Proceedings of the Federated Conference on
Computer Science and Information Systems pp. 799–808

DOI: 10.15439/2015F184
ACSIS, Vol. 5

Shape and colour recognition of dishes for the purpose of customer service process automation in a self-service canteen

Tomasz Kryjak, *Member IEEE*
AGH University of Science and Technology
al. Mickiewicza 30, 30-059 Krakow, Poland
Email: tomasz.kryjak@agh.edu.pl

Damian Król
Email: damians.krol@gmail.com

***Abstract*—In the article a vision system for shape and colour recognition of dishes (plates, bowls, mugs), which can be used to automate the process of customer service in a self-service canteen is described. In consists of three basic components: object segmentation using so-called background model subtraction, shape recognition using geometric invariant moments and SVM classifier, as well as colour recognition using a Gaussian model. In addition, recognition in case of close or abut objects using a distance transform like approach is presented. The solution was evaluated on a dedicated test stand with controlled LED lightning. A 98% accuracy was obtained on over 100 test images, which indicates that the solution could be used in business practise.**

I. Introduction

IN TODAY'S world a rapid automation of customer services in different areas can be observed. Among numerous examples worth to mention are: the growing number of vending and ticket machines, automatic cash registers in supermarkets or recently emerging touch-screen kiosks for ordering and paying in fast food restaurants.

The customer service process in a self-service canteen can also be subjected to automation. This topic is particularly relevant in the context of the currently ongoing economic and social changes. The model, which assumes eating lunch or dinner after work at home changes to dining in canteens, lunch-bars or restaurants. Therefore, an increasing number of various types of self-service restaurants can be observed. There, the customer takes a tray, collects the meals and proceeds to the cash desk, where the content of the tray is priced by the cashier and the payment is made.

The automation of the described process requires the use of two modules: the pricing of products on the tray and the payment collection. The second issue is already well know and applied on a large scale: payment terminals (with PIN based authentication and so-called contactless cards) and CDMs (cash deposit machines). However, the automatic pricing of meals or dishes is still a challenge.

The topic of vision based food classification is addressed in several research papers. It should be noted that the issue is very challenging due to the great variety of possible meals, which are often quite similar. Another big problem is the segmentation of a particular food type on a plate.

In the work [1] the GrabCut segmentation is used to extract individual food types, as well as SURF features and SVM classifier. The accuracy of the system is more than 81%. In paper [2] the concept of using local and global features is described. Additionally, the results from several individual classifiers are combined to improve the accuracy, which for the system is about 80%. Similar results were obtained in the work [3]. The authors of the work [4] used a combination of deformable part-based and a texture model. For particular food recognition a multi-view multi-kernel SVM was utilized. The accuracy of this system reaches 90%.

This short analysis reveals that the topic of food recognition is up to date and intensively researched. However, the high complexity limits the accuracy of the systems to about 90%. This is insufficient for an automated pricing system in a self-service canteen. It is also worth noting, that for similar systems several patents can be found – for example from the SRI company [5]. In addition, an interesting system to automate the pricing of bakery products developed by Brain Corporation is presented in a video available at [6].

In this paper a pricing system based on video analysis of the shape and colour of dishes is presented. To the best knowledge of the authors, this is the only described in the literature vision system operating on this principle. The input to the system is a photo (single video frame) of a tray with one or more dishes (plates, bowls, mugs). Then, the individual objects are extracted and their shape and colour is determined. To use this information in the pricing task it has to be assumed that on a given day, on a particular dish (defined by shape and colour) and single meal is served. For example: on a *large round, green* plate pork chop with french fries and on a *large round, red* plate fish with potatoes is served. It is worth noting that such a system quite strictly defines the operation rules of the canteen. The meals have to be placed by the staff on appropriate dishes. This can be done on a regular basis (applying to the customer's request) or prepared in advance (the problem of keeping the meal warm). In addition, it should

The work presented in this paper was supported by AGH University of Science and Technology project number 15.11.120.476.

be assured that the boarder of the dish is free form parts of the meal.

The remainder of this paper is organized as follows. Section II contains a general overview of the proposed vision system. Detailed information about particular modules i.e. object segmentation, shape recognition, colour recognition and close or abut object recognition are given in Sections IV, V and VI. Evaluation of the system is presented in Section VII. The paper ends with a summary and indication of future research directions.

II. Overview of the proposed vision system

This section provides basic information about the discussed vision system. Is has been implemented in C++ programming language using the OpenCV image processing library [7] and the Qt GUI library [8].

The designed and implemented algorithms were evaluated on a specially constructed test stand, which consisted of:

- digital camera – a typical USB camera with resolution 640×480 pixel was used (Logitech Webcam Pro 9000),
- illuminator – in order to provide good lighting conditions for image acquisition, a LED based illuminator was developed. It was a source of strong, white and diffused light. This reduced the problems of shadows and disturbances caused by external lightning,
- housing – to partially isolate the workspace from external lighting. It was also the mounting point of the illuminator and the camera,
- worktop – it was used for proper tray positioning.

In the application image processing and recognition was performed in two modes:

- continuous (for every frame) – tray and hand presence detection, background model acquisition and update,
- on-demand (at user request) – object segmentation and recognition followed by item pricing.

A. Continuous mode

In order to provide proper classification, it is necessary to correctly place the tray in the field of view of the camera (i.e. in the workspace). It was assumed that the positioning in the axis perpendicular to the typical tray movement will be forced by two guide-rails (movement to and from the user), while in the other axis will be controlled by automatic detection of square markers. The markers are placed in the workspace, in a distance corresponding to the tray width. Using information about their visibility, the presence and position of the tray can be determined.

