Automated Visual Inspection of the Packaging Process of Pills in a Blister

Rafael VALENCIA-CARREÑO and Alfredo SANTANA-DÍAZ

Instituto Tecnológico y de Estudios Superiores de Monterrey, campus Toluca.
Eduardo Monroy Cárdenas #2000, San Antonio Buenavista.
Toluca, Edo. De México CP. 50110 Tel: (722) 2799990
{a00788663, asantana }@itesm.mx

Abstract

This paper describes an image processing methodology, which is used in an automated visual inspection application. This methodology is orientated towards industrial inspection tasks that could benefit from the use of rather coarse color discrimination. The methodology is employed in a software application, in order to inspect the packaging process of pills in a blister pack –it counts the quantity of pills, packed in the blister, and estimates their roundness. The methodology involves the image acquisition –through a web camera–, spatial filtering, color segmentation through HSI color space, morphological filtering, connected-component labeling, and feature extraction. The experimental results, under controlled conditions, always were correct.

1. Introduction

Manual inspection presents many problems that could affect the product quality [5]. A viable alternative to this kind of inspection is automated visual inspection [1, 8].

In this work, we present an image processing methodology, that employs a low-cost camera, applied to automated visual inspection of manufactured goods, where coarse color discrimination could be used, so as to cut costs. The proposed methodology is utilized in a software application, in order to inspect the packaging process of pills in a blister pack.

From an image of the blister pack, the application determines the quantity of pills –specifically, it looks for green pills– and calculates approximately the roundness of every pill. The application is invariant to blister pack position; it can be in any place within the frame.

The image acquisition is achieved by a low-cost camera –web camera– and the image processing algorithms were written in C language and implemented in Visual C++.

The controlled conditions were: white illumination, fixed camera, fixed distance between camera and blister pack, and background without any object, whose color were equal to the color of the pills of interest.

2. Methodology

Figure 1 shows a block diagram that summarizes the methodology. Firstly, a frame –320x240 pixels and 24 bits deep– is acquired from the scene. The acquired image is displayed on screen. This image is filtered in the spatial domain. Then, the filtered image is segmented in order to obtain the objects of interest. Since this image may contain noise, it should be filtered. The segmented image is filtered by morphological operations. Thus, the segmented image will only contain the objects of interest. Such objects are labeled and their roundness is calculated.

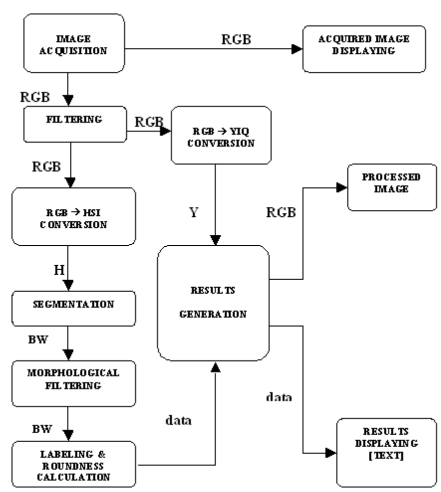

Figure 1: Block diagram.

.

The results consist of:

- A gray scale version (Y component of the YIQ color space) from the acquired image – where detected objects are surrounded by red squares.
- Number of detected pills.
- Roundness of every pill.

2.1. Spatial filtering

In this stage, smoothing spatial filters were employed to remove noise. Since these filters blur the image [7], it was necessary to find a smoothing filter that preserves better the object boundaries.

Three filtering options were tested: three passes of an arithmetic mean filter, two passes with a median filter followed by one pass of an arithmetic mean filter, and three passes of a median filter. The chosen option was the one that uses only median filters.

Every option was evaluated to know which filtering process removes noise effectively and preserves better the object boundaries.

2.2. Segmentation

The objects of interest were green pills, whose distinguishing feature is their color. Therefore, when we extract all the green objects from the image, we are segmenting every object of interest. The color of the objects can be specified by their hue. Thus, the segmentation process was performed in the HSI color space, using only the hue (H) component.

The first step in the segmentation process is to compute the hue based on the RGB components of the image. Equation (1) expresses this operation [7].

$$H = \cos^{-1} \frac{0.5[(R-G)+(R-B)]}{\sqrt{(R-G)(R-G)+(R-B)(G-B)}} \qquad (1)$$

However, it is possible to simplify the calculations to obtain the hue. Equation (2) shows the Bajon's expression [14] to compute the hue from the RGB image components. In this expression, the trigonometric functions are no longer needed.

$$\text{H} = \begin{cases} \textit{Achromatic}, & \textit{If} \quad R = G = B \\ H = \dfrac{G-B}{3(R+G-2B)}, & \textit{If} \quad B = \min(R,G,B) \\ H = \dfrac{B-R}{3(G+B-2R)} + \dfrac{1}{3}, & \textit{If} \quad R = \min(R,G,B) \\ H = \dfrac{R-G}{3(R+B-2G)} + \dfrac{2}{3}, & \textit{If} \quad G = \min(R,G,B) \end{cases} \qquad (2)$$

Because of its simplicity, equation (2) was employed to compute the hue image. The result of (2) is a hue image *H(x, y)*, whose pixels values are within the interval of 0 and 1.

