Robust Extraction of Objects Features in the System FRIEND II

S.M. Grigorescu and D. Ristić-Durrant

Institute of Automation, University of Bremen

Emails: grigorescu, ristic@iat.uni-bremen.de

Keywords: *machine vision framework, robust image segmentation, feature extraction*

Abstract

In this paper, a novel algorithm for robust extraction of object features for reliable feature based object recognition is presented. The algorithm aims at extracting reliable visual information about the objects to be manipulated which is necessary for robust autonomous functioning of the robotic system FRIEND II. The image processing algorithm is used for processing the image Region of Interest (ROI). The image ROI is defined based on information which is obtained from the control architecture of the robotic system. The structure of the Machine Vision Framework within the FRIEND II robotic system is presented along with the algorithm for the object feature extraction in the ROIs in the images captured by the FRIEND II stereo camera system. The presented algorithm is characterized by not using a-priory knowledge about the object of interest, such as knowledge about the shape or color of the object, in the image segmentation stage. The necessary robustness of image segmentation and consequently of object feature extraction is achieved by implementation of appropriate feedback structures, which makes the proposed algorithm novel with respect to algorithms so far published in the related literature. The proposed algorithm can be used in different working scenarios of the FRIEND II system, such as "serving a drink" and "library scenario". Also, algorithm can be used in different phases of a scenario, for example in "serving a drink" scenario in the phase of initial monitoring to answer the question "is there a bottle in the fridge" and in the phase of performing the task to provide visual information necessary for manipulating the bottle by the robot arm The reusing of extracted features of objects of interests in different phases of a working scenario or in different scenarios is provided by their saving in a dynamical database.

1. Introduction

In an image processing system aiming at object recognition the reliability of object recognition is strongly dependent on the result of extraction of object features while, on the other hand, the object feature extraction is strongly influenced by the correctness of the image segmentation [7]. Because of that, the largest part of image processing literature is dedicated to the improvement of image segmentation and consequently of feature extraction. For example, the improvement of feature extraction by using an adaptive segmentation method was treated in [1] where the visual inspection of machined metallic surfaces was done. However, in spite of the number of published image segmentation algorithms, there is still no general method which will lead to reliable feature extraction results in the presence of the numerous external influences that can arise in real-world applications. In the authors previous works [5][8] the novel idea of including feedback structures at different levels of an image processing system to adapt processing parameters for the purpose of improving system robustness against external influences, was investigated and the practical benefits of using control techniques in different image processing applications was demonstrated. This paper intends to give an additional contribution to the topic by presenting a novel robust closed-loop image segmentation method which will lead to reliable extraction of features of a general object from the image Region of Interest (ROI) independently of different external influences.

The target application of the presented method is robot vision for visual guided object grasping, a field of robotics in which a computer controls a manipulator's motion under visual guidance, much like people do in everyday life when reaching for objects. A key requirement in this field is the reliable recognition of objects in the robot's camera image, extraction of object features from the images and, based on the extracted features, subsequent correct object localization in a complex 3D-environment. In particular, this paper concerns the improvement of robustness of the vision system of the robotic system FRIEND II (*Functional Robot arm with frIENdly interface for Disabled people*), developed at the Institute of Automation (IAT) of the University of Bremen [9]. The FRIEND II system is a semi-autonomous robotic system designed to support disabled people with impairments of their upper limbs in activities of daily living and professional life. The facilities of the designed robotic system are materialized in three working scenarios: an *All Day Living* (ADL) scenario where the user is helped in common tasks such as preparing a meal or serving a drink, a *Library* scenario where the user is supposed to work at books handling desk and finally a *Workshop* scenario where the user with the help of the robotic system performs the task of testing different electronic components. In those working scenarios the FRIEND II robotic system has to deal with a variety of objects, including a fridge, bottles, glasses, books, meal trays, fridge door and the microwave oven door handles, etc. Some of these objects are uniformly coloured like the bottles, the glasses and the handle of the meal trays, while some of them are textured such as the books or some bottle types. Furthermore, some of the objects to be recognized may be in clustered environments, for example there may be several objects in the fridge or on the book shelf. Hence, the robot vision system must be robust enough to cope with the clustered environment (complex scenes) and with a variety of different objects as well as with different appearances of the same object in different lighting conditions that arise during the robot functioning (e.g. different artificial and natural lighting).