The second module working in the continuous mode is the detection of the so-called empty workspace i.e. a situation when in the camera's field of view there are no objects: plates, tray or even dirt. It such case, the current image is saved and than used in the segmentation phase as a background model (details in Section III).

The third module is responsible for detecting the presence of objects that have a common part with the boarder of the workspace. This prevents from starting the image analysis, when the user still holds the tray or has his hand in the camera's field of view. A detailed discussion of the three modules is presented in the work [9].

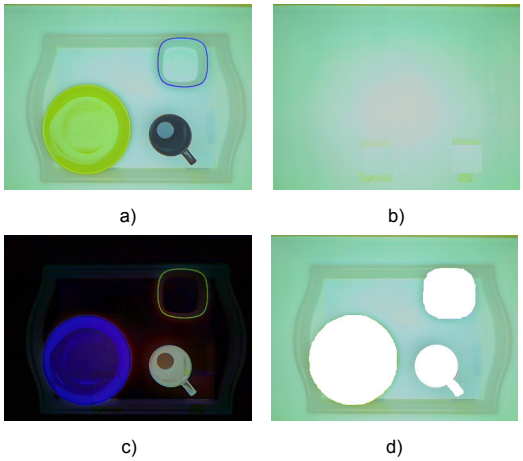

Fig. 1. Object segmentation using background subtraction. a) current workspace image, b) background model (image of an empty workspace), c) differential image, d) thresholding result overlaid on the input image (with same additional post-processing)

B. On demand mode

When the tray is correctly positioned and the user has removed his hand from the workspace, the analysis can be started by pressing a button. It consists of the following steps:

- object segmentation,
- object shape recognition (also for close or abut objects),
- object colour recognition.

These steps will be discussed in detail later in the paper. The final result is the information about detected objects – their shapes and colours. On this basis, it is possible to price the tray.

III. Object segmentation

The segmentation of objects (i.e. dishes/plates) is one of the most important elements of the described vision system. On its accuracy, to a large extent, depend the further processing steps: shape and colour recognition.

Taking into account the specificity of the task, the purpose of segmentation is to isolate:

- the correct shape of the objects – for classification,
- continuous edges of the objects, thick enough to retrieve colour information – the interior of the dishes can not be considered as a reliable source of information about colour, as it usually contains meals.

In addition, the method should reduce the possibility of connecting separate objects due to occurring shadows and have a tolerance to variable lighting conditions.

In the proposed solution object segmentation is based on the so-called background subtraction approach. From the current image with tray and plates, a background image is subtracted (view of an empty workspace). Thresholding the differential

image allows to extract individual objects. The concept is illustrated in Figure 1. It is worth noting that in Figure 1b there are no markers visible on the worktop. This is due to a marker removal procedure, that involves replacing the marker's ROI with a workspace image from another location. This allows to obtain correct segmentation of the tray content, because markers are not detected.

The method has two main advantages. Its concept and the obtained results are easy to interpret i.e. all detected objects are regarded as dishes. Furthermore, it is very efficient. It should also be noted, that the empty workspace detection and background model update procedures allow to compensate lighting changes (e.g. naturally occurring during the day), which could affect the segmentation accuracy.

Limitations include: the need that objects or at least their boarder, have a colour different from the workspace and the requirement that the tray has the same colour as the workspace background – the tray should be transparent for the segmentation procedure.

In the current version of the algorithm, the segmentation is carried out in the RGB colourspace. First, the absolute value of the difference between the current image and the background model is computed – separately for each component. Then the maximal difference is selected and compared with a thershold. The obtained object mask is post-processed using morphological closing with a 5×5 square structuring element. Finally, a hole filling procedure is applied to eliminate the influence of the plate content on the segmentation result. A detailed description of the segmentation procedure is presented in the work [9].

IV. Object shape recognition

The object shape recognition is a two-step process. First, a feature vector, which describes each shape, is generated. Then, this vector is assigned to a pre-defined class (i.e. classified). The input to this procedure is a binary object mask (containing a single object) and the output the class to which this object belongs.

A. Feature vector

To describe the shape of an object (i.e. to generate a feature vector) a common approach involves the use of shape descriptors. This task is not easy, because on one hand a good differentiation of shapes is required (e.g. squares, circles, ellipses, etc.) is required. On the other hand the description must be insensitive to scaling, translation, rotation, affine transformations and some disruptions (e.g. "ragged" shape edge). In the literature a lot of different shape descriptors are described. They can be roughly divided into contour based (only the edge is analysed) and area based (the whole object is analysed). In the paper [10] four of them were evaluated: Fourier descriptors, curvature scale space descriptor, angular radial transform and image moments. The experiment showed that the geometric moments are a good solution for shape description. Furthermore, their are implemented in the popular image processing library OpenCV [7].

Fig. 2. The used shapes: circle, mug, rsquare, elipse

In the described system geometric invariant moments, also often referred to as Hu moments, are used. They have been proposed in the work [11] and are the basis of many shape recognition approaches. For example, in [12] they are utilized for human action recognition.

B. Classifier

As a classifier the Support Vector Machine (SVM) algorithm was used. It was originality proposed by Vladimir Vapnik [13] and is one of the most popular machine learning algorithms. This is due to: intuitiveness of the method, good accuracy, high computational efficiency and quite simple and quick learning procedure (possible to automate because of small number of parameters).

In the basic version, the SVM is a binary linear classifier. However, a modification was proposed (so-called kernel trick), that allows the classification of non-linear problems. It involves the transformation of the feature vectors to a new space with higher dimensionality. Often as the kernel the Gaussian radial basis function (RBF) is used (the default solution in OpenCV library).

