FINDING SIMILARITIES BETWEEN PATTERNS USING AN OPENCV EMBEDDED IMPLEMENTATION

Alexandru-Robert SIMION, Engineer Master Student,
Adrian GLIGOR, Engineer Assistant Professor Ph.D.,
IANTOVICS Barna, Assistant Professor Ph.D.,
"Petru Maior" University of Tîrgu Mureş

Abstract: This paper is focused on exploring the efficiency of some image processing algorithms implemented on embedded systems for finding similarities between given patterns. The research was conducted using the ported OpenCV library on an embedded system. There are presented results obtained by using different algorithms combined with the Canny edge detection algorithm for efficient embedded system resources usage in finding similarities. Conclusions related to implementation on embedded systems of such algorithms are also formulated.

Keywords: image processing and analysis, image filtering, edge detection, finding similarities

1. Introduction

Many civil and industrial applications require some kind of automation level, which many times require sensors that are most related to the hardware implementation, and powerful software applications some times used for images processing as a solution to close control loops for some automation processes. Reliable and flexible software solutions require the dedicated research for development of efficent algorithms given the particularities of the problems being solved. Existing software libraries and hardware platforms offer suitable resources for many kind of applications, but impose special requirements on computing resources, portability etc. An affordable solution, nowadays increasingly used in many fields or life, may come from the embedded platforms [15]. However, this lead to the necessity to be carried out of new research regarding the optimization, development or tuning the parameters of the existing algorithms. This paper mostly focuses on finding the optimal parameters of the existing image processing algorithms, used for finding similarities between two patterns by using limited computing resources in order to create a development framework for applications used on embedded systems such as: smart devices or communication devices.

The paper is structured on three sections related to presenting the theoretical background on some image processing and analysis algorithms used in finding similarities, the work carried out on an embedded system experimental implementation, findings are discussed, and finally, the conclusions are formulated.

2. Theoretical background on finding similarities between patterns

Digital image processing technology is prevalent in many areas of science and technology, starting from well-known everyday applications used in offices or at home, medicine, automotive safety applications, reaching to space technology. Due to numerous applications using these technologies, a continuous effort is undertaken by researchers for this domain development. The image processing requires appropriate computing resources but its history begins in the year 1960 and has seen unprecedented growth since 2000. One aspect that has supported this development is due to the availability of the computing power, development of software libraries and applications.

There are many theoretical descriptions [1, 2, 3, 4] and applied researches [5, 8, 11, 12, 21] for imaging problems that require different type of image processing and analysis. One of the application areas consists in medical imaging [6, 7, 9] allowing visual decision support.

Wu, Si, Gong and Zhu [2] propose a solution for recognizing deformable templates. A template in the formulation of the authors consists in Gabor wavelet elements at selected locations and orientations. The elements may perturb their locations and orientations, after then they are combined to generate the observed image. The authors propose a novel specific learning algorithm. The proposed approach is appropriate for the vision task approached in research. For illustrative purposes the authors realized some experiments.

A specific problem of the image processing domain is the finding similarities. In this respect, according to the intended purpose were studied and proposed different solutions. Helmut Alt et al. [13] treat the problem of probabilistic matching and resemblance evaluation of shapes in trademark images. Ofir Pele and Michael Werman in [14] propose a novel algorithm for similarity detection between two images with respect to a member of a family of image distance measures named “Image Hamming Distance Family”.

Diversity of situations and problems in the field of images processing find their solving in a wide range of solutions. As a consequence it is difficult to formulate a unified handling methodology. For finding of the similarities can be adopted a combination of all or a part of the following types and algorithms of the images processing:

- filtering for noise removal and / or the preservation of the edges;
- morphological operations;
- edge detection operations;
- detecting characteristic points.

Combining such type of algorithms could lead to improvements of the result after processing.

3. Finding similarities by using embedded implementation approach

3.1. The tested embedded system

We have made the experiments using an embedded system. The tested embedded system where equipped with an ARM9 of type AT91SAM9261 microcontroller running at 180 Mhz and having 64 MbRAM. The testing application were performed on embedded Linux Ångström distribution.

3.2. Image pre-processing

Even if is necessary additional processing time, a simple method of improving the operation of edge detection can be achieved by using filters for noise reduction and edge preservation. Their use is in the images pre-processing stage, but their application is also possible after performing the edge detection. We mention some well-known algorithms such as Gaussian filtering, Median filter, Bilateral filter, etc. There are known also more complex algorithms like Gabor or Kalman filters, but the high requirements on implementation side in terms of computing resources (eg. for real-time implementations) are not considered in this paper.

