A Knowledge Based System for the Configuration of Image Processing Algorithms

Ursula Rost, Heiko Münkel, C.-E. Liedtke
Institut für Theoretische Nachrichtentechnik, Universität Hannover
Appelstr. 9a, 30167 Hannover

Abstract

During the last decades image processing systems have been integrated in various automation processes. It has been estimated that only 5% of the automation tasks qualified for image processing systems have so far been realized. This is partly due to the required high engineering effort which is needed to adapt the systems to varying viewing conditions, like changes in illumination, noise artefacts, changes in the surface properties of the observed objects, etc. and to the effort required for the adaptation of the systems to new tasks. This engineering effort could be considerably reduced, if the systems could to some extent adapt themselves automatically. These considerations have led to the development of the knowledge-based system SOLUTION, which is able to adapt low-level image processing algorithms automatically to varying needs. SOLUTION accepts a task description presented in high level terms, configures an appropriate sequence of image processing operators and adapts the free parameter values. SOLUTION therefore uses expert knowledge formulated explicitly by rules. In its present implementation the system is limited to the extraction of contours. The extraction of regions is under development. The results presented here refer to the general task of creating an "optimal" contour-image that can be further processed by subsequent high-level routines. Examples are taken from applications within the areas of medical and industrial image processing.

1 Introduction

In the last 10-15 years image processing systems have begun to play an increasing role within many automation processes. These systems are used to monitor or even control the manufacturing process or to inspect the final product for quality assessment (see for example [1]). With the growing performance of todays microprocessors the implemented image processing algorithms tend to increase in complexity. They often consist of several simple and well known image processing operators where each of them possesses one or more free parameters. Examples for such operators can be found in many books, e.g. in [2]. The operators included in the processing paths and their parameter values differ usually from one processing task to the other depending on the objects to be examined, the structure of the surrounding scene, the expected result and some other general requirements as for example constraints on the processing time. This has led to the situation that most image processing systems are designed to fulfill exactly one task and are thus very specialized. The design process has to be performed by an image processing expert who chooses adequate operators and adjusts their parameters.

The problem that arises lies within the lacking flexibility of the resulting systems. Even slight changes in the environment like for example illuminating effects, changes in the surface properties of the observed objects, extra noise in the image, etc. can cause the algorithm to fail to produce the expected result. In this case recalibration is needed. Bigger changes evoked by the demands of different tasks often require a complete reprogramming of the system. Because of the complexity of these systems a large amount of engineering effort becomes necessary to perform the required changes. This represents an expensive and time consuming factor, which reduces the acceptance of image processing systems considerably.

The system SOLUTION (Solution for a Learning Configuration System for Image Processing), that is described in this paper tries to overcome this problem by two means:

First it strictly separates the task-specific part of the problem to be solved from the image processing part. The user can specify the task using natural spoken terms which refer to general signal properties of the images to be processed, expected qualities of the processed image and supplementary general information like timing requirements, acccuracy of contour position, etc.. The relevant signal properties of the images to be processed can be expressed verbally or be determined by automatic evaluation of a test image within an interactively defined region of interest.

Second SOLUTION uses explicitly formulated expert knowledge to carry out the image configuration part of the problem. SOLUTION provides a set of image processing operators and performs a knowledge-driven choice to select the appropriate operators, combines them to processing paths and adapts their parameter values. The structure of SOLUTION is based on the ideas developed for CONNY [3, 4] but differs from it in many ways. SOLUTION is much more flexible concerning the task specifications and allows a broader variety of different goals. Another important difference lies within the rules used. The rules that contain expert knowledge about how to use the image processing operators are strictly separated from the rules that contain control information such as for example formal dependencies between different operators. Therefore it is much easier to perform changes in the rulebase e.g. to add or delete different rules without risking that the adaptation process will fail.

Section 2 presents the overall structure of SOLUTION. The evaluation mechanism which is used to check whether a task is fulfilled is described in Section 3. Section 4 describes the process of selecting appropriate image processing operators and section 5 explains the way in which the values of the parameters they contain are adjusted. Results that have been achieved for tasks taken from the areas of medical and industrial image processing are presented in section 6.

