment planning: the COVIRA project Computer Vision in Radiology. Technical report, AIM Project A2003, January 1995.

Crubezy, M., Aubry, F., Moisan, S., Chameroy, V., Thonnat, M., Di Paola, R., 1997. Managing complex processing of medical image sequences by program supervision techniques. In: Proc. SPIE Conf. on Medical Imaging, Vol. 3035-85 of Proceedings of the SPIE. SPIE, Bellingham, WA, pp. 614–625.

de Boer, C., Smeulders, A.W.M., 1998. Simulating graphics for performance analysis purposes by combining real world background and synthetic foreground. In: Tombre, K., Chabra, A.K. (Eds.), Graphics Recognition, Algorithms and Systems, Lecture Notes in Computer Science. Springer, Berlin, pp. 347–358.

de Graaf, C.N., Koster, A.S.E., Vincken, K.L., Viergever, M.A., 1992. A methodology for the validation of image segmentation methods. In: Proc. Fifth Annual IEEE Symposium on Computer-Based Medical Systems. IEEE Computer Society Press, Los Alamitos, pp. 17–24.

Eberly, D., Pizer, S.M., 1994. Ridge flow models for image segmentation. In: Proc. SPIE Conf. on Medical Imaging, Vol. 2167 of Proceedings of the SPIE. SPIE, Bellingham, WA, pp. 54–64.

Elliot, P.J., Knapman, J.M., Schlegel, W., 1992. Interactive image segmentation for radiation treatment planning. IBM Systems Journal 31 (4), 620–634.

Falcão, A.X., Udupa, J.K., Samarasereka, S., Sharma, S., 1998. User-steered image segmentation paradigms: live wire and live lane. Graphical Models and Image Processing 60, 233–260.

Fontana, F., Virgili, P., Vottero, G., Casali, A.M., 1993. Multiseed interactive segmentation using different perspective angles for histological structures. In: Proc. SPIE Conf. on Medical Imaging, Vol. 1898 of Proceedings of the SPIE. SPIE, Bellingham, WA, pp. 247–251.

Gauch, J., 1999. Image segmentation and analysis via multiscale gradient watershed hierarchies. IEEE Trans. Image Processing 8 (1), 69–79.

Gerritsen, F.A., van Veelen, C.W.M., W.P.Th.M. Mali, A.J.M., Bart, H.L.T., de Bliek, J., Buurman, A.H.W., van Eeuwijk, M.J., Hartkamp, S., Lobregt, L.M.P., Ramos, L.J., Polman, P.C., van Rijen, P., Visser, C., 1995. Some requirements and experience with COVIRA algorithms for registration and segmentation. In: Beolchi, L., Kuhn, M.H. (Eds.), Medical Imaging Analysis of Multimodality 2-D/3-D Images, Studies in Health Technology and Informatics. IOS Press, pp. 5–27.

Gevers, T., Ghebreab, S., Smeulders, A.W.M., 1998. Color invariant snakes. In: Lewis, P., Nixon, M. (Eds.), Proc. Ninth British Vision Conf. British Machine Vision Association, pp. 578–588.

Gill, J.D., Ladak, H.M., Steinman, D.A., Fenster, A., 1999. Development and evaluation of a semi-automatic 3D segmentation of the carotid arteries from 3D ultrasound images. In: Proc. SPIE Conf. on Medical Imaging, Vol. 3661 of Proceedings of the SPIE. SPIE, Bellingham, WA, pp. 214–221.

Gleicher, M., 1995. Image snapping. In: Cook, R. (Ed.), Proc. SIGGRAPH'95, Annual Conference Series. ACM SIGGRAPH, pp. 183–190.

Griffin, L.D., Colchester, S.A., Roll, S.A., Studholme, C.S., 1994. Hierarchical segmentation satisfying constraints. In: Hancock, E. (Ed.), Proc. British Machine Vision Conf. BMVA Press, pp. 135–144.

Gülch, E., Müller, H., Läbe, T., Ragia, L., 1998. On the performance of semi-automatic building extraction. In: Proc. Int. Society for Photogrammetry and Remote Sensing (ISPRS) Commission III Symposium, Vol. 32 – Part 3/1 of International Archives of Photogrammetry and Remote Sensing. ISPRS, pp. 331–338.

Hastreiter, P., Ertl, T., 1998. Fast and interactive 3D segmentation of medical volume data. In: Niemann, H., Seidel, H.-P., Girod, B. (Eds.), Proc. Conf. on Image and Multi-dimensional Digital Signal Processing. infix, pp. 41–44.

Higgins, W.E., Reinhardt, J.M., Sharp, W.L., 1994a. Semi-automatic construction of 3D medical image-segmentation processes. In: Proc. Conf. on Visualization in Biomedical Computing, Vol. 2359 of Proceedings of the SPIE. SPIE, Bellingham, WA, pp. 59–71.