From the image processing point of view, the objects to be recognized in the FRIEND II system are classified in two categories: container objects (such as fridge, microwave oven and book shelf) and objects to be manipulated (such as bottles, glasses, meal trays, meal tray handlers, books, etc.). The focus in this paper is on robust extraction of features of objects which have appropriate size and shape for manipulation by the robot arm. The recognition of containers, achieved by a state-of-the-art model based recognition method [4], is taken for granted. Moreover, the result of recognition of containers is used as the possible starting point for definition of image ROI as will be explained in Section 2. The rest of the paper is organized as follows. The FRIEND II system architecture including the Machine Vision Framework is presented in Section 2. The closed-loop segmentation of image ROI is explained in Section 3. Section 4 presents the features treated at the feature extraction level of the object recognition system in the FRIEND II. The experimental results on the recognition of different objects from the FRIEND II environment are given in Section 5.

2. FRIEND II system architecture

2.1 Overall control architecture

The control of a complex robotic system such as FRIEND II can only be achieved using an appropriate control framework. The used architecture, entitled MASSiVE (*MultiLayer*

Architecture for SemiAutonomous Service Robots with Verified Task Execution), represents a distributed robotic control system which combines reactive behaviour with classical artificial intelligence based task planning capabilities [2]. The MASSiVE architecture is represented in Figure 1. The structure is divided in four specific modules. The *Human-Machine Interface* (HMI) operates at user interaction level. The user commands are acquired with the help of different input methods like Brain-Computer Interface (BCI), speech recognition or kin control, and translated further in machine language for interpretation [3]. The processing algorithms that converts a user request into robot actions resides in the *Reactive Layer*. Here, the data collected from different *Sensors* (such as stereo cameras, tactile tray, etc.) are processed in order to "understand the environment" and they are further converted into actions by the available *Actuators* (such as 7 degrees-of-freedom manipulator). The sequence of operations needed to perform a specific task is computed by the *Sequencer* module. The *Sequencer* plays the role of a *Discrete Event Controller* (DEC) that plans sequences of operations by means of predefined task knowledge [2]. Through the functioning of the system, the computed data is shared between the modules with the help of the *World Model*. The *World Model* defines the information produced and consumed by the operations in the *Reactive Layer*. The software interconnection between the processing layers is implemented using CORBA (*Common Object Request Broker Architecture*).

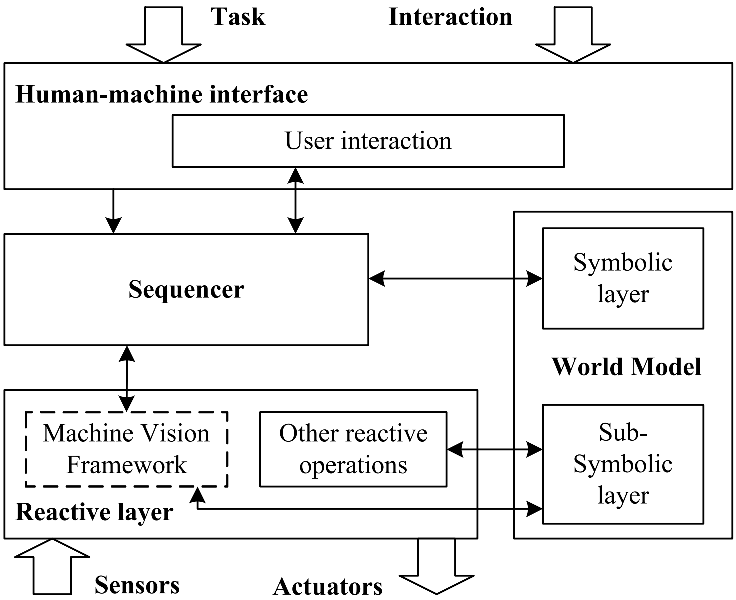


Fig. 1. FRIEND II overall control architecture.

2.2 Machine Vision Framework

The Machine Vision Framework of the FRIEND II system is composed of a Bumblebee® Stereo Camera system and a computer for running the image processing algorithms, which the stereo cameras are connected to. The stereo camera system is mounted on a frame-rack behind the head of the user sitting in the wheelchair. The cameras view the scene in front of the robotic system including the manipulator and the tray which is mounted on the wheelchair.

The images acquired by the cameras are sent to the image processing system in RGB (*Red*, *Green*, *Blue*) form. At the image preprocessing level, the conversion between RGB and HSI (*Hue*, *Saturation*, *Intensity*) color spaces or conversion of RGB to intensity images are done as demanded for the next steps in processing algorithms. In order to perform complex vision tasks, different machine vision algorithms such as container recognition and segmentation of objects to be manipulated are interconnected. Also, the machine vision algorithms are connected to external modules (such as HMI) as shown in Figure 2. The flow of processing operations shown in Figure 2 corresponds to the case of detecting a bottle in the fridge. The overall goal of the Machine Vision Framework is the reconstructed 3D environment which is the information used further by the manipulative skills for virtual modelling of the scene viewed by the robotic system. The virtual model of the robot environment is used for the path planning and object grasping by the manipulator.