Thus, the corresponding hue of the green color is, approximately, between 0.3 and 0.5. A binary image *BW(x, y)* was calculated by image thresholding with equation (3). Thus, pixels labeled 255 (white pixels) correspond to objects of interest, whereas pixels labeled 0 (black pixels) correspond to the background.

$$BW(x,y) = \begin{cases} 255, & \textit{If} \ 0.3 \le H(x,y) \le 0.5 \\ 0, & \textit{otherwise} \end{cases} \qquad (3)$$

2.3. Morphological filtering

This process cleans the binary image, computed in the segmentation stage. Mathematical morphology operations are used to improve or to make clear the shape of objects that have been discriminated in a binary image [3, 7].

The binary image was processed with morphological filters: three erosions, with a 3x3 structuring element of ones, followed by three dilations, with a 5x5 structuring element of ones.

2.4. Labeling

In order to count the segmented objects, a connected-component labeling algorithm was designed.

The aim of this algorithm is to determine if pixels belong to a specific object. The algorithm works within the vicinity of a pixel to find out which of their neighbors are connected to it [9, 12].

The algorithm takes for granted there is enough memory and it can label every object in two sweeps. This algorithm is helpful when we know the maximum number of objects in advance; thus, we can allocate just the quantity of memory we need. Dillencourt [4] shows a detailed description of how to implement an optimal connected-component labeling algorithm.

The designed algorithm assumes 4-connectivity between pixels and employs an equivalence table. Equation (4) shows an example of equivalence table. The ones express that there is equivalence between labels (e.g. label *a* and b are the same), whereas the zeros mean there is not equivalence.

$$B = \begin{array}{c} \\ a \\ b \\ c \\ d \\ e \end{array} \begin{array}{c} \begin{array}{ccccc} a & b & c & d & e \end{array} \\ \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array} \qquad (4)$$

In order to label objects, the algorithm sweeps the image by rows –from top to bottom and from left to right.

Let pixels q and z be the north and west neighbors of pixel p. The algorithm assigns a number or label to pixel p in the following way:

- If pixels q and z do not have label, a new label is generated for p.
- If only one, q or z, has label, pixel p will take that label.
- If both q and z have label, it is said that there is a label collision, and pixel p will take the shortest label. This label collision is registered in the equivalence table.

After first sweep, in the equivalence table (visualizing the equivalence sets as mathematical relations), the equivalence sets are transformed into equivalence relations (i.e. *reflexive*, *symmetric* and *transitive* relations). Thus, it is possible to join the equivalence relations in k disjoint sets or equivalence classes [10].

Therefore, when equivalence is registered in the table, the algorithm makes this equivalence symmetric. Transitive closure was employed to transform equivalence sets into transitive relations. Warshalls's algorithm [18] was implemented to perform transitive closure operation.

Finally, the algorithm makes the second sweep. This sweep is made backward –from bottom to top and from right to left–, while changing labels.

2.5. Roundness

In addition to the quantity of pills, roundness was another characteristic extracted from the image. In order to compute the roundness of objects, according to equation (6), it is necessary to know their area and perimeter.

$$roundness = 4\pi \frac{area}{perimeter^2} \qquad (6)$$

The squares that surround every object were used to estimate the area –product of the height of the square and its corresponding base.

On the other hand, a contour tracing algorithm was employed to compute the perimeter. *Moore-Neighbor tracing* algorithm [17] was implemented –assuming 4-connectivity.

This algorithm does not the visit inner perimeter, in case the object has gaps. There could be cases in which the algorithm misses pixels, especially in 4-connectivity; however, since the width of neighborhood is normally greater than a pixel, the algorithm has many chances to succeed.

3. Results

Under the established controlled conditions, the developed program always detects the correct number of pills, within the blister pack, and estimates adequately their roundness.