To enable the classification of more than one object type the multi-class problem is transformed into multiple binary classifications. For example, for three shapes S_1, S_2, S_3 three classifiers are required: C_1 to distinguish S_1 from S_2 and S_3, C_2 to distinguish S_2 from S_1 and S_3 and C_3 to distinguish S_3 from S_1 and S_2.

C. Training dataset preparation

The use of a machine learning approach requires the preparation of three datasets: training, validation and test. The first is used to train the classifier i.e. to obtain the required parameters. The second to test various options (e.q. SVM parameters or number of used features). The last to asses the final solution. Typically, the input dataset is divided at a ratio of 60%, 20%, 20%.

In the current version of the system the following shapes are used:

- circle (big plate, small plate, bowl),
- mug (circle with a handle),
- rounded square (rsquare) (plate, small bowl),
- elipse (platter).

The used shapes are presented in Figure 2. Using these templates three datasets: training, validate and test were generated. For this purpose the templates were: scaled, rotated, subjected to affine transform, disturbed.

A template and exemplary samples (rotated and disturbed) are presented in Figure 3.

Fig. 3. Shape template (*rsquare*) and two samples: rotated and disturbed

Finally, the training dataset consisted of 2696 samples (673 for each shape), the validation dataset of 748 samples (187 for each shape) and test dataset of 876 samples (219 for each shape). For each sample the first three Hu moments were computed. They formed a feature vector used in training and evaluation of the classifier. The features were subjected to normalization given by: $s_n = (s - f_{min})/(f_{max} - f_{min})$, where f_{max} and f_{min} are respectively the largest and smallest value of the feature (particular Hu moment).

D. Classifier training and evaluation

Training and evaluation of the classifier was performed on the prepared training, validation and test datasets. The SVM with RBF kernel available in OpenCV was used. The classifier was trained with the `train_auto` function, which performs a multiple cross-validation procedure to select the best SVM parameters.

During the experiments, the impact of number of used Hu moments on the classification performance was evaluated. The accuracy $ACC = TD/(TD + FD) * 100\%$ measure was utilized, where TD – number of correct classifications, FD – number of incorrect classifications.

The obtained results are summarized in Table I. The analysis indicates that using only the first two Hu moments should allow to obtain very good classification accuracy. For this case an experiment on the test dataset was carried out and the following results were achieved: *circle* – 100%, *elipse* – 100%, *mug* – 100%, *rsquare* – 100%.

E. Results and comments

The proposed shape recognition method achieved almost 100% accuracy. It turned out, that Hu moments are well suited for distinguishing simple geometric shapes. For the considered application the use of the first two invariants resulted in satisfactory performance. The SVM classifier is very easy to use (lots of libraries, available in OpenCV and Matlab) and fast during classification. The solution was designed to allow a simple extension of the feature vector – for example adding other Hu moments or shape descriptors. It is also worth to notice, that the automatic generation of training samples significantly facilitated the evaluation of the approach.

V. Object colour recognition

Colour, next to shape, is the second feature which is used to identify a particular dish in the system. In this section the method of obtaining colour samples, evaluation of various colour models, as well as the used solution is presented. In the current version of the system colours: blue, orange, green, brown[1] are recognized.

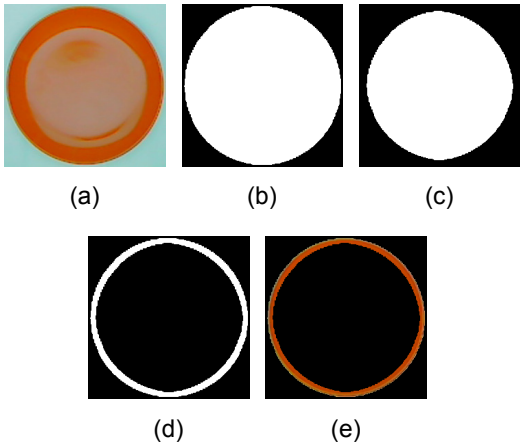

Fig. 4. Edge extraction demo. (a) current image, (b) object mask, (c) mask after erosion, (d) subtraction of (b) and (c), (e) the extracted edge

A. Obtaining colour samples

The process of obtaining samples used to build the colour model is an element of the system calibration procedure. During its course, the dishes with particulars colour are placed in the workspace. Then object segmentation and edge extraction are performed. Reliable colour information can be only obtained from a narrow edge of the dish, mainly due to presence of meals in its central part.

The edge is extracted using a morphology-based approach. First, the input mask (obtained in the segmentation stage) is subjected several times to erosion with a square structural element of size 3×3 . Then the input mask and erosion result are subtracted. Finally, the edge with colour samples is obtained. Examples of this procedure is presented in Figure 4.

B. The analysed colour models

In this subsection the analysed colour spaces and colour models (Gaussian and histogram based) are described.

a) Colour spaces: In the experiments several common colour spaces were used:

- RGB (*Red, Green, Blue*) – the basic colour space used in input (cameras) and output (displays) devices,
- YCbCr – a colourspace with a luminance (Y) and two chrominance (Cb, Cr) components. It is used in image and video stream compression (JPEG/MJPEG),
- CIE Lab – colour space similar to YCbCr, but defined to be perceptually uniform i.e. the distance (Euclidean) between two colour samples corresponds with the difference noted by a human.

The HSV (Hue, Saturation, Value) colour space was not analysed, because of the angular coordinates of the H component, which causes some difficulties in models using mean and standard deviation.

