

# VISION-BASED SYSTEM FOR QUALITY CONTROL OF SOME FOOD PRODUCTS

O. Alhusain <sup>a,b,\*</sup>, Z. Tóth <sup>a</sup>, Á. Rakusz <sup>a</sup>, L. Almási <sup>a</sup>, B. Farkas <sup>c</sup>

<sup>a</sup> University of Technology and Economics, Department of Photogrammetry and Geoinformatics, H- 1521 Muegyetem rkp. 3., Budapest, Hungary - [alhusain@eik.bme.hu](mailto:alhusain@eik.bme.hu), [tothzoli@index.hu](mailto:tothzoli@index.hu), [aaaah@mailbox.hu](mailto:aaaah@mailbox.hu)

<sup>b</sup> HAS-TUB Geoinformation Research Group, H- 1521 Muegyetem rkp. 3., Budapest, Hungary

<sup>c</sup> Miklós Zrínyi National Defence University, H- 1581, pf. 15, Budapest, Hungary - [farkasb@zmne.hu](mailto:farkasb@zmne.hu)

## Commission III, WG III/5

**KEY WORDS:** Industry, Inspection, Pattern, Recognition, Vision

### ABSTRACT:

Mass production of crop, produce, and food stuff has resulted in great increase in the efficiency of food producing plants. This in turn has led to considerable decline in food prices. However, the mass production of food was also associated with two major problems. The first one is the decline in food quality, and the second one is the “waste” problem associated with processing and preparation operations. The wastage in many cases is a direct consequence of the quality problem, where the quality decline reaches unaccepted limits. Hence, is the need for quality inspection and assurance mechanisms to be installed in the production lines of such mass food processing and producing plants.

In this paper, quality control and inspection system relying on image processing techniques will be discussed and presented in details. 2D and 3D visual characteristics are collected about three kinds of food products that are in one way or another go through a mass production process. These products are Arabic style pita bread and Mexican tortillas. Visual characteristics of interest for this study are collected during the baking phase for the first two products and during classifying and sorting process for the last one. Characters measured for the products were size (width, length, volume, and area), shape, and color (dominant color, localized colors, average color).

Imaging system utilizing 5-megapixel digital camera was used to acquire the images for this study. Different image processing procedures like filtering, binarizing, zoning, masking, and enhancement were used to derive the quality control parameters from the visual characters of the products under investigation. This study has shown that machine-based inspection of food products can be implemented effectively, reducing or even eliminating the need for both intensive human intervention and addition of conditioning chemicals to assure quality. The concept and results presented in this study can be applied in solving more complicated pattern recognition problems.

## 1. INTRODUCTION

Recent developments in machine vision and supporting technologies has resulted in general acceptance of the feasibility and profitability of implementing visual inspecting systems in quality assurance operations of food producing lines. Machine vision benefited the most from the increase in processing and storage powers of modern chips, and from the emergence of megapixel sensing and imaging devices. Machine vision technology utilizes image processing techniques for the purpose of extracting visual features about an object for a variety of qualitative, quantitative and control applications.

During the last few years, the trend in designing and building machine-based inspection systems has almost always followed one of two patterns. These are the traditional sensor (camera) – computer – interface pattern, and a newer pattern– the so called smart camera pattern. The two patterns are widely presented and deeply discussed in recent literature ( ICD, 2003; Wilson, 2003).

### 1.1 Sensor (camera) – computer – interface pattern

The inspection system consists of four distinct entities, these are:

1. Sensor system: Which is generally CCD or CMOS camera, the resolution in these cameras vary from one system to the other, common resolutions found in these systems range from sub mega pixel on the low side to almost 10-20 mega pixels on the high side. Inspection data is fed to a computer system for processing, evaluation and decision making.
2. The computing facility: In most cases it comprises a complete computer system with its operation system, application programs and/or task-specific programs (library). The hardware is built around Pentium processors or power PC ones. However, it seems to be that Pentium-dependant implementations are more common in scientific laboratories and academia mostly for flexibility reasons. While power PC implementations are more popular in real-life production lines mainly for stability characters of software implemented on these systems. Nevertheless, operating efforts and kills required to operate a power PC – based system are less demanding than those required to operate a similar Pentium – based system.
3. Input/Output, and Networking facility: These facilities are easy to implement since they rely on utilizing the computer system’s facilities to achieve this goal. In this regard the communication between the computer

---

\* Corresponding author.

system and production lines is of crucial importance, since the collection of visual features is fed to the computer system for processing, analysis and decision making, the decision made is ought to be fed back to the production lines.