2 The structure of SOLUTION

The basic idea of SOLUTION is to create a system that in some way acts like a human image processing expert. That means, depending on a given task, the system should be able to select a sequence of appropriate image processing operators and to adjust the parameters they contain, so that the result has a defined quality. It is thereby assumed that the system owns a sufficiently large set of operators and that it has the ability to evoke their execution. It is further assumed that the quality to be reached can be specified by a certain number of properties the resulting image is expected to have and which can be measured automatically. Presently SOLUTION is used to configure the low level part of an image processing system. The purpose is to provide succeeding high-level routines with an input of even quality. The examples presented in the remainder of this paper will refer to the image processing task of creating a contour image. The quality of the contour image is aimed to represent the best possible compromise regarding all criteria in the task specification. The properties taken to measure the obtained image quality refer to visible features like the thickness of the contours, the fragmentation, or their frayness.

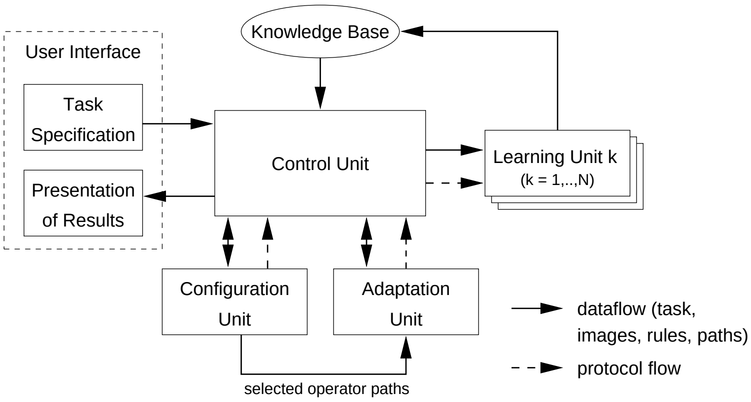


Figure 1: Structure of SOLUTION

Figure 1 shows the general structure of SOLUTION. The control unit in the center is responsible for the coordination of all system activities. It provides all other modules with the information they need and controls the data and protocol flow.

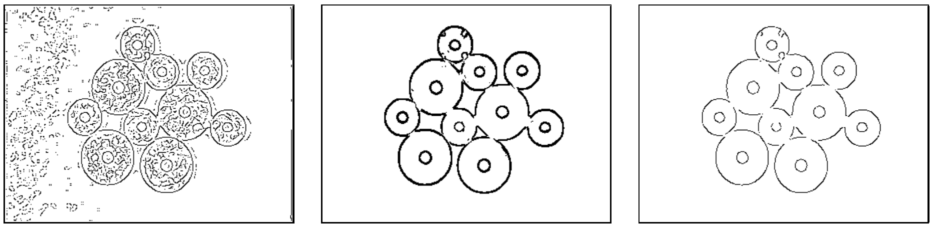
The image processing task that has to be performed can be specified via a graphical user interface which is also responsible for presenting the results that have been achieved. The task description itself contains two major parts, an explicit and an implicit one.

First of all the user has to specify a couple of explicit properties the result is expected to have. This refers on the one hand to some general constraints concerning the choice of the operators, like constraints on the processing time or the accuracy of edge positions which have to be detected. On the other hand the resulting image should have a defined quality which can be specified by a couple of symbolic parameters quantized in a few values where both, the parameters and the quantized parameter values, can be expressed in natural spoken terminology. An example is that the "thickness" of the contour-lines has to be “weak” and that their “frayness” should be “at-most” “average”.

The implicit part of the task specification is given by some signal properties of the input image, respectively the class of images that have to be processed like the noise or the proportion of disturbances in the images. These properties are either expressed verbally or are measured from a representative test image with the aid of the user who can specify regions of interest (ROIs) such as a rectangle within the background in which the variance is computed or a line perpendicular to an important edge for which the height and the ascendant width are computed.