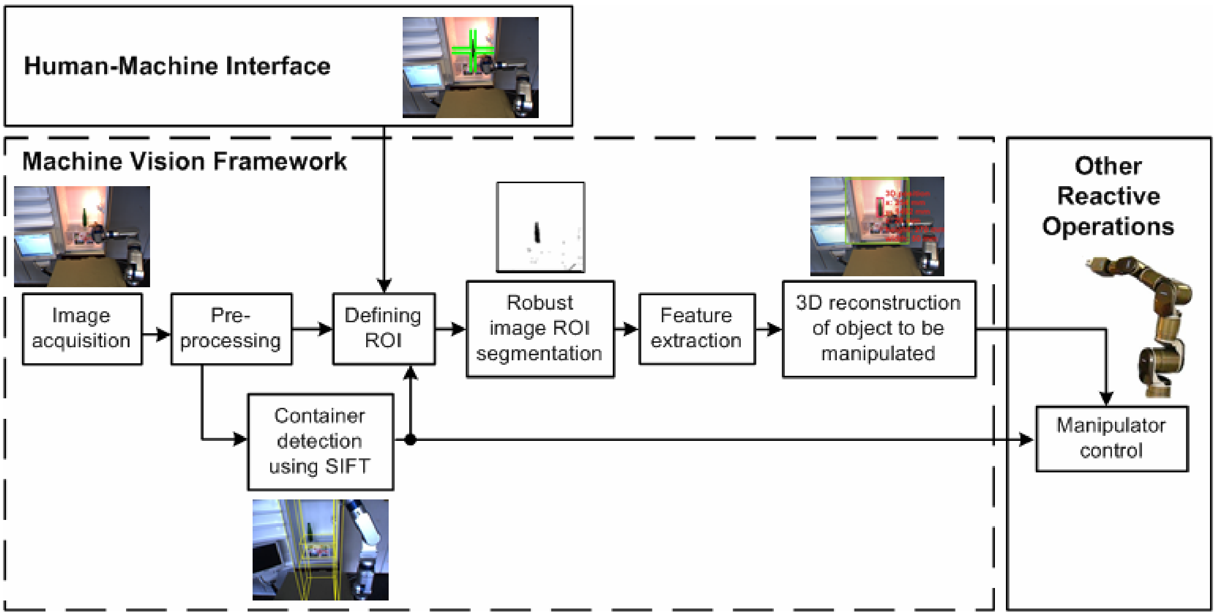


Fig. 2. Flow of image processing operations in the system FRIEND II.

2.2.1 Defining image ROI

The purpose of finding the image ROI is to reduce the complexity of the imaged scene, that is to reduce the object search area from the whole imaged scene to a smaller image area containing the object(s) to be recognized. In the presented vision system, image ROI can be defined for two cases which differ with respect to the level of a-priori knowledge about the location of the object of interest within the image. In the first case only partial knowledge about the object environment is available. For example, in the FRIEND II system the available information is of the form: “the object is in the fridge”, “the object is in the microwave oven” or “the object is on the shelf”. In the second case, precise information on the object position within the image is available. Namely, the assumption is that the image coordinates of at least one object pixel are known. In the semi-autonomous rehabilitation robotic system FRIEND II such information can be obtained through the human-machine interface (HMI) where the user locates the object of interest by using a particular action, such as clicking on the displayed image using a special input device.

Bearing in mind that the container objects in the FRIEND II environment are constant in scenarios, a state-of-the-art model based recognition method named SIFT (*Scale Invariant Feature Transform*) [4] is used for their recognition. This method uses a model image to off-line train a classifier. During the on-line system operation the SIFT algorithm searches for the model image in the scene through a matching based algorithm. Once the model image has been detected its pose (position and orientation in 3D space) can be reconstructed. Knowing the position of the model image placed on a container, e.g. on the fridge, the container pose can be further reconstructed. Starting from the reconstructed 3D pose of the container, the container region in the image is obtained using 3D to 2D mapping. The resulted image region enclosing the container in which the object of interest is located is the image ROI. Hence, in this case, the defined ROI will enclose all the objects present in the container, not just the object of interest. In contrast to this, the achieved ROI when the image coordinates of an object pixel are known through the HMI will surround only the object of interest as shown in Figure 3. As shown, in both cases the result of a ROI definition is a rectangular region defined in 2D image coordinates with the origin pixel (0, 0) located at the top-left corner of the image. The boundary image ROI points p_i, i=1,…,4, are obtained either through the mapping of 3D container points to 2D or through the automatic adjustment of the ROI size when the initial point of the ROI is obtained through the HMI. The automatic adaptation of the size of the ROI when it is defined through the HMI (starting from one object pixel defined by the user as explained above) is out of the scope of this paper.