4. The frame generating rate: Which is one of the characteristics of these systems rather than one of their components. The frame generating rate in these systems ranges from few frames to few thousands frames per second. As a general rule, the higher the frame rate is the better, but raising the frame rate in many cases happens on the „cost” of the resolution, so choosing a system that balance the resolution and the frame rate suiting both of them to the application at hand is of at most importance.

## 1.2 Embedded closed systems

These systems are known by their popular name, the “smart cameras”. In these smart camera systems a CCD or CMOS sensors coupled with processor (CPU or RISC), DRAM memory, and In/Out facilities are incorporated and embedded in one package making for the smart camera. Because smart cameras function as complete vision system, and because it is almost always impossible to modify the build set up of these cameras, so their individual characteristics have to be evaluated fully and thoroughly when the task of implementing such cameras comes to one’s hands. The determinant characteristics of smart camera that ought to go through study and evaluation process before choosing a smart camera for an inspection task at hand are the following:

1. Sensor resolution: Sensors implemented in smart cameras usually one of two kinds, either CCD or CMOS, although CCD sensors seem to be more common in the smart cameras. The resolution of these sensors is still in the range of submegapixel to few megapixels, sometimes they exceed this resolution but this generally happens on the expense of spectrum richness.
2. Embedded Processor and memory: The processing requirements in smart cameras are carried out by an embedded processor rather than a separate computer. The most common processor implemented in smart cameras is the Motorola Power PC, it is not uncommon to find the Intel Processors built in these cameras too. Whether it were Power PC or Ix86 Processors these provide the designers of these smart cameras with ample tools of computing and control abilities. Sometimes these embedded processors are supported by task – specific facilities to boost their performance like floating point processors, digital signal processors, and the like. Beside the processor the smart cameras are usually provided with 16 – 32 MByte DRAM memory to provide the processor (and the camera) with fast processing and storage facility.
3. Application Programming Interface (API): These are software packages, mostly written in C/C++ programming language, and developed mainly to perform machine vision tasks, like extracting useful data from captured images, processing the extracted data, and generating reaction data (parameters) accordingly. If the machine vision tasks are well defined, than the application programming interface (API) program can be satisfactorily designed and developed enabling the smart camera to provide an effective alternative for PC based open systems.

However, once the API package is designed and developed, it will be difficult if not impossible to carry any change or modification on the package, a property that turns smart cameras to a nonflexible machines.

4. Programmable I/O Facilities: Programmable Input/Output facilities are an important tool that enables setting the smart camera parameters to suite achieving the task of machine vision at hand. The programmable I/O plays a mediatory role between conditions present at production environment and reference conditions and actions.
5. Video Output, Networking, and Interfaces: These are tools to enable access, communication and graphic data and event feed to/from the camera from/to the outside surroundings. These tools include but not limited to configurations like SUGA output, serial and/or parallel interfaces, digital I/O lines, Ethernet connections, Hotlink lines, etc.

In few words, since smart cameras are generally deployed to perform specific task, they excel when the task required to be served is very well defined and can be expressed in terms of well designed series of data extracting and handling steps.

On the contrary of this, if the task at hand can’t be defined in specific steps, or if the conditions associated with carrying out the task have the tendency to change in the future, or if the task to be achieved itself might be modified, then it is better to opt for an open machine vision system, which makes distinction between the imaging unit of the machine vision system and the computer system that guides, controls and handles the whole machine vision system.

In this study, choosing flexible open-structure machine-vision system was adopted due to the fact that this study had two main goals. First, the investigation of the possibilities of implementing quality assurance system as part of the production lines of pita and tortilla breads. Second, developing procedures for collecting visual features of these products. These procedures might be generalized and later applied in more complicated pattern recognition applications.