The configuration unit takes the information gained from these specifications or measurements to select appropriate image processing operators. That means that only those operators are chosen which are known to work well on images with the specified properties. The operators are combined in every way that is formally correct to form so-called image processing paths. The parameters of the processing paths are initialized according to the included operators using again the properties of the test image. The configuration as well as the initialization are performed by rule inference processes. The rules therefore used contain information about how to perform reasonable decisions about the selection of image processing operators. The resulting paths are then passed to the adaptation unit.

In contrast to the configuration the adaptation is an iterative process performed separately for each path. The parameters included in every path are corrected stepwise until the properties of the resulting image match the respective task specifications. Therefore rules are used which contain expert knowledge specifying the parameter changes which have to be performed in order to obtain desired changes of one or more of the properties of the resulting image.



Thickness: not	Thickness: average	Thickness: not
Distinctness: weak	Distinctness: strong	Distinctness: strong
Granulation: strong	Granulation: not	Granulation: not

Figure 2: Example of contour images with corresponding description in natural spoken terms.

As it is very difficult and time consuming to create manually a complete and consistent rulebase that meets the needs of all possible tasks, SOLUTION contains some learning units which support the process of building and maintaining different parts of the rulebase. The units developed until now are able to learn adaptation rules by goal-directed experimentation and by a special knowledge-based genetic algorithm. These algorithms are described in [5] and will not be discussed here further.

3 Sensors

The sensors used in SOLUTION are procedures that are able to measure the image properties needed to make decisions about the choice of operators and the adaptation of the parameters. All sensors are assigned to one of two general classes which represent the role of the image they work on. The first class contains all sensors that measure properties of the input image, e.g. the mean noise, the amount of disturbances, the edge height or the ascendant width of selected lines in the image. The second class comprises sensors which measure features of the resulting image. The sensors needed for the special purpose of evaluating contour images are grouped in the subclass of binary sensors.

Figure 2 shows three examples of contour images and the values of the sensors "thickness", "distinctness" and "granulation". The thickness is used to determine the amount of thick lines in the image. The value is derived from the number of points that are eliminated by a thinning algorithm without counting isolated points or small point groups. The distinctness refers to the lines and ellipses that can be detected in the result image and is defined by the ratio of points in the image that lie on such objects divided by all points in the image. The granulation specifies the proportion of small point groups (grains) in the image.

The sensors used for the input images describe facts about signal properties of the class of images that is to be processed. They are mainly needed for the selection of appropriate operators and the initialization of their parameter values. In contrast to this the output image sensors describe the appearance of the result which is task-dependant. For this reason the sensors are strongly related to the task definition performed by the user who defines which values these sensors should take for a resulting image of sufficient quality. The advantage of this lies in the possibility to use the differences between the parameter values specified and the parameter values measured for a concrete path to control the adaption process.

4 Configuration

As described in section 2 the configuration unit is responsible for the selection of appropriate image processing operators and for combining them to so-called image processing paths.

Furthermore it initializes the values of the parameters included in the selected operators. Figure 3 shows the structure of the configuration process.

All decisions about the choice of operators and the initial settings of the parameter values depend on the properties of the input image and some general requirements. The values of these properties such as the noise, the amount of disturbances, the height and the ascendant width of the edges in the image can be specified explicitly by the user or interactively measured by defining ROIs. In the first case the user has to tell the system the values of the properties. In consideration of the unability of humans to express image qualities precisely on a continuous scale or to obtain precise values for the same quality from different people only four possible choices for the parameter values are provided, which are "not" "weak" "average" and "strong". Examples for specifications using these values are "the noise in the image is weak" and "the ascendant width of the edges is average".