Higgins, W.E., Sharp, W.L., Hansen, M.W., Reinhardt, J.M., 1994b. A graphical user interface system for 3D image analysis. In: Proc. SPIE Conf. on Medical Imaging, Vol. 2164 of Proceedings of the SPIE. SPIE, Bellingham, WA, pp. 13–18.

Hinshaw, K.P., Altman, R.B., Brinkley, J.F., 1995. Shape-based models for interactive segmentation of medical images. In: Proc. SPIE Conf. on Medical Imaging, Vol. 2434 of Proceedings of the SPIE. SPIE, Bellingham, WA, pp. 771–780.

Höhne, K.H., Hanson, W.A., 1992. Interactive 3D segmentation of MRI and CT volumes using morphological operators. J. Comput. Assist. Tomogr. 16 (2), 285–294.

Houtsmuller, A.B., Smeulders, A.W.M., Van der Voort, H.T.M., Oud, J.L., Nanninga, N., 1993. The homing cursor: a tool for three-dimensional chromosome analysis. Cytometry 14, 501–509.

Kass, M., Witkin, A., Terzopoulos, D., 1987. Snakes: active contour models. Int. J. Computer Vision 1 (4), 321–331.

Keller, P., Keller, M., 1993. In: Visual Cues. IEEE Computer Society Press, Los Alamitos, CA.

Kostantinides, K., Rasure, J.R., 1994. The Khoros software development for image and signal processing. IEEE Trans. Image Processing 3 (3), 243–252.

Krivanek, A., Sonka, M., 1998. Ovarian ultrasound image analysis: follicle segmentation. IEEE Trans. Medical Imaging 17 (6), 935–944.

Lifshitz, L.M., Pizer, S.M., 1990. Multiresolution hierarchical approach to image segmentation based on intensity extrema. IEEE Trans. Pattern Anal. Machine Intelligence 12 (6), 529–540.

Maes, F., 1998. Segmentation and registration of multimodal images: from theory, implementation and validation to a useful tool in clinical practice. PhD thesis, Katholieke Universiteit Leuven, Leuven, BE.

Matsuyama, T., 1989. Expert systems for image processing: knowledge-based compositions of image analysis processes. Computer Vision, Graphics and Image Processing 48, 22–49.

McInerney, T., Terzopoulos, D., 1996. Deformable models in medical image analysis: a survey. Medical Image Analysis 1 (2), 91–108.

Mortensen, E.N., Barrett, W.A., 1998. Interactive segmentation with intelligent scissors. Graphical Models and Image Processing 60, 349–384.

Neuenschwander, W.M., Fua, P., Iverson, L., Székely, G., Kübler, O., 1997. Ziplock snakes. Int. J. Computer Vision 25 (3), 191–201.

Neufeldt, V., Sparks, A.N., 1995. In: Webster's NewWorld Compact School and Office Dictionary. MacMillan, Indianapolis, IN, USA.

Neugebauer, P., 1995. Interactive segmentation of dentistry range images in CIM-systems for the construction of ceramic interlays using edge tracing. In: Proc. Computer Assisted Radiology Conf. (CAR'95). Springer, Berlin, pp. 969–974.

Neumann, A., Lorenz, C., 1999. Statistical shape model based segmentation of medical images. Comput. Med. Imaging. Graph. 22 (2), 133–143.

Olabarriaga, S., December 1999. Human-Computer Interaction for the Segmentation of Medical Images. PhD thesis, University of Amsterdam, Amsterdam, NL.

Olabarriaga, S.D., Smeulders, A.W.M., 1997. Setting the mind for intelligent interactive segmentation: Overview, requirements, and framework. In: Duncan, J., Gindi, G. (Eds.). Information Processing in Medical Imaging: Proc. Int. Conf. (IPMI'97), Vol. 1230 of Lecture Notes in Computer Science. Springer, Berlin, pp. 417–422.

Olabarriaga, S.D., Smeulders, A.W.M., Marijnissen, A.C.A., Vincken, K.L., 1999. An intelligent interactive segmentation method for the joint space in osteoarthritic ankles. In: Kuba, A., Sámal, M., Todd-Pokropek, A. (Eds.). Information Processing in Medical Imaging: Proc. Int. Conf. (IPMI'99), Vol. 1613 of Lecture Notes in Computer Science. Springer, Berlin, pp. 394–399.

Olabarriaga, S.D., Pfluger, P.R., Smeulders, A.W.M., submitted. Piecewise DM: a locally controllable deformable model.

Pizer, S.M., Cullip, T.J., Frederiksen, R.E., 1990. Toward interactive object definition in 3D scalar images. In: Höhne, K.H., Fuchs, H., Pizer, S.M. (Eds.). 3D Imaging in Medicine – algorithms, systems and applications, Vol. 60 of NATO ASI Series. Springer, Berlin, pp. 83–105.