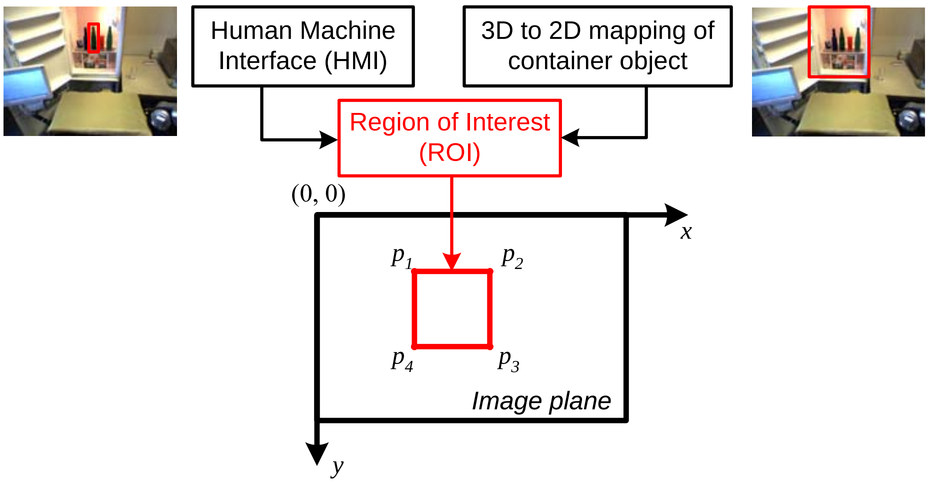


Fig. 3. The ROI definition process.

In contrast to container objects the variety of objects to be manipulated in FRIEND II scenarios is large. Because of that, it is not appropriate to use model based recognition methods since they would demand storing of a large number of models and would fail in the presence of entirely unknown objects. The idea is to develop an object recognition method which does not need a-priory knowledge about the objects to be recognized such as knowledge on color of the object. Besides, the developed method must be robust enough to cope with different external influences such as changes in illumination. Such a method which is based on inclusion of closed-loop control at image segmentation level is described in the following section.

3. Closed-loop segmentation of image ROI

The reliable object features extraction from the object segmented image can only be done in the case of segmented image of good quality. A binary segmented image is said to be of good quality if contains all pixels of the object of interest forming a “full" (unbroken) and well shaped segmented object region. The image segmentation algorithm presented in this paper employs the idea of inclusion of feedback structures to control the quality of binary segmented image ROI. The idea behind this is to adjust the parameters of image segmentation in a closed-loop manner so that current segmented image ROI is driven to the one of reference quality. The motivation comes from the fact that error-based closed-loop systems have a natural ability to provide disturbance rejection and robustness against system uncertainty, which is a fundamental concept in control theory.

3.1 Choice of actuator and controlled variables

The use of closed-loop control in image processing differs significantly from its use in conventional industrial control, especially concerning the choice of the actuator and the controlled variables. Generally, the actuator variables are those that directly influence the result of image processing. In the system presented here, the image segmentation is done by thresholding of the so-called *Hue* image, which contains the pure color information of the original RGB image of a scene from a FRIEND II working scenario. In thresholding each pixel from the *Hue* image to be segmented is set to the foreground black color in the output segmented image if its pixel value belongs to a particular interval of the color values [8]. To further explain the thresholding operation, the *Hue* image is defined as a two-dimensional function $f(x,y)$, and the object color interval as $C_l = [T_{min}, T_{max}]$ where T_{min} and T_{max} are the minimum and maximum color values across the object's pixels. Then, the thresholding operation is defined as:

$$t(x,y) = \begin{cases} 1, & \text{if } f(x,y) \in C_l, \\ 0, & \text{if } f(x,y) \notin C_l. \end{cases} \tag{1}$$

where $t(x,y)$ is the segmented binary image, 1 and 0 represent black and white color respectively, and x and y are the *Hue* image pixel coordinates. For the sake of clarity an object color interval C_l in the following is referred to as an *object thresholding interval.*