## 2. SIZE AND SHAPE MEASUREMENTS

Pita bread and tortillas are similar in shape and colour, the differences between these two kinds of breads is that pita bread has two layers like a pouch or pocket while tortilla has one. Consequently, during backing phase pita bread will get a spherical shape while tortillas remain flat. However, when pita bread cools down the upper layer which is the one having spherical shape collapses down to the lower layer which remains flat during backing. This collapsing process results in flattening both layers, this process happens within about five minutes after the loaves roll out of the backing belt, and it is most apparent if the loaves are stacked over each other. As a result any measurement on the shape of pita bread has to take place immediately after they roll out of backing belt (before collapsing). The other difference between pita and tortilla breads is that the dominant colour in pita is lighter brown with darker brown spots, while tortillas have a light colour as the dominant colour with darker brown spots.

In relation to visual features measurements, the procedures and methods of collecting measurements data are the same in both kinds of breads, so the discussion here will be related only to pita bread, while it can be generalized to tortillas without any change. The size and shape measurements carried were the following:

## 2.1 Diameter and Radius

The pita bread diameter (D) can be measured from an upper view (or front view) image of the loaf. If an upper view image is used to measure the diameter, then the number of pixels from one point on the perimeter of the loaf to its counterpart point on the other end of the perimeter can be calculated by moving in 45 degrees toward the centre point of the loaf circle. If the number of pixels calculated is multiplied by the pixel's radius (R) of the loaf which is half the diameter ( $R=D/2$ ), figure 1. If the diameter (D) is calculated from a front view image of the loaf, then the diameter will be in this case equal to the number of pixels along one horizontal line (the diameter itself) extending between the most left point and the most right point of the loaf's perimeter, multiplied by the pixel width.

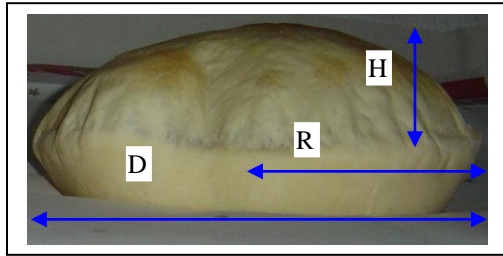


Figure 1. Arabic pita bread

## 2.2 Height

The loaf height is the highest point in the loaf. It can be determined by calculating the number of pixels along a vertical line extending from the peak point of the loaf and base line of the loaf and multiplying the number of pixels by the pixel's height, figure 1.

## 2.3 Loaf area

Loaf area can be calculated in one of two ways. The first one simple calculation in which the loaf is considered to be a flat disk, the area simply calculated as:

$$S_{simple} = \pi \times R^2 \quad (1)$$

In the second and more accurate case, the loaf is considered to be a sphere sector; this case is true right after the baking phase and before the collapse of the loaf. Here the loaf's surface (face) area is calculated as:

$$S_{up} = \pi \times (R^2 + H^2) \quad (2)$$

## 2.4 Loaf volume

Loaf crust volume is calculated according to the following equation:

$$V = \pi \times H \times (3 \times R^2 + H^2) / 6 \quad (3)$$

## 3. VISUAL FEATURES EXTRACTION

### 3.1 Colour Features Measurements

At the beginning and during the backing phase, all conditions are set to produce pita bread loaf with ideal characters. Conditions such as dough's water content (humidity), thickness of loaf dough, movement speed of backing belts, and backing heat are all important factors in producing ideal crusty brownish pita bread.

In many cases, however, the resultant dominant backing colour of the bread is not always the same as the wanted backing colour. The ideal backing colour is determined by choosing samples from areas that were successfully backed to the wanted brownish colour, then, the average value of the grey level of these samples is determined and adopted as the grey level representing the ideal backing colour.

Dominant colour is defined as a spectrum range with minimum and maximum grey level values centred around the grey level value associated with the greatest number of pixel found in the pita bread image. The two maximum/minimum thresholds can be adjusted at will; this will result in significant flexibility in determining the dominant colour by including/excluding wider range of the colour spectrum (grey level). Light and dark areas in the pita bread loaf are those areas exposed to less or excessive backing heat. In the context of this study they are defined as the summation of lighter/darker pixels deviated from the ideal backing threshold.