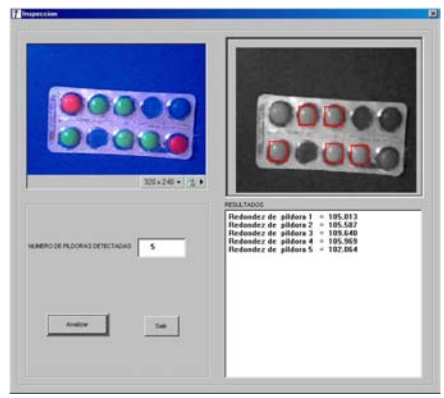

Figure 5: Application window –developed with Visual C++.

Figure 5 shows the application window, implemented with Visual C++. In the window we can observe the acquired image (left) and the processed image (right). The window also shows the quantity of pills and their roundness (text boxes).

4. Conclusions

Manual inspection presents many problems that could affect the product quality. An alternative to this kind of inspection is automated visual inspection.

In this paper was described an image processing methodology to perform automated visual inspection of products, which cuts costs by employing a low-cost camera. The proposed methodology was implemented in a software application, developed in Visual C++, in order to inspect the packaging process of pills in a blister pack. The experimental results, under controlled conditions, always were correct.

This work could be extended in many ways. One of them could be an implementation of this methodology in an embedded system or in reconfigurable hardware [2]. Another extension could involve the study of other digital filtering techniques [15] –e.g. *wavelets* filtering [16]– as well as the study of some techniques to give this methodology robustness before illumination changes. This could be achieved with a dynamic color calibration, other color spaces (e.g. CIELAB color space), or changes in color models [6, 11, 13].

5. References

[1] Batchelor, B., and P.F Whelan, *Intelligent Vision Systems for Industry*, Springer-Verlag London Ltd., UK. 2002.

[2] V. Bonato, A.K. Sanches, M.M. Fernández, M. P. Cardoso, E. D. V. Simoes and E. Marques, "A Real Time Gesture Recognition System for Mobile Robots", *ICINCO(2)*, 2004.

[3] Díaz de leon, J. L., C. Yáñez márquez, *Introducción a la morfología matemática de conjuntos*, IPN(CIC)-UNAM-FCE, México, 2003.

[4] M.B.Dillencourt, H. Samet, M. Tamminen, "A General Approach to Connected-Component Labeling for Arbitrary Image Representations", *Journal of the ACM*, Vol. 39, No. 2, April 1992.

[5] Evans, J.R. and W.M. Lindsay, *The Management and Control of Quality*, 4th Edition, South Western College Publishing, USA, 1999.

[6] G. Finlayson and G. Schaefer, "Hue that is invariant to brightness and gamma", School of Information Systems, University of East Anglia, UK, 2000.

[7] Gonzales, R.C. and R. E Woods, *Digital Image Processing*, 2nd Edition, Prentice Hall, USA, 2002.

[8] Graves, M. and B. Batchelor (Eds.), *Machine Vision for the Inspection of Natural Products*, Springer-Verlag London Ltd., UK, 2003.

[9] Haralick, R.M and L.G. Shapiro, *Computer and Robot Vision, Vols. I and II*, Addison-Wesley, Reading, MA., USA, 1993.

[10] Hoffman, K. and R. Kunze, *Linear Algebra*, 2nd edition, Prentice-Hall, USA, 1971.

[11] Y. B. Lee, B. J. You, and S. W. Lee, "A real-time Color-based Object Tracking robust to Irregular Illumination Variations", *Proceedings of the 2001 IEEE International Conference on Robotics & Automation*, Seoul, Korea 2001.

[12] Pajares, G., J. M. De La Cruz, *Visión por computador: imágenes digitales y aplicaciones*, RA-MA. España, 2001.

[13] J.B. Park, Y. Yoon, and A.C Kak, "A New Color Representation for Object Tracking under Non-white Illumination Conditions", Robot Vision Laboratory, Pardue University, USA, 2003.

[14] Sangwine, S.G. and R.E.N. Horne (Eds.), *The Colour Image Processing Handbook*, Chapman & Hall, UK, 1998.

[15] A. Santana-Diaz; N. Hernandez-Gress; D. Esteve, "Digital filtering: application on the driver's impairment detection", *Intelligent Transportation Systems Proceedings. 2000 IEEE* 1-3 Oct. 2000.

[16] A. Santana-Diaz; B. Jammes; D. Esteve; M. Gonzalez-Mendoza, "Driver hypovigilance diagnosis using wavelets and statistical analysis", *The IEEE 5th International Conference on Intelligent Transportation Systems*, 2002.

[17]. Toussaint, G., *Pattern Recognition Course Notes: Grids, connectivity and contour tracing*, McGill University Montreal, Quebec, 1997.

[18] S. Warshall, "A theorem on Boolean matrices", *Journal of the ACM*, 9(1):11-12, 1962.