[1]This colours were available in the used dish set

TABLE I
CLASSIFICATION RESULTS FOR DIFFERENT FEATURE VECTORS

	circle		elipse		mug		rsquare	
Hu moments	Train	Validate	Train	Validate	Train	Validate	Train	Validate
1	99.9629 %	99.8656%	89.9703 %	90.5914 %	89.8217%	89.7849 %	99.9629 %	100 %
1,2	100 %	100 %	100 %	100 %	100 %	100 %	100 %	100 %
1,2,3	99.9629 %	100 %	100 %	100 %	100 %	100 %	100 %	100 %

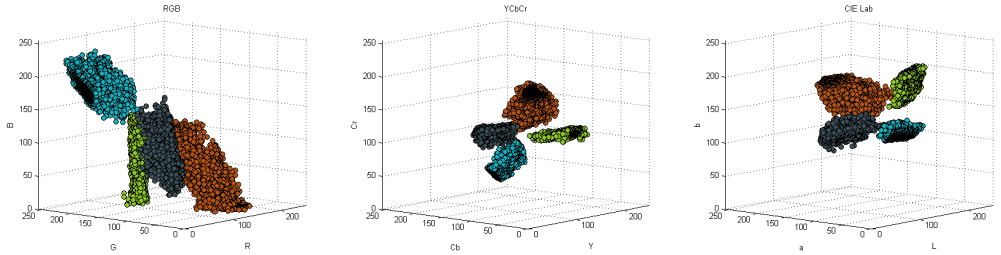

Fig. 5. Colour model samples displayed on a 3D scatter plot. From left: RGB, YCbCr, CIE Lab

TABLE II
STANDARD DEVIATION ANALYSIS FOR USED COLOUR SPACES

Colour	RGB	YCbCr	CIE Lab
orange	14352,61	2282,61	2351,64
blue	4030,69	932,80	150,01
green	3966,19	569,31	230,09
brown	11199,39	424,27	679,77

In the first step, the used colour samples were displayed on a 3D plot – Figure 5. On this basis it can be concluded that for the YCbCr and CIE Lab colour spaces the samples are more separated from each other than for RGB. This observation is also confirmed by calculating the standard deviations of the individual components of the samples. In Table II the product of standard deviations for each component is presented. It illustrates the dispersion degree of samples around the mean value. The analysis confirms the earlier observation that YCbCr and CIE Lab have better properties than RGB. It should also be noted that YCbCr and CIE Lab are quite comparable, with a slight indication of the latter. However, due to much more efficient conversion between RGB and YCbCr comapring to RGB and CIE Lab in the further analysis the YCbCr colour space is used.

The obtained colour samples are used to create a representation (model) for classification. In this work models based on Gaussian distribution, 1D histograms and 3D histograms were evaluated. More advanced approaches like Gaussian Mixture Models [14], artificial neural networks [15] or typical machine learning [16] approach were not considered. However, if in future versions of the system, more, especially quite similar, colours should be recognized, than this methods could provide better performance and reliability.

b) Gaussian distribution model: For each vector of training samples assigned to a particular colour, mean and standard deviation can be computed. On that basis a Gaussian model can be build. However, there are two options available. In general, it is assumed that the individual colour components are interdependent. Then the probability that a given pixel belongs to a particular model is given by:

$$p(x|\theta) = \frac{1}{(2\pi)^{n/2}\sqrt{(det\Sigma)}} e^{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)} \quad (1)$$

where: x – given pixel, θ – given colour, n – number of colour components (3), Σ – covariance matrix, μ – mean value vector.

It can be also assumed that the components are independent, which simplifies the equation:

$$p_c(x|\theta) = \frac{1}{\sqrt{(2\pi S)}} e^{-\frac{(x-\mu)^2}{2S}} \quad (2)$$

where: S – standard deviation.

In this case the probability is obtained separately for each component. Therefore, to define it for a pixel, the particular results should be summed up: $p(x|\theta) = p_{c1}(x|\theta) + p_{c2}(x|\theta) + p_{c3}(x|\theta)$.

Thus, for a given pixel the probability of belonging to a particular colour model is obtained. The computations are performed for the whole edge of the dish. At this point, there are two possibilities. First, individual decisions about colour assignment for each pixel can be made. Than the obtained "votes" can be summed up to perform a final classification. Another way involves the summing up of the probabilities and making the decision for the entire image. In the experiments, the second approach was used.

c) 1D histogram model: Another common used colour model is a histogram. It can be one-dimensional or three-dimensional (described in the next paragraph). In the first case the given colour is described by three separate histograms, one for each colour component. An important parameter of the method is the number of histogram bins. If the value is less than 256 (assuming that the colour component values are in range 0-255), then nearby values are aggregated.

Colour classification can be performed in two ways. In the first, using the pixel values as an "address" – the corresponding

histogram values are obtained and summed. The resulting number is a measure of probability that the pixel has a given colour (histogram normalization is assumed). The numbers, like in the Gaussian model, should be summed up for the whole image. Following this procedure for each colour model (i.e. 4×3 histograms) and then selecting the maximal value allows to perform the recognition.

In the second approach, a histogram is computed for edge pixels. Then it is compared with the model using a distance measure between two histograms. The most common are correlation, intersection and Bhattacharyya distance.

d) 3D histogram model: A three-dimensional histogram can also be used as a colour model. A certain disadvantage of this approach is its high memory complexity – in the default case the histogram has 256^3 bins. Therefore, aggregation (bin number reduction) is frequently used. The classification can be done in two ways – analogous to those described above for the 1D histogram.

C. Evaluation and results

When evaluating different colour models, two aspects should be addressed. Firstly, high classification accuracy is required. This means that for each test image the correct result should be obtained. Unfortunately, during research it turned out that this is not a sufficient criterion. The test image database does not include all possible cases that may occur during operation of the system. This applies particularly to potential changes in lightning. Therefore, it is advisable to propose and use an additional measure that will determine which model is better, even in the case when more than one has 100% accuracy on the test dataset.