In addition to this there are some general requirements for which the user is asked to give a priority value which has to be one of the values "necessary", "important", "unimportant" or "don't care". The requirements refer on the one hand to some global constraints as for example the importance of a short processing time and on the other hand to a couple of properties the resulting algorithm should have and which affect the choice of the operators such as the importance of obtaining the accurate positions of edges or vertices. Example specifications of these priorities are "it is important to get accurate edge positions" and "it is necessary to have a short processing time". The selection of operators is performed by a rule inference process where each rule consists of a condition part including one or more of the specifications described above and an action part which removes exactly one operator from the list of possible operators that are known to the system. Examples for such rules are:

if the ascendant width of the edges is <= weak
then don't take the Ranking operator

if the noise is <= weak **and**
a short processing time is >= important
then don't take the Nonlinear Laplace operator

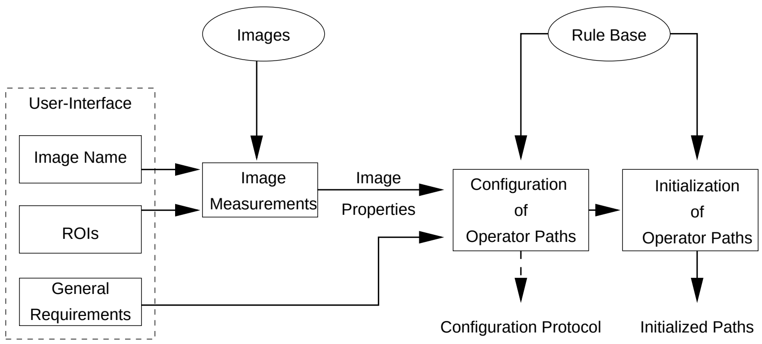


Figure 3: Configuration of operator paths

The remaining operators are taken to construct image processing paths. Therefore the operators are combined in every possible way. This process is constrained by the working principles and the purpose of the operators involved. The operators used by SOLUTION are divided into several groups with each representing one possible step with a special purpose within a whole image processing path. The four groups which are defined for the overall task of creating a contour image and the implemented corresponding image processing operators

Table 1: Implemented image processing operators

purpose	implemented operators
preprocessing	Median, Gauss, Average
contour enhancement	Sobel, Roberts Nonlinear Laplace Morphologial Contour
contour finding	Ranking, Binarize
postprocessing	Pratt

are shown in Table 1. Backward chaining rules are used to specify the structure of a valid path. An example for a rule which determines that every path has to contain a contour-enhancement operator but that in case the Nonlinear Laplace operator has been chosen no further operator is allowed is shown in the following:

An operator path is correct
if a preprocessing operator is in the path
and if the contour enhancement operator in the path is the nonlinear Laplace operator
and if no contour finding operator is in the path
and if no postprocessing operator is in the path

After the processing paths have been constructed the parameter values of the operators have to be initialized. The initialization depends on the one hand on the properties of the input image specified in the task description and on the other hand on some expert knowledge about the interrelation of parameter values for special operator combinations. For example the convolution with kernels of increasing window size as used by the Roberts and the Sobel filters yield gradient images with local extrema increasing by a factor 4. This kind of knowledge is also represented by forward chaining rules. The initialization is carried out as a rule inference process. The following two examples show the structure of these rules:

if the edge height is weak **and**
the path contains a Sobel operator **and**
the path contains a Binarize operator
then set the threshold of the Binarize operator to 32

if the edge height is weak
the path contains a Roberts operator
the path contains a Binarize operator
then set the threshold of the Binarize operator to 8

Although the following adaptation process should be able to cope with every possible parameter combination a good initialization usually makes it work faster and more reliable.

5 Adaptation

The process of adaptation imitates the strategy of an image processing expert. The expert knows what the resulting image has to look like. He adjusts the parameters of an operator path stepwise by changing some selected parameter values. According to the results thereby achieved he decides whether further changes have to be made. If for example the contours found are still too thick he will increase the value of the binarization threshold by a certain amount in order to reduce the contour width.

Figure 4 shows the adaptation process. The input needed is received by the control unit and consists of a test image to be processed, the initialized path P_0 created by the configuration unit and the description of a global goal. This goal includes all task specifications referring to the properties which the resulting (contour) image is expected to have and which can be measured by the sensors described in section 3. Each part of the global goal refers to exactly one sensor and specifies which value it should reach, in which way and with which priority. A global goal specification could be that it is "necessary" that the "thickness" of the detected contours is "equal to" "weak" and that it is "important" that their "frayness" is "at most" "average".

