Acquiring Object Models Using Vision Operations

Tilo Messer

Bayerisches Forschungszentrum für Wissensbasierte Systeme (FORWISS)
Forschungsgruppe Kognitive Systeme
Orleansstr. 34, W - 8000 München 80
e-mail: messer@forwiss.tu-muenchen.de

Abstract

In this paper, a new approach is introduced to acquire models of man-made, flat industrial parts by generating automatically image interpretation algorithms. In a gray-scale image, a closed contour describes the real object roughly. Values of pixels in the near environment of the contour supply the generation of the algorithms. Two masks are built from the contour: a filled region and a narrow stripe whose skeleton is the contour itself. Regarding the two masks the resulting regions of the algorithms are matched. The better the regions match the given mask the higher the algorithm is assessed. The highest ranked algorithm of each list is selected. Their resulting regions are used to form a new closed contour. The new improved contour describes the object more exactly. The whole process, starting from the generation of image interpretation algorithms and ending with the creation of a new contour, can be repeated until a stable state is achieved.

1 Introduction

For a few years, the number of researchers who deal with combining computer vision and artificial intelligence has been growing. Main topics are image understanding support systems and computer vision expert systems. [Nazif & Levine 84] worked on an intelligent rule-based segmentation system. Much work has been done to support the user: beginner or professional, to handle an image processing tool ([Matsuyama 86], [Matsuyama 89], [Grimm & Bunke 89]). [Ender 87] describes a model-based adaption of parameters within a fixed set of vision programs (see also [Liedtke & Ender 86]). His approach requires the searched-for interpretation (i.e. the model) of the object, which should be identified, as a necessary assumption. There are also two ESPRIT-projects whose participating firms and institutes investigate the automatic programming of vision systems (VIDIMUS[1] and MUSIP[2]). However in general, an automatic evaluation of the generated vision system has turned out to be too difficult and complex. Thus, only the computer vision expert is able to modify the generated resulting systems. Even a user who is familiar with the application is hardly able to handle these problems as she/he has little expertise in computer vision, and probably only a minimum of expertise in the implementation details of the based image processing tool.

In our approach, we intend to generate image interpretation algorithms automatically in a knowledge based manner. As an important aspect we incorporate common sense computer vision reasoning even into the very first steps of the generating process. Knowledge about expectations, which are explicitly described, is used in the whole process.

Now, we discuss the different contours and masks which are used for the evaluation of the generated algorithms and the creation of the new improved contour. The first step of the process is the creation of a *initial closed contour.* This can be done automatically or interactively by the user. Fig. 2b shows an example of such a contour. The original gray-scale image is shown in Fig. 2a. Next, two masks are formed by using this contour: a *region-like mask* (Fig. 2c) and a *contour-like mask* (Fig. 2d). The more exact the resulting regions of the generated image interpretation algorithms match one of both masks the higher the algorithm is evaluated. The result of the highest ranked algorithms is shown in Fig. 2e, respectively in Fig. 2f. The different masks force the selection of two different algorithms. A smooth merging of both resulting regions leads to a *new closed contour* (Fig. 2g). This contour replaces the mentioned initial closed contour, and the process can be started again until a stable state is achieved. The new closed contour describes the real

1. VIDIMUS: A generic vision system for industrial applications; project id. 2592.
2. MUSIP: Multi-sensor image processing; project id. 2316.