In this research a classification certainty factor was used. For methods with a probability output the following coefficient can be computed: $cS = \frac{P(m)}{\sum P(c)}$, where: $P(c)$ denotes the probability than an object (i.e. a dish edge) belongs to the colour class c and $P(m)$ is the maximum of this factor. In the ideal case this value should be one, which indicates that all pixels were correctly classified.

For methods based on histogram comparison this coefficient should be modified. In case of correlation, a positive value is obtained when the classification is correct (positive correlation). For other colours the value is usually negative. Therefore, it seems to be eligible to use the absolute value and repeat the above described scheme. When using the histogram intersection, the cS coefficient can be used directly. However, for the Bhattacharyya distance it is necessary to subtract the final values from 1 ($cS = 1 - cS$).

The finally implemented model was selected after a series of experiments. A dataset of 64 images containing dishes in different workspace locations and orientations was used. Additionally, possible lightning conditions were simulated: brightening, darkening and adding gradient (using appropriate gamma correction).

In the first stage, the models with 100% classification accuracy on the test dataset were selected. This were the Gaussian model with dependencies between components and 3D histogram. However, it is worth noting that the remaining models obtained results ranging from 97% to 99%.

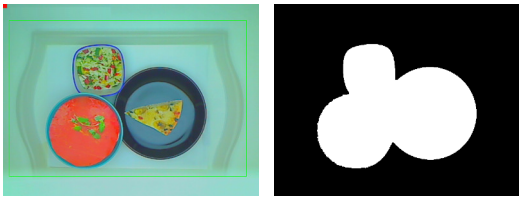

Fig. 6. Example of abut objects. On the left the input image, on the right the corresponding object mask

In the second stage the classification certainty coefficient was evaluated. For the Gaussian model the value $cS = 0.99$ was obtained. Similarly for the probability version of 3D histogram, as well as correlation, intersection and Bhattacharyya histogram distance (using 256, 128 and 64 bins). Therefore, all this approaches should be regarded as very robust.

It should be noted that there was no significant impact of the number of histogram bins on the classification performance. This could be due to the considered colours, which were quite different from each other (separated in the colour component space). Finally, the Gaussian model was used because of its lower memory complexity.

VI. Close or abut object recognition

In this section a procedure of shape and colour recognition for close or abut objects is described. An example of such a situation is presented in Figure 6.

It should be noted that in the initial vision system specification it was assumed that objects should be placed apart from each other i.e. on the object mask each of them should appear separately. Therefore, when close or abut object are detected, a message for the user is generated: "Please correct the positioning of the dishes". However, to speed-up the service and increase system capabilities an automatic procedure for this case was developed. The issue is quite complex and at least two main problems have to be addressed and solved:

- determining that two or more objects are segmented as one (are close or abut) – distinguishing this situation from typical arrangement of dishes on the tray,
- detecting abut objects with the same colour. In a general case the information about colour can not be used in this procedure, also due to possible influence of food or meal colour on the segmentation results.

A. Abut objects detection

A situation in which two or more objects are close or abut can be detected by analysing:

- object's shape – there is no recognition by the SVM classifier (i.e. the abut objects form a new shape, which is not similar to any other recognized by the system),
- object's size – it is larger than other recognized by the system.

Thus, if during the shape analysis an object with unrecognised shape and large area appears, it is regarded as a collection of abut or close objects. Here it is assumed, that several small objects can not "form" a large, recognized object (e.g. a big plate formed by several small mugs). Correct recognition in this case seems to be very difficult and would require more sophisticated segmentation algorithms.

B. Different concepts of separating objects

During preliminary research different methods of separating abut objects were analysed. The most straightforward solution is the use of information about edges, because this allows a human to correctly recognize connected objects. Unfortunately, the approach of: detecting edges using Sobel or Canny, thickening them and subtracting from the input mask does not allow to properly separate objects. Mainly due to discontinuities of detected edges and disturbances caused by meals or non-uniform lightning. Furthermore, the obtained object masks are usually smaller then the input ones, which causes difficulties during colour sample extraction.

Another promising solution is the Hough transform, but it has a quite high computational complexity, especially for shapes, whose analytical description requires multiple parameters (circle, ellipse) and it can not be used for all kind of shapes (non-analytical curves).

However, the use of some kind of "knowledge" about the recognized shapes seems to be the key to proper connected object segmentation. This is the result of the observation that knowledge of how the objects look like, in combination with edges and colour information make the task of separating objects quite easy for a human. Therefore, basing on the results published in [17] a distance transform (DT) based solution was used in the proposed system.

C. The proposed shape recognition method

The starting point for the shape recognition algorithm development was the analysis of the distance transform method described in [17]. It uses two of the mentioned observations: object edge analysis and pattern detection (i.e. knowledge).

In the first step of the method, for the input image edges are determined. The resulting binary edge mask is used to compute the so-called distance transform (DT). It is an image in which a pixel value represents the distance (Euclidean, chessboard, Manhattan, quasi-Euclidean) to the nearest edge. An example is presented in Figure 7c.

In the developed algorithm, extracting individual objects requires the use of shape templates in the form of an edge mask and a binary mask. The template is moved across the DT image using a sliding window approach. For each location, the following factor is computed:

$$D_{coef} = \frac{1}{|T|} \sum_{t \in T} DT(t); \tag{3}$$

where: $|T|$ – number of pixels for a given template, $DT(t)$ – distance transform value for pixel t from template T.