In the first step the image processing part is performed, i.e. the path P_0 is processed and an output image I_0 is created. Afterwards this image is evaluated by the sensors and the measured values are compared to the values defined in the global goal. From the difference for each item in the task specification a so called local goal is created which describes a sensor change that has still to be achieved in order to reach the global goal. If for example the "thickness" of the lines is in fact "average" but the global goal demands that it has to be "weak", a local goal is created that states that the "thickness" should be "decreased weakly".

The inference process is carried out using rules which represent the knowledge of an image processing expert who knows which parameters have to be changed to which extent in order to reach one or more local goals dependant on the present appearance of the image. The rules are therefore constructed such that the condition part contains the local goals and, in addition to this, some specifications concerning the state of the image in which the rule can be applied. The state is defined by the properties of the resulting image and is measured by the sensor values. The action part of each rule consists of exactly one parameter change. The value of change appearing in the rules can be either "weak" or "strong". The numeric values hidden behind this depend on the parameters themselves and vary according to their possible ranges. Two example adaptation rules are given below:

if the fragmentation should be decreased weakly **and**
the thickness is less than strong
then increase the threshold of the Binarize operator weakly

if the fragmentation should be decreased weakly **and**
the thickness is strong
then decrease the threshold of the Binarize operator weakly

Depending on the chosen adaptation strategy one or more rules can fire in step i. Just as an experienced user is able to change several parameters at a time one possible strategy is to permit the rules for each appearing local goal to act simultaneously. A more cautious strategy applies only one rule per step performing thus only one parameter change per step. In the latter case possibly more steps are needed to reach the global goal, but as the results of the changes can be tracked more precisely this strategy is especially advantegeous during learning.

The parameter changes performed lead to the creation of a new path $P_{i+1} (i = 0, \ldots, n-1)$ which is used in the next step of the adaptation loop. The process continues until the difference to the global goal is sufficiently small to ensure a satisfactory result which is then returned to the control unit.

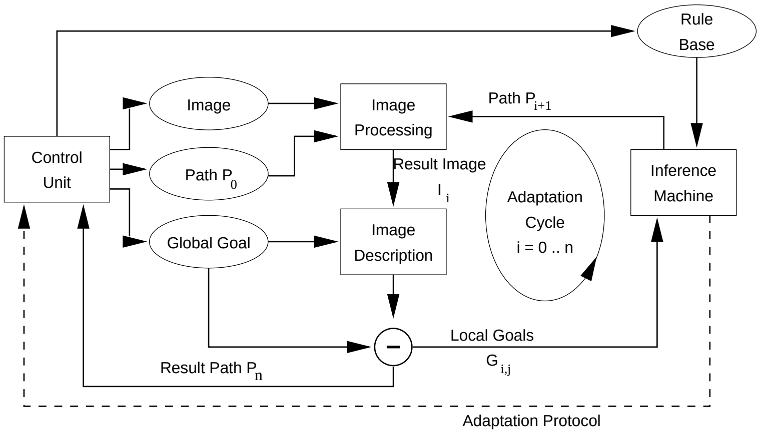


Figure 4: Adaptation of parameters

6 Results

The results which will be described in the following originate in the area of quality surveillance. In order to obtain a high inspection rate a short processing time was specified to be important. Further it has been assumed that the exact position of the contours is relevant for the following quality assessment. Therefore it was stated in the task that a high geometric accuracy concerning the detection of edges and vertices is necessary. Two kinds of test images were used. The first group depicts industrial objects lying on a light background. Since the images contain very little noise and nearly no disturbances as can be seen in Figure 5 the value for the property "noise" was set to "weak" and the value for the property "disturbances" to "not". According to some measurements near important contours the "ascendant width" was set to the value "weak" and the "edge height" was set to "strong". The second group of images have been scanned from x-ray images. They show the projection of a human knee in which a knee joint endoprosthesis was implanted. The properties of these images are similar but not equal to those of the first group. The "edge height" for example is only "average". The images of this group were taken to test the robustness of the adaptation concerning varying types of images. The overall goal concerning the result images was to separate the objects that can be seen in the input images from the background. The contour images should show as much of the object contours as possible and as few as possible contours that belong to the background. Therefore the global goal contains the specification that the "granulation" should be identical to "not" and that the "distinction" should be identical to "strong".