In case of Gaussian-like distribution of the noise to be removed the linear low pass filter from the Gaussian class algorithms are recommended for theirs property of reducing the edge blurring, computationally efficiency, performing at the same characteristics in all directions. In practical application is recommended to perform noise estimation before being used for specific operations like edge detection [16].

Median filter algorithms are a class of popular filters that offer low computational requirements [16] and have good performances on removing additive white noise because of its low-pass filter characteristics, but also provide efficient filtering in case of noise characterized by a long-tailed distribution such as Laplacian distribution [17].

Another popular algorithm is the Bilateral filter. This is a non-linear filter mainly designed for edge-preserving and noise-reducing smoothing. The success of this filter rely on its simple formulation, dependency on reduces number of parameters (only two parameter: the size and the features to preserve), non-iterative usage [20], it can be used in soft real-time application with no very restricted time constraint. However in real-time application can be used if dedicated hardware is available (eg. graphics controller) [20].

3.3. The Canny edge detection

In many imaging problems solving, including the similarity finding between images, a step consists in edge detection. One of the most frequently used edge detector algorithm is the Canny edge detector. The Canny edge detector, proposed by Canny [10] uses a multi-stage algorithm to detect edges in images. It is an algorithm with a general purpose that could be used for a large variety of images. The Canny algorithm is based on a calculus of variations in order to optimize a functional. The aims of this algorithm development, formulated by Canny were: good detection (marking as many edges in the image as possible); good localization (edges marked should be as close as possible to the edge in the image); minimal response (a given edge in the image should only be marked once).

Canny edge detector algorithm presents the main steps of the Canny algorithm applied for edge detection.

Canny Edge Detector Algorithm

Input: IN (input image)

Output: OUT (output image)

Step 1

@Establishment of the parameters of the algorithm

Step 2

@Filter out the existent noise from the image using the Gaussian filter.

Step 3

@Find the intensity gradient of the image, procedure analogous to Sobel.

Step 4

@*Applying of non-maximum* suppression. This removes pixels that are not considered to be part of an edge. Hence, only thin lines (candidate edges) will remain.

Step 5

@*Hysteresis* using an upper and lower threshold.

```
For (each pixel from the image except the least lines from the image) execute
    If (a pixel gradient is higher than the upper threshold) then
        @the pixel is accepted as an edge.
    EndIf
    If (a pixel gradient value is below the lower threshold) then
        @the pixel is rejected as an edge.
    EndIf
    If (((the pixel gradient is higher then the lower threshold) and (the pixel gradient is
    lower then the upper threshold)) and (the pixel is connected with another pixel that is
    above the upper threshold)) Then
        @the pixel is accepted as an edge.
    EndIF
Endfor

EndCannyEdgeDetectorAlgorithm
```

Step 1 consists in the establishment of the parameters of the algorithm. The variable parameters during the experiments where: convolution matrix dimension by NxN, where N=2xK+1, we have made experiments with K=1 (N=3) and K=2 (N=5); *upper* and *lower* threshold (with value of 150 and 255).

Step 3 consists in find the intensity gradient of the image based on a procedure analogous to Sobel that consists in two steps

Step 3. A) Apply a pair of convolution masks (in two directions namely X and Y):

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix}$$

$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix}$$

Step 3. B) Find the gradient strength and direction (the direction is rounded to one of the following possible angles 0, 45, 90 or 135 with:

$$G = \sqrt{G_x^2 + G_y^2}$$

$$\theta = \arctan\left(\frac{G_y}{G_x}\right)$$

Step 4 of the algorithm consists in the applying of *non-maximum* suppression. This has as effect the removing of pixels that are not considered to be part of an edge, which makes the lines thinner.

Figure 1 presents the result obtained applying the algorithm Canny edge detector. The parameters of the experiment where: convolution matrix is of size 3x3, lower threshold = 150 and upper threshold = 255.