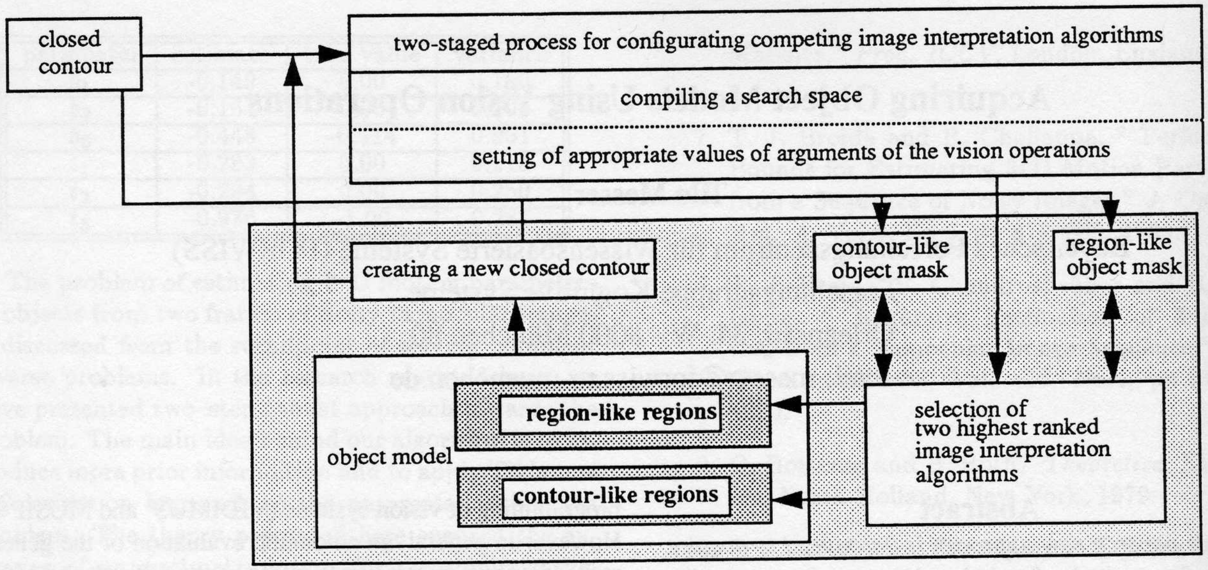
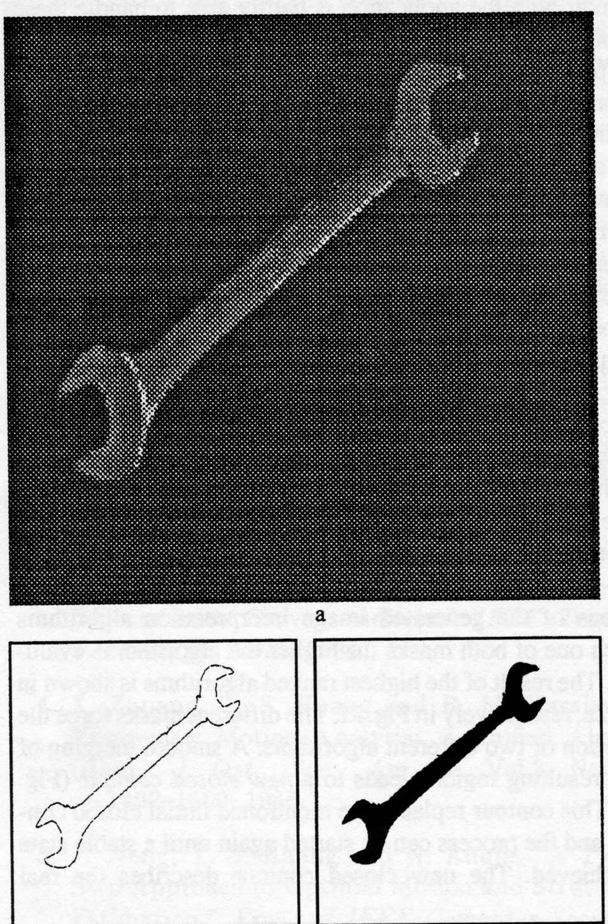


Fig. 1 The control structure of our model acquisition system: the arrows show the direction of data and control flow.



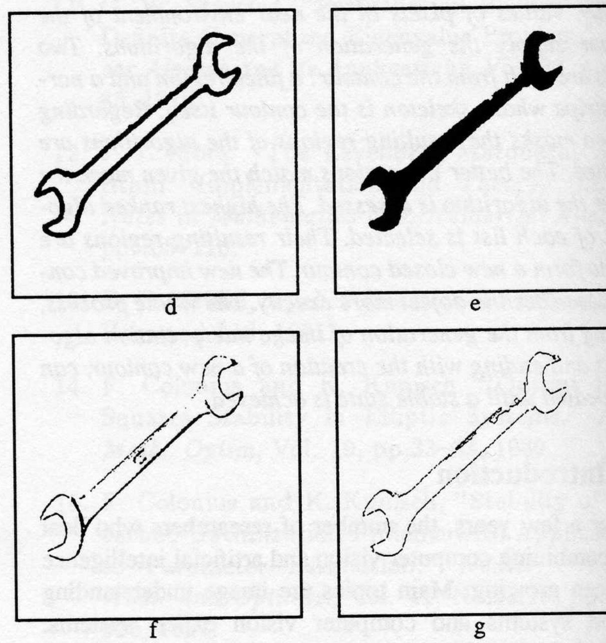


Fig. 2 a: original grayscale image; b: automatically generated initial closed contour; c: region-like object mask; d: contour-like object mask; e: best evaluated result regarding criterion area; f: best evaluated result regarding criterion contour; g: created new closed contour;

object more exactly. The selected image interpretation algorithms, however, solve the segmentation problem by using general vision operations of an underlying image processing tool.