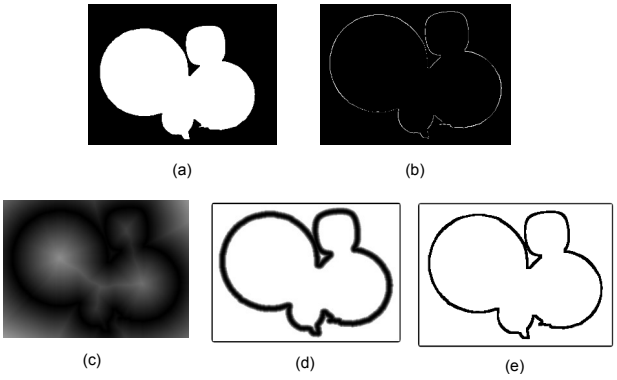

Fig. 7. Distance transform example: (a) input image – object mask, (b) binary edge mask, (c) DT result, (d) modified DT result, (e) binary DT result

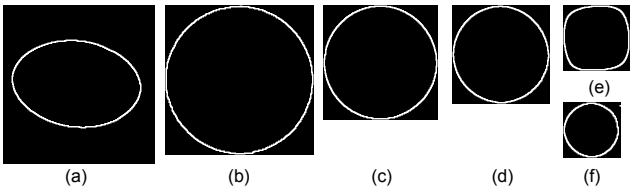

Fig. 8. Used shape templates: (a) – elipse, (b) – big plate, (c) – small plate, (d) – bowl, (e) – small bowl (rounded square), mug (without handle)

The location with the best match for a given template is characterized by the smallest D_{coef} value.

In the general case, to ensure proper operation for this type of algorithm, template modifications like translation, scaling and rotation are required. Translation is "built-in" in the sliding window approach. Scaling is not required, because the distance between the camera and object is fixed. However, rotating the template is necessary for selected, asymmetrical shapes like an ellipse.

The used shape templates are presented in Figure 8. It should be noted, that for the *mug* template the shape was simplified to a circle (the handle was omitted). This eliminates the rotation necessity, but does not affect the recognition performance, as there are no objects of similar size and shape in the system.

During preliminary research three variants of the DT based approach were considered: basic DT, modified DT and DT reduced to a binary mask. In the modified version the initial DT image was transformed according to the following formula: $DT(i,j) = DT(i,j)^{\alpha}$ (during tests $\alpha = 2$). Such an approach promotes locations with good template match. An example image is presented in Figure 7d.

A further simplification is the assumption that the DT image has a binary form – in a small neighbourhood of the edges the DT values are set to 0 and in the remaining part to a maximal value. This further enhances the promotion of good-match locations. An example is presented in Figure 7e. It is worth noting, that in this case computing the "DT" image involves only performing a morphological dilation of the edge mask,

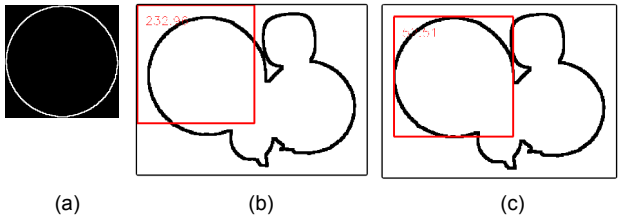

Fig. 9. Template search with sliding window approach: (a) template, (b) initial situation ($D_{coef} = 232.96$), (c) best match ($D_{coef} = 52.51$)

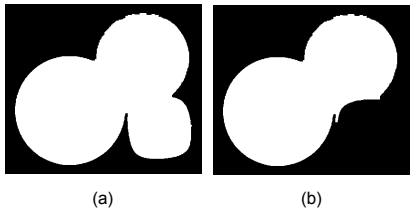

Fig. 10. Shape removal example: (a) – input mask, (b) – mask after shape removal (*rsquare*).

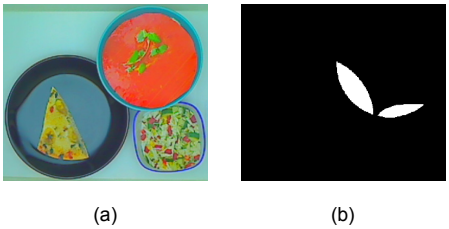

Fig. 11. Example of obscuration area detection: (a) – input image, (b) – obscuration mask.

which greatly improves the computational efficiency.

Initial experiments demonstrated a similar performance of the three described DT variants. Therefore, in the final version of the system, the binary DT approach was used, since it is most computationally efficient. However, it is worth considering why such a simple solution obtains comparable results with the "full" DT transform. First, in the described system, the algorithm operates only on the object's outer edge mask, whereas the original solution used all edges of the input image. This significantly limits the number of edges that could potentially affect the shape recognition. Second, the used templates are quite simple, rigid, with well defined size and only slight variation caused by perspective issues. Finally, due to strictly controlled lighting conditions the object's boarder segmentation is very reliable.

D. Shape recognition procedure for connected objects

During designing the solution, the primary goal was to ensure high recognition reliability. Therefore, an iterative search procedure was proposed. In a single iteration one most probable shape is found.

First, the edge mask and the DT binary image are computed. Next, for each shape template the sliding window procedure is applied and the best location (lowest D_{coef}) stored. At the configuration stage, it can be determined if a particular template should be rotated. In the current version of the system, the rotation is only performed for the *ellipse* with 30° step, which results in 5 different templates. For the non-symmetrical template *rsquare* it turned out that no rotation is necessary. Example of the search procedure is presented in Figure 9.

After obtaining the D_{coef} (compare Equation (3)) values for all used shape templates, the minimum is selected. In this way, the most probable shape is detected.