Figure 4 shows the results of segmentation of a green bottle which is located in the fridge in the FRIEND II environment. As evident, only the correct choice of object thresholding interval $[T_{min},T_{max}] = [35, 55]$ yields the segmented ROI of good quality, containing whole, well segmented object. In contrast, an incorrect choice of the thresholding interval causes the segmentation failure. As shown in Figure 4, intervals $[T_{min},T_{max}] = [0, 20]$ and $[T_{min},T_{max}] = [113, 133]$, which lay outside the interval of the object of interest pixel values, yield respectively a ROI with segmented background pixels (noise) and a ROI without any segmented pixel. Therefore, in order to achieve good object segmentation it is necessary to adjust the object thresholding interval. For this reason, the object thresholding interval, that is the threshold increment i, $[T_{min} + i, T_{max} + i]$, is considered as the actuator variable $u = i$ in the presented system.

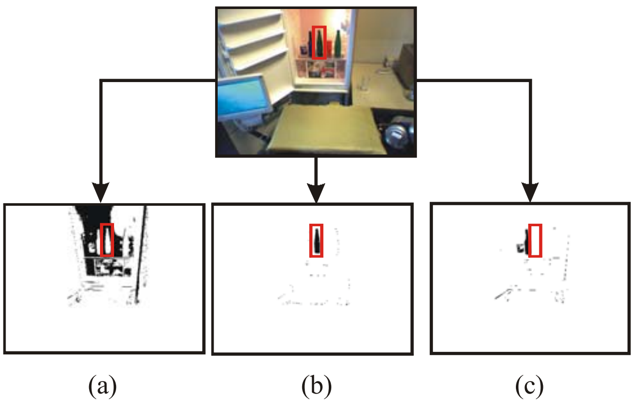


Fig. 4. Binary segmentation of a green bottle in the FRIEND II environment corresponding to different thresholding intervals: (a) $[T_{min},T_{max}] = [0, 20]$, (b) $[T_{min},T_{max}] = [35, 55]$, (c) $[T_{min},T_{max}] = [113, 133]$.

In order to automatically adjust the actuator variable so that the current quality of segmented image ROI is driven to the desired, reference, value a controlled variable has to be defined. The chosen controlled variable has to be appropriate from the control, as well as from the image processing, point of view. From the image processing point of view, a feedback variable must be an appropriate measure of image ROI quality. Two basic requirements for control are that it should be possible to calculate the chosen quality measure easily from the image and the closed-loop should satisfy input-output controllability conditions. Input-output controllability primarily means that for the selected output (controlled variable) an input (actuator variable) which has a significant effect on it must exist in the image processing chain.

Bearing in mind qualitative definition of segmented image ROI of good quality given above, the following quantitative measure of segmented image ROI quality has been proposed:

$$I = -\log_2 p_8\,, \quad I(0) = 0\,, \tag{2}$$

where p_8 is the relative frequency, that is, the estimate of the probability of a segmented pixel surrounded with 8 segmented pixels in its 8-pixel neighbourhood:

$$p_8 = \frac{\text{number of segmented pixels surrounded with 8 segmented pixels}}{\text{total number of segmented pixels in the image ROI}}. \tag{3}$$

In the calculation of p_8 the segmented pixels whose 8-pixel neighbourhood is completely within the ROI are only taken into account. Bearing in mind that a good segmented image ROI contains "full" (without holes) segmented object region, it is evident from (3) that the small probability p_8 corresponds to the large disorder in a binary segmented image ROI. Due to that, the large *uncertainty I*, defined by (2), is assigned to the segmented image ROI. Therefore, the goal is to achieve the segmented image ROI having as small as possible uncertainty measure *I* in order to get reliable segmentation result.

In order to investigate the input-output controllability of the system when considering the threshold increment as the input (actuator) variable, and the proposed uncertainty measure *I* as the output (controlled) variable, the thresholding of the ROI, containing only the green bottle, in the image shown in Figure 4 was done. The thresholding interval was set to an initial state $[T_{min}, T_{max}] = [0,\ 20]$. To this interval the increment $u = i$ was added as $[T_{min} + i, T_{max} + i]$. For each segmented image corresponding to the increment $i \in [0,\ 159]$, the uncertainty measure *I* was calculated. The resulting input-output characteristic is presented in Figure 5(b). As it can be seen, the uncertainty *I* is sensitive to the chosen actuator variable across its effective operating range. Also, it is clear that each input value is mapped to at most one output value and that it is possible to achieve the minimum of *I*, which corresponds to the segmented object image of reference good quality, by changing the thresholding boundaries. The satisfaction of these prerequisites for successful control action to be performed demonstrates the pair "threshold increment – uncertainty measure *I*" as a good "actuator variable – controlled variable" pair.