The paper is organized as follows: first, we briefly describe the knowledge bases which incorporate detailed information about vision operations and knowledge as to determine appropriate values for their arguments. In the next section, the initial mask which is used as the description of the vision task is discussed. Next, the evaluation process of the generated image interpretation algorithms is described. Finally, we demonstrate some results.

2 Generating image interpretation algorithms

An important assumption for our approach is the existence of a set of competing image interpretation algorithms. We define a *image interpretation algorithms* as a *sequence of vision operations* which transform an original grayscale image into a segmented image. This image may consist of more than one connected components. We call them *regions*.

In our system, a two-staged process controls the generation of the image interpretation algorithms (Fig. 1). In the first layer a *search space* of all possible algorithms is constructed. The underlying vision operations are represented in a frame-based knowledge base [Polensky & Messer 89]. Various rules and specific relationships between the operations restrict the space of the image interpretation algorithms to be generated and therefore determine the size of the search space of the generating process. Static restrictions determine the order of vision operations within an image interpretation algorithm.

An example of such a rule is:

> *If an algorithm consists of a dynamic segmentation operation, then a smoothing operation is suggested to be applied before the segmentation algorithm.*

Regarding those kinds of rules it is obvious that they also refer to classes of vision operations. This is an advantage of the underlying frame-based knowledge base. The hierarchical inheritance allows the transmission of properties of classes of vision operations to the vision operations themselves as elements of classes.

The second layer of the generating process mainly works on the *appropriate setting of values of the arguments* of the vision operations. The arguments of the vision operations are also represented in a frame-based knowledge base. Arguments are image objects (i.e. (original) input data and output data, as well as masks) and controlling values (e.g. sizes of matrices, values of threshold, factors). The knowledge base contains a rather important and explicitly represented part of common computer vision expertise of the model acquisition system. Rules which control how arguments of vision operations are set are included in the knowledge base.

The knowledge about how arguments are set appropriately originates from various sources. On the one hand, a lot of *generally valid computer vision expertise* is incorporated into rules. The knowledge is independent of a specific image processing tool serving as a basis of the generating process. On the other hand, a lot of *tool-specific knowledge* is necessary for the appropriate setting of the arguments. How specific controlling arguments have to be set in order to get the desired result is dependent on the image processing tool. This serves as a basis of our model acquisition system (e.g. the dynamic segmentation operation, which is used in the image processing tool HORUS ([Eckstein 91]) and is called *dyn_threshold*, with its arguments *input image object*, *output image object*, *input grayvalue component* (e.g. original image), *compared grayvalue component* (e.g. smoothed grayscale image), *add* (value which changes the difference of all pixels between the input grayvalues and the compared grayvalues)).

To adapt values of arguments to a given vision task, for instance to recognize a specific object in an image, is not a trivial task. In most cases, computer vision experts find an appropriate set of values by processing 'trial-and-error'! They evaluate various values. In order to work on a well-structured knowledge about the setting of values for arguments it is necessary to acquire it appropriately, represent it in a clear structure, and perhaps, formalize it. This is a problem which cannot be underestimated. Actually, the knowledge about the functionality of vision operations can be found in literature, but the representation of this knowledge is mostly not appropriate to be transformed into a knowledge base how we need it. As an example, an exact mathematical formulation of a convolution does not directly lead to the sizes of the matrices depending on a given vision task. Another difficulty is the distributed and differing representation of the needed knowledge. Therefore, the knowledge is best acquired from a computer vision expert.[3]

3 Creating an initial closed contour

Experiments have shown that a complete and useful setting of values can be achieved by using some dominant features of a given closed contour which roughly describes the real object. It does not play an important role, however, as accurate the given contour is.[4] Only some conditions should be met by the contour. These requirements are explained when we will discuss the evaluation of the image interpretation algorithms. The closed contour should enclose the object in the image. The setting of most values

3. The problems involving in knowledge acquisition are sufficiently known. It is helpful first, if an interview is rather short, second, if the questions are precisely put and finally, if the knowledge is able to be acquired within several sessions.