In the next step this shape is removed from the object mask. The new mask is subjected to filtration. This involves the removal of connected components with area smaller than a preset threshold (so-called area open) and morphological processing. An example in shown in Figure 10. The described iteration is repeated until all object are removed.

a) Template localization improvement: During experiments it was noted that the obtained, in the above described procedure, shape locations are not very accurate. This has a great impact on the subsequent colour extraction method, particularly for dishes with narrow edge like a bowl. In order to compensate this error a simple template position improvement method was proposed. First, using the initial location estimate, a ROI from the input object mask is extracted. It size is slightly bigger than those of the considered template. Next, in this ROI a sliding window approach is used to find the location, where subtracting the template mask from the object mask results in removing maximal number of pixels. This is then regarded as the improved object location.

b) Analysis of the detected objects: The result of the above described procedure is the information about all detected objects i.e. their shape and location. For the purpose of colour recognition edge extraction should be performed. However, in case of close or abut objects often some obscuration occurs. For example a bowl could cover a part of a plate. Extracting colour samples in this areas could lead to errors.

Therefore, such regions should be detected and excluded from further analysis. This is done using a so-called obscuration mask – example presented in Figure 11.

In the next step, for the detected objects, edge masks are obtained. This is done in three steps:

- determining the logical AND (intersection) of the objects mask and template mask,
- extraction the edge of this object (using the described prior morphological approach),
- determining the logical AND of the edge mask with the negation of the obscuration mask.

The obtained masks are used in the colour recognition procedure. An example of the described approach is presented in Figure 12. It is worth to notice, that despite some minor errors in determining the localization, the final detection is correct.

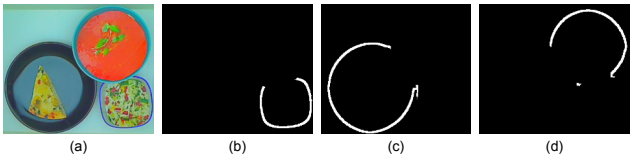

Fig. 12. Example of abut objects splitting. (a) – input image, (b), (c), (d) – edge masks for the detected objects

Fig. 13. Examples of test images used in evaluation of the close or abut object recognition procedure

E. Evaluation and parameter selection

The impact of different parameters on the connected object separation procedure was evaluated. For this purpose 30 test images with different close or abut objects were used. Two examples are presented in Figure 13.

The accuracy of the procedure was evaluated by comparing the returned shape and colour recognition with reference ground truth (more details in Section VII). The following parameters of the method were analysed: template edge thickness, image resolution, sliding window step.

For the first parameter it was observed, that above a certain edge thickness value, the obtained results were incorrect. Finally, for edge extraction an erosion with 7×7 square structuring element was used. This results in 3-4 pixels thick edges. In case of reducing the input image resolution, the size of structuring element should also be reduced.

The resolution of the analysed image (object mask) has a great impact on the performance of the described procedure. This is directly related to the sliding window approach and D_{coef} calculation. Therefore, reducing the image size brings significant acceleration to the entire vision system. In the experiments it was determined that changing the size from 640×480 to 192×144 (30% of the input value) does not result in loss of accuracy – all objects present on test images were correctly recognized.

The sliding window step value is a compromise between template location accuracy and calculation speed. During experiments it was determined, that the value 4 pixel (vertical and horizontal) is suitable for the used resolution of 192×144.

Additionally, in an experiment is was proved that switching off the template localization improvement procedure leads to errors in colour classification for the bowl shape.

VII. Vision System Evaluation

The developed vision system was subjected to a series of test on the designed stand (compare Section II). Their goal was to evaluate basic functionalities like:

- proper tray position detection,
- empty workspace detection,
- presence of hand in the workspace detection,
- object segmentation,
- shape and colour recognition,
- recognition of close or abut objects.

In addition, elements of the algorithms that require further improvement and modification were identified.

Evaluation of the first three functionalities were conducted on-line i.e. the system behaviour was observed during real-time analysis of the video stream acquired by the camera. The tray position was detected correctly. The only potential problem that occurred was the partial visibility of a marker (the marker was not fully covered by the tray). However, even if it was detected as an object, due to small size it was excluded from further analysis. In addition, it is assumed that the user will co-operate i.e. place the tray as accurate as possible (supported by the feedback provided by the system).

The empty workspace detection module worked correctly in case of proper camera placement and marker position calibration. However, it should be noted that presence of some kind of dirt in the workspace may cause errors – be recognized as an object. Consequently, the background model update mechanism will fail. It is worth mentioning that a frequent background model update is necessary for proper segmentation. During experiments, despite the used LED illuminator, the external lighting changes had impact on the segmentation.

The presence of hand in the workspace detection works correctly. The only exceptional case is when the user wears a sweater or sweatshirt with long sleeves in colour similar to the workspace. In the future, skin-colour regions segmentation could be considered. Whereby, the result interpretation could be quite difficult – some food could have colour similar to skin.

Other functionalities were tested both on-line and off-line on selected test images.

A. Test images annotation

In order to automate the vision system evaluation and parameter selection process a tool for object annotation was designed. Each object (dish) present on a test image can be described by: location (rectangle inscribed in the object), shape, colour.

For the purpose of frame annotation, a simple GUI application was created. It allows to browse a directory with images and set information about location, shape and colour of different dishes.

B. Object segmentation evaluation

The used segmentation method worked correctly when the background model was up to date. The only observed problems occurred when cast shadows, present due to specific dish location, caused incorrect operation of the abut object recognition procedure.

TABLE III
SHAPE AND COLOUR OF DISHES USED IN THE EVALUATION.

Shape name	C 1	C 2	C 3	C 4
`circle_huge` [big plate]	brown	blue	green	
`circle_big` [small plate]	brown	blue	green	orange
`circle_small` [bowl]	brown	blue	green	orange
`mug` [mug with handle]	brown	blue		orange
`rsquare` [small bowl, plate]	—-			
`elipse` [platter]	—-			

C. Object shape and colour recognition evaluation

Evaluation of the object shape and colour recognition process was performed by comparing the results returned by the application with prepared manual annotation (ground truth). A series of test scenarios was prepared: from simple one with one dish, to complex involving close or abut objects, as well as presence of additional objects (like napkins or cutlery). The used dishes – they shapes and colours – are summarized in Table III.