The average colour of the whole pita bread's loaf is an important factor in determining the success in obtaining the required backing colour or approximating that goal. The average colour of the whole pita bread's loaf (Average Grey Scale: AGS) is determined by computing the summation of the grey levels of all pixels (Grey Scale of Pixels: GSP) and dividing it by the number of pixels in the loaf (Number of All Pixels: NAP). The following equation illustrates the calculation of the average grey level of the loaf.

$$AGS = \frac{\sum GSP}{NAP} \quad (4)$$

### 3.2 Image Processing Procedures

Visual feature extraction was carried out on the pita bread images using a combination of program code developed specifically for this project and the commercially available image processing extension package of the MathCAD 2000 program. Image processing procedures that were implemented in this study are the followings:

**3.2.1. Reading Image Files:** Images of the incoming loafs of bread are captured in the JPEG image file format. Then the binary data is converted to a matrix carrying the values of the grey scale of the pixels as its elements, and each element in this matrix will be associated with row and column coordinates. If the image captured is saved as a row monochrome image, then the image can simply be converted to an (M\*N) element matrix, where M is the image's rows, and N is the image's columns. On the other hand if the image captured is saved as RGB coloured image, then the image can be converted to an (M x 3N) element matrix, where M is the image's rows, and 3\*N is the image's columns. This apparent triplication of the columns

is due to the fact that each colour component (R,G,B) is stored in separate section of the matrix (e.g. the Red component is stored in first third of the matrix, the Green component is stored in the second third of the matrix, and the Blue component is stored in the third (last) third of the matrix.

**3.2.2. Segmentation of Pita images:** Segmentation is a concept in which an image is subdivided into its constituent regions or objects. The level to which the subdivision is carried depends on the goal of carrying out the segmentation process (Gonzalez and Woods,1992; Parker,1991; Perez and Gonzalez, 1987). Segmentation procedures applied on monochrome images generally utilize one of two properties of the gray level values for the segmentation process, these are discontinuity and similarity. If the discontinuity is utilized, then the approach is to partition an image based on abrupt changes in gray level. The main areas of interest within this category are detection of isolated points, lines, and edges in an image. On the other hand if the similarity property is utilized, then factors such as region growing, region splitting, and thresholding processes can be applied. (Rosenfeld and Kak, 1982; Prat, 1978; White and Rohrer, 1983). The segmentation process mentioned here was applied only when the other algorithms couldn't provide satisfactory measurements.

Majority of image processing operations in this study were carried out by the use of the Image Processing Extension Package and its conventions of the MathCAD Professional 2000 Program, © MathSoft Copyright (Mathsoft, 1999).

To carry out segmentation, first a mask must be constructed to delineate the edge of the pita loaf and within that edge to delineate the border of different objects in the bread. Areas that were darkened by baking were delineated easily by segmentation process. To carry out segmentation, the image of the loaf was first binarized, and then upper and lower thresholds were chosen to include the loaf's entity and exclude its background. If the outside edge delineation of loaf is not satisfactory, then erosion can be applied to enhance the delineation process.

Figures 3 and 4, show binarized image of the pita bread and its eroded version, it is evident in these images that the background was separated from the front of the image. After binarizing the geometric measurements are carried out to determine the diameter, the radius, and height of the loaf, and consequently the surface area and the volume of loaf before collapsing, simple area of loaf (after collapsing) can be measured too. NPLR is abbreviated for Number of Pixels in the Longest Row in the image, and NPLC is abbreviated for Number of Pixels in the Longest Column in the image. If one pixel in loaf image is considered to be having dimensions of (width x length= area) synonymous to (1 x 1 = 1), then the measurements can be done as follows:

Loaf diameter:

$$D = NPLR \quad (5)$$

Loaf radius:

$$R = D / 2 \quad (6)$$

Loaf Height

$$H = NPLC \quad (7)$$

Loaf upper surface area:

$$S_{up} = \pi \times (R^2 + H^2) \quad (8)$$

Loaf simple surface area:

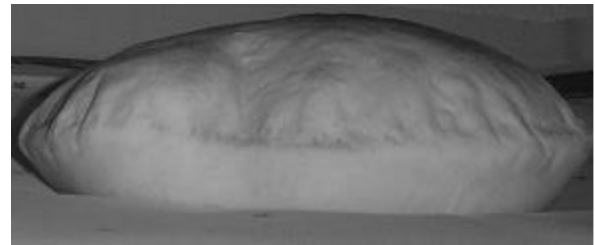
$$S_{simple} = \pi \times R^2 \quad (9)$$