The same experiment, as described above, was done also for the case of the ROI containing more objects. The resulting input-output characteristic is shown in Figure 5(a). As it can be see, the characteristic has more local minimas. Each minimum corresponds to the good segmentation of a particular object. For example, the first minimum achieved for the optimal threshold interval [35, 55] corresponds to the green bottle, as in the case of the previously described experiment and characteristic shown in Figure 5(a), while the second minimum at [125, 145] corresponds to a blue object.

Based on the above discussion it can be said that the original problem, that of finding the optimal object threshold interval that provides segmented object image of good quality, appropriate for subsequent object features extraction, can be interpreted and converted to the problem of finding the minimum uncertainty *I* of the object region in the binary segmented image.

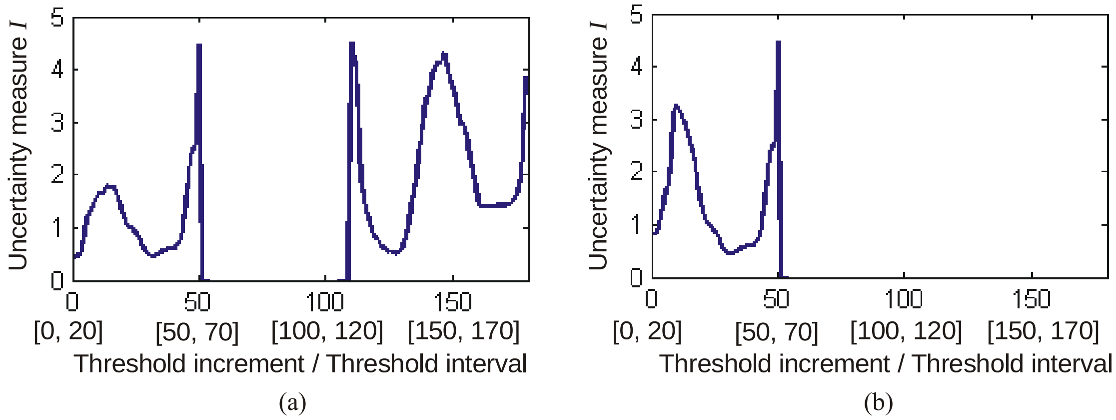


Fig. 5. The uncertainty measure *I* of segmented pixels vs. threshold increment / threshold interval. (a) ROI containing more objects. (b) ROI containing only one object.

3.2 Closed-loop control design

As discussed in the previous section, the reference value of the chosen controlled variable is not explicitly known in the presented system. However, the selection of an image ROI quality measure whose minimal value corresponds to the image ROI of good quality has been suggested for the controlled variable. Hence, the optimal value of the chosen controlled variable is achieved by an optimization using an appropriate extremum searching algorithm through a control structure, as shown in Figure 6. Here, in principle, the feedback information on the image ROI quality is used to choose the optimal value u_{opt} of the actuator variable u, that is, to drive the current image ROI to one with the reference optimal quality.

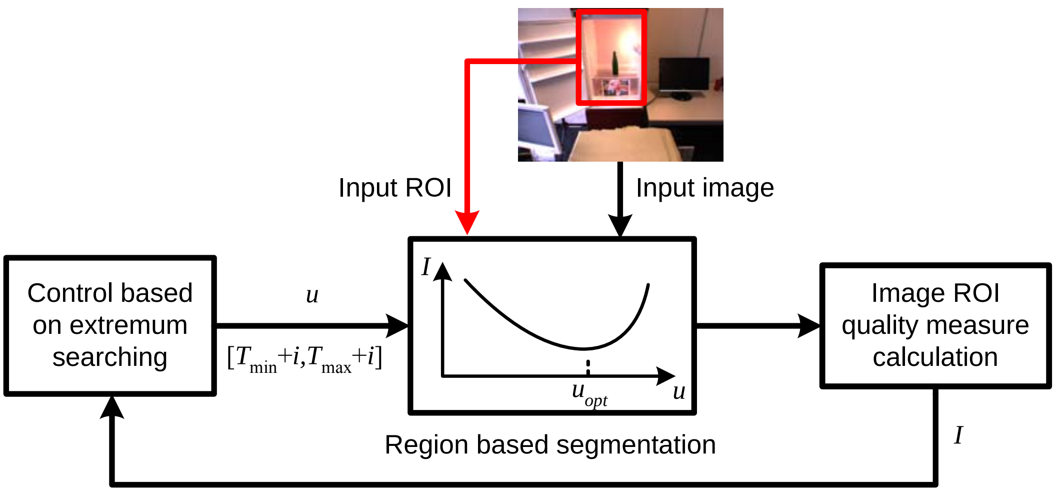


Fig. 6. Block diagram of the proposed closed-loop system.