Schneiderman, B., 1997. In: Designing The User Interface. Addison-Wesley, Reading, MA.

Sequeira, J., Pinson, F., 1990. Matching free-form primitives with 3D medical data to represent organs and anatomical structures. In: Höhne, K.H., Fuchs, H., Pizer, S.M. (Eds.). 3D Imaging in Medicine - algorithms, systems and applications, Vol. 60 of NATO ASI Series. Springer-Verlag, Berlin, pp. 158–171.

Shyu, C., Brodley, C.E., Kak, A.C., Kosaka, A., 1999. ASSERT: A physician-in-the-loop content-based retrieval system for HRCT image databases. Computer Vision and Image Understanding 75 (1/2), 111–132.

Sijbers, J., Van der Linden, A., Scheunders, P., Van Audekerke, J., Van Dyck, D., Raman, E., 1996. Volume quantization of the mouse cerebellum by semi-automatic 3D segmentation of magnetic resonance images. In: Proc. SPIE Conf. on Medical Imaging, Vol. 2710 of Proceedings of the SPIE. SPIE, Bellingham, WA, pp. 553–560.

Sivewright, G.J., Elliot, P.J., 1994. Interactive region and volume growing for segmenting volumes in MR and CT images. Med. Inform. 19 (1), 71–80.

Smeulders, A.W.M., Olabarriaga, S.D., Van den Boomgaard, R., Worring, M., 1997. Design considerations for interactive segmentation. In: Jain, R., Santini, S. (Eds.), Proc. Conf. on Visual Information Systems (VISUAL'97). Knowledge Systems Institute, pp. 5–12.

Stiehl, H.S., 1990. 3D image understanding in radiology. IEEE Engineering in Medicine and Biology Magazine 9, 24–28.

Tieck, S., Gerloff, S., Stiehl, H.S., 1998. Interactive graph-based editing of watershed-segmented 2-D images. In: Workshop on Interactive Segmentation of Medical Images. Unpublished.

Treece, G., Prager, R., Gee, A., Berman, L., 1999. Volume measurement in sequential freehand 3-D ultrasound. In: Kuba, A., Sámal, M., Todd-Pokropek, A. (Eds.). Information Processing in Medical Imaging: Proc. Int. Conf. (IPMI'99), Vol. 1613 of Lecture Notes in Computer Science. Springer, Berlin, pp. 70–83.

Udupa, J., 1982. Interactive segmentation and boundary surface formation for 3D digital images. Computer Graphics and Image Processing 18, 213–235.

Udupa, J.K., Herman, G.T., 2000. In: 3D Imaging in Medicine. CRC Press, Boca Raton, FL.

Udupa, J.K., Wei, L., Samarasekera, S., Van Buchem, M.A., Grossman, R.I., 1997. Multiple sclerosis lesion quantification using fuzzy-connectedness principles. IEEE Trans. Medical Imaging 16 (5), 598–609.

Upson, C. et al., 1989. The application visualization system: a computational environment for scientific visualization. IEEE Computer Graphics and Applications 9 (4), 30–42.

Vehkomäki, T., Gerig, G., Székely, G., 1997. A user-guided tool for efficient segmentation of medical data. In: Troccaz, J., Grimson, E. (Eds.). Proc. Conf. on Computer Vision, Virtual Reality and Robotics in Medicine and Medical Robotics and Computer-Assisted Surgery (CVRMed-MRCAS'97), Vol. 1205 of Lecture Notes in Computer Science. Springer, Berlin, pp. 685–694.

Wink, O., Zuiderveld, K.J., Viergever, M.A., 1997. Interactive volume segmentation using local similarity measurements. In: Lemke, H.U., Vanier, M.W., Inamura, K. (Eds.), Proc. Computer Assisted Radiology Conf. Elsevier, Amsterdam, p. 996.

Worth, A.J., Makris, N., Meyer, Jr. J.W., Caviness, V.S., Kennedy, D.N., 1997. Automated segmentation of brain exterior in MR images driven by empirical procedures and anatomical knowledge. In: Duncan, J., Gindi, G. (Eds.). Information Processing in Medical Imaging: Proc. Int. Conf. (IPMI'97), Vol. 1230 of Lecture Notes in Computer Science. Springer-Verlag, Berlin, pp. 99–112.

Xiaohan, Y., Yla-Jaaski, J., 1995. Interactive surface segmentation for medical images. In: Proc. XVIII Conf. on Applications of Digital Image Processing, Vol. 2564 of Proceedings of the SPIE. SPIE, Bellingham, WA, pp. 519–527.

Yasnoff, W., Mui, J., Bacus, J., 1977. Error measures for scene segmentation. Pattern Recognition 9, 217–231.