Loaf volume:

$$V = \pi \times H \times (3 \times R^2 + H^2) / 6 \quad (10)$$

The colour and extension of the dominant colour in the loaf can be determined by applying labeling process on the image and choosing the largest component in the labeled image and considering it the dominant colour, then setting upper and lower threshold for the gray scale to be included in the dominant colour, and calculating the number of pixels in this component. First the pita bread loaf entity was separated from the background of the image. One good and practical way of separating the front from the background of the loaf image is to associate region with its colour and implement the hue saturation vector of the whole image, then carry out a binarization process. This can be done by converting the RGB representation of the image to HSV and applying an extraction process to emphasize the association of region with its corresponding colour (Zhoi, Chalana, and Kim, 1998).

```
Pita1 := READBMP("C:\Pics\Fig1 2.bmp" )
```



```
Pita1 := READ_RGB("C:\Pics\Fig1 2.bmp" )
```

```
Pita1 := extract(rgb_to_hsv (Pita1), 2)
```

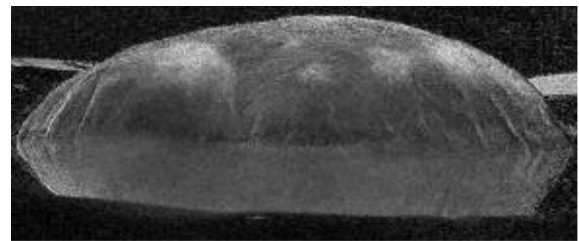


Figure 2. RGB and hue saturation images of pita bread

To separate the forefront of the image from its background, a binarization process at a suitable gray scale threshold is applied (Zhoi, Chalana, and Kim, 1998), then holes, spots, noise, and

non-consistency in the binarized image are eliminated by carrying out a morphological structuring on the image as following:

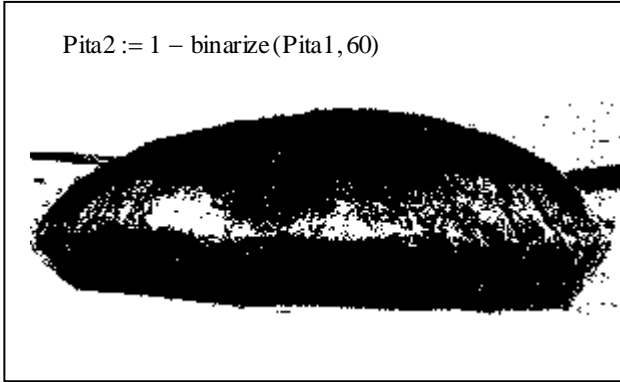


Figure 3. Binary image of the pita bread

```

i := 0..4          j := 0..4          msi,j := 0
msi,j := if[(i - 4)2 + (j - 4)2 < 10, 1, 0]
Pita3 := open (Pita2, ms, 1)
Pita4 := close (Pita3, ms, 1)

```



Figure 4. Morphologically structured image of the pita bread

From the binarized image, the surface area of the pita loaf can be measured easily as follows:

```

nr := rows(Pita4)
nc := cols(Pita4)
Area := ∑i=0nr-1 ∑j=0nc-1 Pita4i,j
Area = 10975

```

The colour and extension of the dominant colour in the loaf was determined by applying masking process on the hue saturation image of the loaf using the binarized image for the mask, then

binarizing the resulting masked image using upper and lower thresholds to define the dominant colour. The extension of the dominant colour is determined by subtraction of the two threshold images. The area of the dominant colour and its percentage compared to the loaf area was calculated according to the following equations:

```

UT := 240      LT := 10
Pita5 := mask(Pita1, Pita4)
ThresholdU := binarize(Pita5, UT)
ThresholdL := binarize(Pita5, LT)
Dominant_Colour := ThresholdL - ThresholdU

```

$$\text{Dominant\_Area} := \sum_{i=0}^{nr-1} \sum_{j=0}^{nc-1} \text{Dominant\_Colour}_{i,j}$$

Dominant\_Area = 9609

Percentage of dominant colour:

$$\text{Percentage\_of\_Dominant\_Colour} := \frac{\text{Dominant\_Area}}{\text{Area}} \cdot 100$$