The goal of the presented closed-loop system is to extract the optimal threshold intervals for the objects present in the image ROI. Then, input image of a scene from the FRIEND II environment is segmented subsequently using the extracted threshold intervals. Therefore the segmentation result in one binary image in the case of an image ROI containing only one object, or it results in as much binary images as much objects are present in the defined image ROI.

4. Feature extraction

The input to the feature extraction module in the Machine Vision Framework of the FRIEND II system are binary segmented images resulted from the closed-loop segmentation procedure described above. First, the feature extraction module searches for the regions of connected segmented black pixels known as blobs. Due to the properties of presented closed-loop segmentation, the dominant blob in the input binary image corresponds to the objects of interest. However, due to the color distribution in input color image some background pixels can also be segmented as foreground black pixels though they represent noise. But, again due to the properties of the presented closed-loop segmentation method, noise blobs are of such size that can be filtered.

In the presented system, for each dominant blob (segmented region) a number of features are calculated such as:

- *Area* – the number of pixels in the segmented region,
- *Bounding box* – the smallest rectangle containing the segmented region,
- *Width* – the width of the bounding box along each dimension,
- *Centroid* – the center of mass of the segmented region,
- *Central* and *invariant moments*, such as Hu moments [6][8], etc.

Along with the above listed features the optimal value u_{opt} of the actuator variable, threshold interval, resulted from the presented closed-loop structure is extracted as the object feature since it uniquely defines the object color.

The extracted object features are used at the classification step to classify the object as belonging to a particular class. In the FRIEND II system object classification is followed by the object feature points extractions. Feature points for a particular object class (bottle, glass, meal tray handler, etc), which are necessary for 3D object reconstruction, are specified in the FRIEND II database, that is in the *World Model* of the FRIEND II control architecture.

5. Experimental results

In order to evaluate the performance of the proposed closed-loop segmentation technique, the thresholding of ROIs in images containing different objects from FRIEND II environment, blue glass, green bottle and red meal tray handlers, was done. The segmentation results are shown in Figure 7. As evident, closed-loop segmentation results in segmented image ROIs of good quality, since they contained “full” well shaped object regions. The effectiveness of the proposed method is emphasised by comparing its performance with the performance a traditional open-loop local thresholding [5]. This traditional method is based on finding the optimum threshold for particular image region by investigating the distribution of pixel values in the region. The optimal local threshold is defined as the *mean* value of the pixel value distribution. It can be seen that the open-loop segmentation of the glass and the bottle results in segmented regions of poor quality since they contain small number of scattered object pixels. In the case of red meal tray handler open-loop segmentation completely failed since the background pixels are segmented as foreground object pixels. These poor open-loop segmentation results will lead to incorrect object features extraction. In contrast, good closed-loop segmentation result yields reliable objects features extracted from the segmented regions. This can be seen from Table 1 where the results on features extraction are given along with the ground truth. The listed features in Table 1 are the color and the height of the object, in case of the glass and the bottle, that is the object length in the case of the meal tray handler. Extracted object height, that is object length, is actually the height of the *Bounding Box* described in Section 4 given in pixels (px) and estimated in cm.