4. A contour is called accurate if the contour matches the boundary of the real object in the grayscale image.

of arguments bases on an analysis of the grayscale values of the pixels in the near environment of the contour. These pixels are called *adaption space*. This space helps us to compute the average gradient of the object edges, its length and the average brightness of the object and of the background.

The initial closed contour can be given interactively by a user, or it can be build up by a simple automatic difference-based procedure. The automatic generating process of the initial closed contour requires the original image, which shows only one object, as well as the background image without any object. Then, the grayscale values of the pixels of both images are mutually subtracted. Finally, both resulting images are combined to form the initial closed contour.

Let O be the original image, B the background image, and I an intermediate grayscale image. Let $g_j(O)$ be the grayscale value of pixel j in the original image; $g_j(B)$ and $g_j(I)$ are defined analogous. Let $\theta_{min,max}$ be a threshold function with minimum value *min* and maximum value *max*. Then, the following equations hold for the initial closed contour C:

$$\begin{aligned} \forall j \in O, B: g_j(I) &= |g_j(O) - g_j(B)| + |(g_j(B) - g_j(O))| \\ C &= \theta_{\min,\max}(g_j(I)) \end{aligned} \qquad \text{(Eq 1)}$$

This equation leads to an initial closed contour which depends on image quality and noise ratio of both images, on the contrast between object and on the homogeneity of the background image. An example of an initial closed contour is shown in Fig. 2b.

4 Evaluation criteria

As a next step, two different *object masks* are generated in order to prepare the evaluation of the generated image interpretation algorithms. Our aim is to select two different image interpretation algorithms among all. Therefore, we form two different masks from the initial, respectively given, closed contour. The *region-like object mask* is build by filling the closed contour (see Fig. 2c). The *contour-like object mask* is built by dilating the closed contour on a narrow stripe whose skeleton is the contour itself. The application of a morphological operation with a small structuring element (we use a circle with an radius dependent on an estimated length of the gradient of the boundary of the real object) forms the narrow stripe (see Fig. 2d).

The generated image interpretation algorithms are evaluated depending on the following two evaluation criteria which correspond to both masks:

- criterion *area*: select that algorithm whose resulting regions match best the given region-like object mask.
- criterion *contour*: select that algorithm whose resulting regions match best the given contour-like object mask.

The evaluation function includes the following measures of the resulting regions and the object mask in its computation: let A be the area of the region, which results when the closed contour I is filled up; regarding the criterion *contour* let A be the area of a narrow stripe around the given contour of the object mask; let AR be the area of the resulting regions and let AC be the common area of AR and A: then, the *hitratio* is defined as follows:

$$hitratio = \frac{AC}{A + AR - AC} \qquad \text{(Eq 2)}$$

The evaluation regarding both criteria mentioned above leads to the selection of a *region-based* image interpretation algorithm and of an *edge-based* algorithm.

It is not obvious that the matching of the resulting regions with the given two masks which are quite definitively not accurate (resp. it is not known how accurate they are), leads to reasonable results. Nevertheless, some *features* of the object masks let our approach seem to be reasonable. The features are given consciously imprecise, such as follows:

- The outer contour of the region-like object mask approximately lays in the near environment of the boundary of the real object.
- Parts of the region-like object mask should be quite accurate.
- The length of the skeleton of the contour-like object mask nearly fits the real length of the outer boundary of the real object.

Now, an important intermediate result is achieved. Dependent on an initial closed contour which describes a man-made flat industrial part, a set of image interpretation algorithms are automatically generated. Two of them are selected by a domain independent evaluation function. These two image interpretation algorithms solve the segmentation problem regarding the surface and the boundaries of the object in the image.

5 Modifying the closed contour

The next step, the modification of the given closed contour starts with both resulting regions of the highest ranked image interpretation algorithms. In order to preserve the global outer shape of the closed contour the new contour is created by preserving the shapes of the resulting regions.

The creation of the new contour starts with the following resulting regions: let R be the resulting region of the image interpretation algorithm regarding the criterion *area*, let C be the resulting region regarding the criterion *contour*; the steps of the *modification algorithm* are described as follows:

1. compute the outer contour of R; call them OCR;
2. let the OCR getting wider inwards (see Fig. 3a); call this IR;
3. merge IR with C; call the resulting region U (see Fig. 3b);
4. fill up U; call it FU (see Fig. 3c);
5. if FU has more than one connected components, create a compact region FUC which exists of only one connected component; in order to create a compact region the smallest connected component of FU is enlarged till it touch another connected component;
6. if FU has more than one connected component, go to step 4;
7. compute the outer contour of FUC (see Fig. 3d).