For the test procedure 81 images with single dish, 9 with multiple dishes (not close) and 33 with close or abut dishes were prepared. The current version of the system returned an incorrect result in 2 out of 123 cases (accuracy 98%). Both errors were related to a quite significant shape change due to location far from the centre of the camera's optical axis. This could be eliminated by proper camera calibration.

VIII. CONCLUSIONS

In this paper a vision system able to recognize the shape and colour of objects was described. It can be used to automate to process of customer service in a self-service canteen. It is built of four main modules: object segmentation, shape classification, colour classification and recognition of close or abut objects. When designing each module, different possibilities were considered and those with high efficiency and low computational complexity were selected. The final system meets the design constrains i.e. accuracy above 95% and computing time < 1s on an typical PC.

The solution can be further developed in several directions. One of them is the broadly understood relaxation of restrictions. A good example is the use of tray with colour different than the workspace (i.e. not "transparent" for the segmentation). Another one would be the determination of the minimum edge thickness required for proper colour recognition. This would allow to use dishes with only a thin colour boarder instead of "full" colour and therefore use more different colours.

In addition, the use of a lens distortion correction module or another camera should be considered. In a long term perspective, a food recognition module could also be added. An interesting and promising direction is the development of a smart camera vision system to perform the described image processing, analysis and recognition.

ACKNOWLEDGEMENTS

The work presented in this paper was supported by the Malopolska Regional Development Agency under the program "Knowledge, Practice, Experience – the key to success in business" in cooperation with Qumak S.A., Kobierzynska 2, 30-363 Krakow

The authors wish to thank Qumak S.A. Krakow company, and particularly Mr. Konrad Pogódź for supporting this research, as well as Dr. Zbigniew Mikrut for help and many valuable comments.

REFERENCES

[1] Y. Kawano and K. Yanai, "Real-Time Mobile Food Recognition System," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2013 IEEE Conference on*, June 2013. doi: 10.1109/CVPRW.2013.5 pp. 1–7.

[2] F. Zhu, M. Bosch, N. Khanna, C. Boushey, and E. Delp, "Multiple Hypotheses Image Segmentation and Classification With Application to Dietary Assessment," *Biomedical and Health Informatics, IEEE Journal of*, vol. 19, no. 1, pp. 377–388, Jan 2015. doi: 10.1109/JBHI.2014.2304925

[3] V. Bettadapura, E. Thomaz, A. Parnami, G. Abowd, and I. Essa, "Leveraging Context to Support Automated Food Recognition in Restaurants," in *Applications of Computer Vision (WACV), 2015 IEEE Winter Conference on*, Jan 2015. doi: 10.1109/WACV.2015.83 pp. 580–587.

[4] H. He, F. Kong, and J. Tan, "DietCam: Multi-View Food Recognition Using a Multi-Kernel SVM," *Biomedical and Health Informatics, IEEE Journal of*, vol. PP, no. 99, pp. 1–1, 2015. doi: 10.1109/JBHI.2015.2419251

[5] SRI, "`http://www.sri.com/engage/products-solutions/food-recognition-technology` (last access 03.05.2015)."

[6] BrainCorporation, "`http://www.diginfo.tv/v/12-0145-r-en.php` (last access 03.05.2015)."

[7] OpenCV, "`opencv.org` (last access 03.05.2015)."

[8] QT, "`www.qt.io` (last access 03.05.2015)."

[9] T. Kryjak, "Segmentation of dishes for the purposes of customer service process automation in a self-service canteen (under review)," 2015.

[10] A. Amanatiadis, V. Kaburlasos, A. Gasteratos, and S. Papadakis, "Evaluation of shape descriptors for shape-based image retrieval," *Image Processing, IET*, vol. 5, no. 5, pp. 493–499, August 2011. doi: 10.1049/iet-ipr.2009.0246

[11] M.-K. Hu, "Visual pattern recognition by moment invariants," *Information Theory, IRE Transactions on*, vol. 8, no. 2, pp. 179–187, February 1962. doi: 10.1109/TIT.1962.1057692

[12] V. Megavannan, B. Agarwal, and R. Venkatesh Babu, "Human action recognition using depth maps," in *Signal Processing and Communications (SPCOM), 2012 International Conference on*, July 2012. doi: 10.1109/SPCOM.2012.6290032 pp. 1–5.

[13] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, Sep. 1995. doi: 10.1023/A:1022627411411. [Online]. Available: http://dx.doi.org/10.1023/A:1022627411411

[14] Z. Fu and L. Wang, "Color Image Segmentation Using Gaussian Mixture Model and EM Algorithm," in *Multimedia and Signal Processing*, ser. Communications in Computer and Information Science, F. Wang, J. Lei, R. Lau, and J. Zhang, Eds. Springer Berlin Heidelberg, 2012, vol. 346, pp. 61–66. ISBN 978-3-642-35285-0. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-35286-7_9

[15] S. Mikrut, Z. Mikrut, A. Moskal, and E. Pastucha, "Detection and recognition of selected class railway signs," *Image Processing & Communications.*, vol. 19, no. 2-3, p. 83–96, 2015. doi: 10.1515/ipc-2015-0013

[16] M. Querini and G. Italiano, "Color classifiers for 2d color barcodes," in *Computer Science and Information Systems (FedCSIS), 2013 Federated Conference on*, Sept 2013, pp. 611–618.

[17] D. Gavrila, "A bayesian, exemplar-based approach to hierarchical shape matching," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 29, no. 8, pp. 1408–1421, Aug 2007. doi: 10.1109/TPAMI.2007.1062