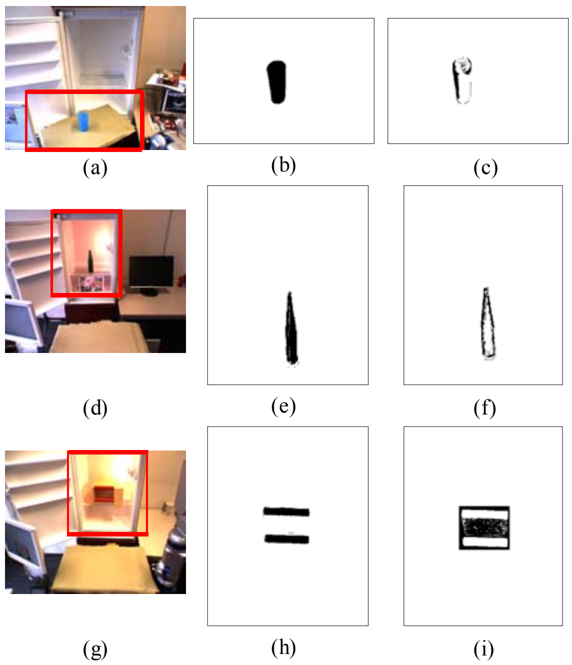


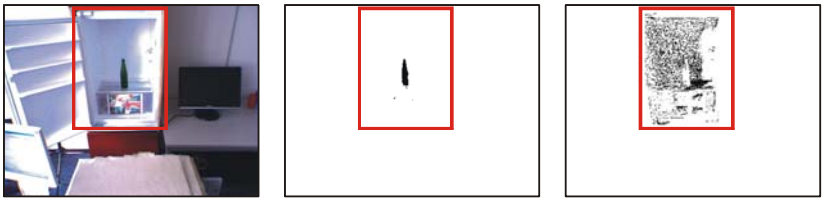
Fig. 7. Test images (a,d,g) of scenes from FRIEND II environment along with the corresponding closed-loop (b,e,h) and open-loop (c,f,i) results of segmentation of ROIs.

Table 1. Extracted and ground truth features of the three objects of interest.

Feature		Object 1	Object 2	Object 3
Extracted by proposed method	**Segmentation interval - Color**	[97,117]Blue	[35, 55] Green	[0,8]Red
Extracted by proposed method	**Height/length**	91px 14cm	118px 27.7cm	111px 18.5cm
Ground truth	**Color**	Blue (glass)	Green (bottle)	Red (handler)
Ground truth	**Height/length**	12.6m	28cm	18cm

The feature extraction results given in Table 1 demonstrate the robustness of the proposed closed-loop technique with respect to the object variation. This confirms benefit of the use of feedback information about the image ROI quality to adjust the image ROI processing parameters.

The proposed method has also necessary robustness with respect to illumination which can be seen from Figure 8. Here the results of the proposed closed-loop and considered open-loop segmentation of the image of the same scene as in Figure 7(d), only imaged in different illumination conditions, are shown. As evident closed-loop technique resulted in the segmented image ROI of good quality independently of the illumination condition.



(a) (b) (c)

Fig. 8. Test image (a) of a scene from FRIEND II environment along with the corresponding closed-loop (b) and open-loop (c) results of segmentation of ROI.

6. Conclusions and outlook

This paper considers the benefit of the use of control techniques in image processing, representing a new control application field. The usefulness of the closed-loop image segmentation is demonstrated through the results on object recognition in the robotic system FRIEND II. Presented results on comparison of the performance of proposed closed-loop method with the traditional open-loop object recognition demonstrate the need for the feedback adaptation of processing parameters to different external influences. Also, the importance of the choice of an appropriate "actuator variable-controlled variable" pair when implementing closed-loop image processing for improvement of object recognition is demonstrated.

7. References

[1] Pernkopf, F.; O'Leary, P.: Visual inspection of machined metallic high-precision surfaces, *EURASIP Journal on Applied Signal Processing*, nr. 7, pp. 667–678, 2002.

[2] Martens, C.; Prenzel, O.; Feuser, J.; Gräser, A.: MASSiVE: MultiLayer Architecture for SemiAutonomous Service Robots with Verified Task Execution," in *Proc. 10th Int. Conf. Optimization of Electrical and Electronic Equipments*, pp. 107–112, Brasov, 2006.

[3] Prenzel, O.; Martens, C.; Cyriacks, M.; Wang C.; Gräser A.: System controlled user interaction within the service robotic control arhitecture MASSiVE," *Robotica, Special Issue*, 2007.

[4] Lowe, D.G.: Object Recognition from Local Scale-Invariant Features, in *Proc. of the Int. Conf. on Computer Vision, Corfu*, September 1999.

[5] Ristić, D., Feedback structures in image processing. *PhD. thesis*, 2007.

[6] Hu, M.K.: Visual pattern recognition by moment invariants, *IRE Trans. on Information Theory*, pp. 179–187, 1962.

[7] Gonzalez, R. C., Woods, R. E.: *Digital Image Processing*, Prentice-Hall, New Jersey, 2002.

[8] Vuppala, S. K., Grigorescu, S. M., Ristić, D., Gräser, A.: Robust Color Object Recognition for a Service Robotic Task in the System FRIEND II, *10th International Conference on Rehabilitation Robotics - ICORR'07,* 2007.

[9] Volosyak , I., Ivlev, O., Gräser, A.: Rehabilitation robot FRIEND II - the general concept and current implementation, *9th International Conference on Rehabilitation Robotics - ICORR'05*, pp. 540--544, Chicago, 2005.