This process stops if there is only one connected component. The advantage of the mentioned enlargement is that the global shape of the contour can be preserved, because the small connected components are assumed within the outer contour of the region-like object mask. The new contour is itself in turn appropriate to serve as input for a new configuration cycle of a set of image interpretation algorithms.

It turned out that the termination of the whole process is in fact a problem. In our experiments, we have observed that the evaluation values and the generated image interpretation algorithms achieve a stable state.

6 Results

The following examples of generated image interpretation algorithms demonstrate the behaviour of the system. Fig. 4a shows the original grayscale image, seen in Fig. 2a, and the final closed contour (Fig. 4b), after the process achieved a stable state after the forth cycle. The further evaluation values oscillate continuously, but the highest ranked image interpretation algorithms and their arguments do not change. Fig. 4c and Fig. 4d show the resulting regions of the best evaluated algorithms regarding the criteria area and contour. Fig. 4e-h show another example: original grayscale image (Fig. 4e), the created final contour after the ninth cycle (Fig. 4f), and the resulting regions regarding the criteria area (Fig. 4g) and contour (Fig. 4h).

The system is implemented in IF/Prolog. We use a set of about 70 image interpretation algorithms. Each cycle runs in about five minutes, including I/O and full application of the used vision operations. The resulting regions of the highest ranked algorithms serve as input data of an object recognition system based on morphological operations.

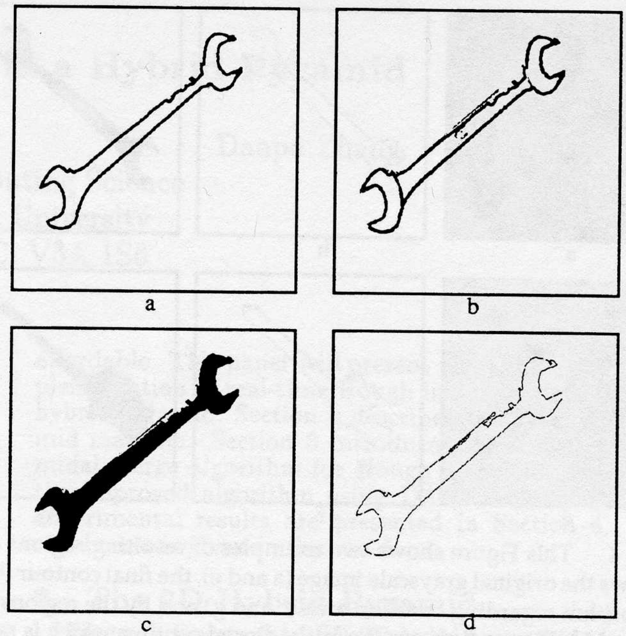


Fig. 3 This Figure shows the intermediate results during the creation of a new contour: a: the inwards dilated outer conter of the resulting regions of the best evaluated algorithm regarding criterion area; b: union of that outer contour and the resulting regions of the best evaluated algorithm regarding criterion contour; c: filling up this region; d: the resulting new contour.

7 Conclusion

In this paper, we describe a process which allows us to generate automatically image interpretation algorithms. An (initial) closed contour describes a real object in a grayscale image. We are able to improve the given contour by an iterative application of the generating process. The iterative application generates a set of image interpretation algorithms. Two of them are selected regarding two evaluation functions which are applied to their resulting regions. The evaluation criteria refer to area and to contour. The process terminates if a stable state is achieved.

The termination of the process is a known problem. Our process cycle does not stop automatically. A stable state is usually achieved after about 10 cycles. For an automatic model acquisition system it is necessary that an appropriate termination criterion is realized.

The rules, which control how the argument values of the vision operations are set, are not depending on the application and are formulated based on acquired computer vision expertise. What is even more, the evaluation function is also independent of the application. Thus, the