Percentage\_of\_Dominant\_Colour = 87.554

The average gray scale of the pita bread can be calculated as the summation of gray scales larger than zero and dividing this sum by the number of pixels having this "larger than zero" gray scale, the calculation is done according to the following function:

```

Av_GS (IMG) := | k ← 0
                  AV ← 0
                  for i ∈ 0..rows(IMG) - 1
                    for j ∈ 0..cols(IMG) - 1
                      if IMGi,j > 0
                        | AV ← AV + IMGi,j
                        | k ← k + 1
                  return AV / k

```

Av\_GS(Pita1) = 66.272

The gray scale and extension of localized colours can be determined and calculated in the same way as in the case of dominant colour, where upper and lower thresholds for the localized colour has to be chosen, then the subimage of the localized colour is determined by subtracting the lower threshold subimage from the upper threshold subimage. The area of the localized colour is calculated in the same way used in the case of the dominant colour except that the subimage of

the localized colour has to be used. Similarly, the average gray scale for the localized colour can be calculated by the summation of gray scales falling within the limits of the upper and lower thresholds for the localized colour and dividing this sum by the number of their corresponding pixels.

#### 4. CONCLUSIONS

The goal of this study was to investigate specific phases of the design and implementation of vision-based inspection system to improve the quality of some baked products namely Arabic Pita and Mexican tortilla breads, and to improve the efficiency of production lines of these products. The investigation, in this context, concentrated on developing and implementing methods and procedures to measure, process and analyze visual data and geometric characteristics of baked products that can be used to implement a vision-based quality control system for these baked products. From the experience gained in this project, it is clear that the visual inspection process involves three main distinct stages; these are visual data collection, data processing and feature extraction, and control decision making. This investigation was done concentrating on the first two stages.

The core of the visual system for this study was a 5-megapixel camera, which facilitated the acquisition of high resolution images up to (1500x3000) pixels. Although these high resolution capabilities enable the production of high definition images with the finest of details, however, in relation to the available hardware and its computing capabilities, and specially when trying to use a C++ code to carry out the image processing operations, it was clear that the produced images exceed the capabilities of the visual system's unit responsible for software execution and feature extraction. In many cases and in order to be able to process the images in near-real time, it was unavoidable but to crop the images around the objects that undergo feature extraction process, this cropping process results in reducing the size of image matrices to be processed so speeding up different processing and analysis operations.

Feature extraction procedures developed for this investigation are suitable for many integrated shape-colour pattern recognition applications. A preliminary investigation will be carried out on the possibilities and problems associated with implementing visual feature-based cashier system for food store.

#### References:

- Gonzales R. C., and Woods R. E. (1992): Digital Image Processing, Addison-Wesley Pub. USA. p. 716.
- ICD., (2003): Industrial Camera Directory- Camera Manufacturers, Products and Specifications. Vision Systems Design, vol. 8., no. 9, pp. 20-55.
- MathSoft Corporation (1999): Help Text and Built in Functions of the Image Processing Extension Package.
- Parker, J.R. (1991): Gray Level Thresholding in Badly Illuminated Images, IEEE Trans. Pattern Anal. Machine Intell., vol. 13, no. 8, pp. 225-233.
- Perez, A., and Gonzalez, R.C. (1987): An Interactive Thresholding Algorithm for Image Segmentation., IEEE Trans. Pattern Anal. Machine Intell., vol. PAMI-9, no. 6., pp. 742-751.
- Pratt, W.K. (1978): Digital Image Processing, John Wiley and Sons, NY, USA.
- Rosenfeld, A., and Kak, A. C. (1982): Digital picture processing, 2nd ed., Academic Press, NY, USA.
- White, J.M., and Rohrer, G.D. (1983); Image Thresholding for official character Recognition and Other Applications Requiring

Character Image Extraction., IBM Journal of Research and Development, vol. 27, no. 4., pp. 400-411.

Wilson, A. (2003): Smart Cameras Embed Processor Power, Vision Systems Design, vol. 8., no. 9, pp. 95-99.

Zhoi, L., Chalana, V., and Kim, Y. (1998): PC-based Machine Vision System for Real-Time computer-Aided Potato Inspection, Journal of Imaging system Technology, vol. 9, pp. 423-433.