Using all operators listed in Table 1, fourty different paths can be generated. The configuration process restricts the list of possible operators so that much fewer paths remain: Due to the importance of getting a short processing time the morphological contour operator and the nonlinear Laplace operator are ruled out during the configuration. As the images contain little noise and nearly no disturbances no filtering is needed so that the Median operator, the Gauss operator and the Average operator are eliminated from the list. Finally the Pratt operator is excluded because of the "high accuracy" demanded concerning the positions of edges and vertices. The remaining operators can be used to form four different paths: (Sobel → Ranking), (Sobel → Binarize), (Roberts → Ranking), (Roberts → Binarize).

Figure 5 shows the results that have been achieved for the path (Sobel → Ranking) with the images lockwashers, grid and knee. This path contains two parameters that had to be adapted. By separately testing all parameter combinations for this path it could be verified that only about 1% of them are satisfying all conditions defined in the global goal. The adaptation performed took between 8 and 12 steps and 4 different rules were used. At the

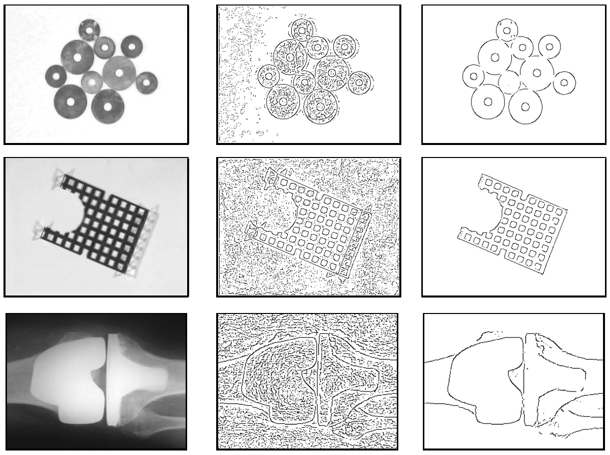


Figure 5: Results for the images lockwashers, grid and knee

left side the input images are shown, in the middle the result images of the paths P_0 can be seen. The right image depicts the result image achieved after the adaptation had finished. In all cases the global goal could be reached. The images show clearly that the quality of the contour images has been improved significantly with respect to the defined goals. Even a human expert is hardly able to achieve a better result using the implemented operators.

7 Acknowledgements

The project has been supported by a grant of the Deutsche Forschungsgemeinschaft.

References

[1] T.S. Newman, A.K. Jain (Ed.): "A Survey of Automated Visual Inspection", Computer Vision an Image Understanding, Vol. 61, No. 2, pp. 231–262, March 1995.

[2] I. Pitas: "Digital Image Processing Algorithms", Prentice Hall, 1993.

[3] C.-E. Liedtke, A. Blömer, T. Gahm, "Knowledge-Based Configuration of Image Segmentation Processes", International Journal of Imaging Systems and Technology, Vol.2, pp. 285–295, 1990.

[4] C.-E. Liedtke, A. Blömer: "Architecture of the Knowledge Based Configuration System for Image Analysis Conny", Proceedings of the 11th ICPR, International Conference on Pattern Recognition, The Hague, Vol I, pp. 375–378, Sept. 1992.

[5] U. Rost, P. Oechtering: "Knowledge-Based Genetic Learning", Proceedings of the Sixth Scandinavian Conference on Artificial Intelligence (SCAI 97), August 18.-20., Helsinki, Finland, pp. 107-118.