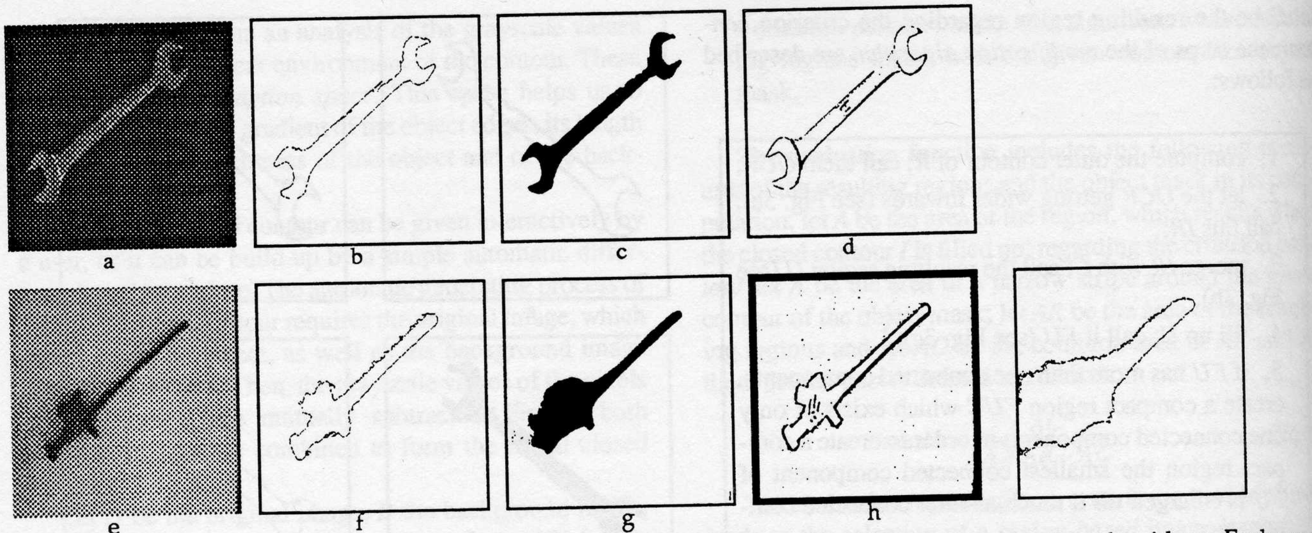


Fig. 4 This Figure shows two examples of resulting regions of best evaluated image interpretation algorithms. Each row shows the original grayscale image (a and e), the final contour (b and f), which is build up by the regions of the best evaluated algorithm regarding criterion area (c and g) and by the regions of the best evaluated algorithms regarding criterion contour (d and h). Figure 4i shows the initial closed contour which is rather inexact.

described model acquisition system itself is independent of a special application and is appropriate to serve as a kernel for a more complex expert system for the generation of image interpretation systems.

Acknowledgements

We gratefully acknowledge the constructive and helpful comments of Bernd Radig. Many thanks also go to Elisabeth and to Andrew Verden for revising this paper.

References

[Eckstein 91]: Wolfgang Eckstein: HORUS Referenz-Manual. Version 3.08, Institut für Informatik der Technischen Universität München, April 1991.

[Ender 87]: Manfred Ender: Ein Beitrag zur automatischen wissensbasierten Konfiguration von Bildinterpretationssystemen, Fortschritt-Berichte VDI, Reihe 10: Informatik/ Kommunikationstechnik Nr. 68, Düsseldorf, 1987.

[Grimm & Bunke 89]: Felix Grimm, Horst Bunke: An Expert System for Software Configuration and its Application to Computer Vision, in: Applications of Artificial Intelligence VII, March 28-30, 1989, Orlando, SPIE Vol. 1095, pp. 173 - 183.

[Liedtke & Ender 86]: Claus-E. Liedtke, Manfred Ender: A Knowledge Based Vision System for the Automated Adaption to New Scene Contents, in: 8. International Conference on Pattern Recognition, Oct. 27-31, 1986, Paris, pp. 795 - 797.

[Matsuyama 86]: T. Matsuyama: Expert Systems for Image Processing: An Overview, in: IT Young et al. (Ed.), Signal Processing III, New York, 1986, pp. 869 - 872.

[Matsuyama 89]: Takashi Matsuyama: Expert Systems for Image Processing: Composition of Image Analysis Processes, in: Computer Vision, Graphics, and Image Processing 48, 1989, pp. 22 - 49.

[Nazif & Levine 84]: A.M. Nazif, M.D. Levine: Low Level Image Segmentation: An Expert System, in: IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-6 (5), 1984, pp. 555 - 557.

[Polensky & Messer 89]: Gudrun Polensky, Tilo Messer: Ein Expertensystem zur frame-basierten Steuerung der Low- und Medium-Level-Bildverarbeitung, in: 11. DAGM-Symposium 89, Oct. 2-4, 1989, Hamburg, pp. 406 - 410.