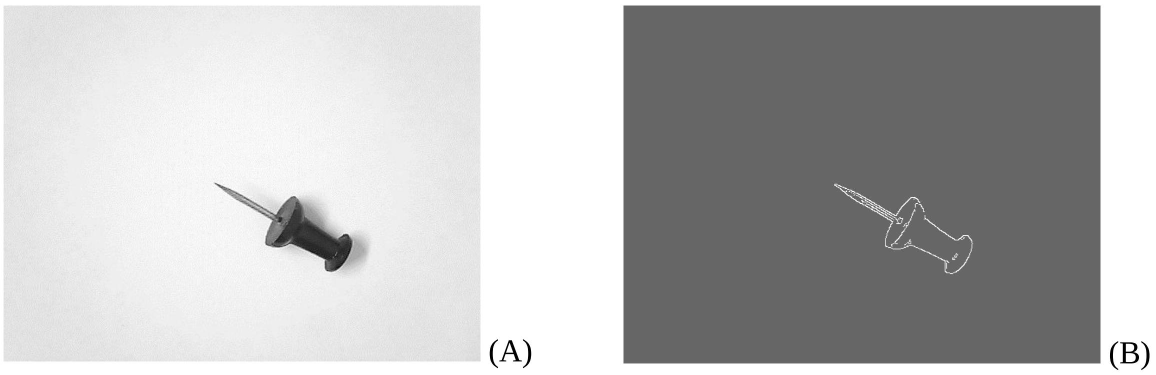


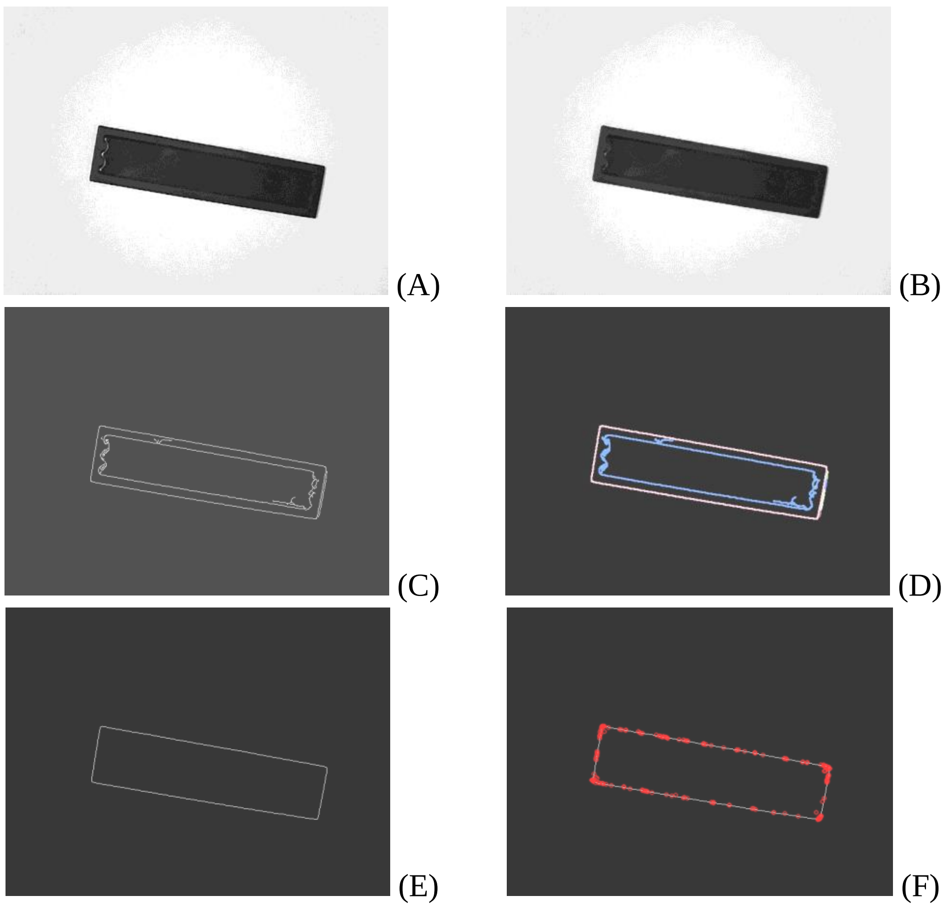
Fig. 1. *An example for the processing realized by the „CannyEdgeDetector” algorithm; A) Input test image; B) Output image (binarized image)*

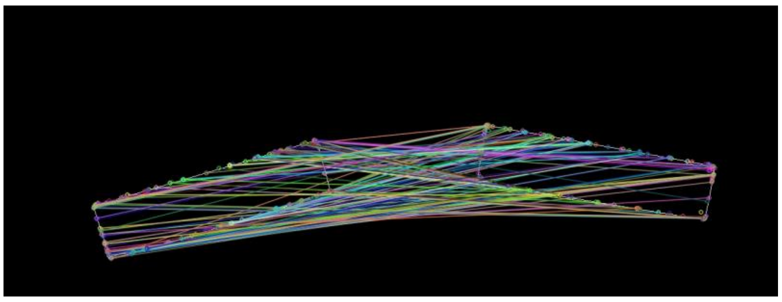
3.4. Finding similarities

Basically the similarities finding can be obtained from two computational stages:

- Feature detection
- Descriptor extraction

Figure 2 presents the results obtained applying the proposed similarities finding algorithm with the following parameters: convolution matrix by size 5x5, lower threshold = 150 and upper threshold = 255.




(G)

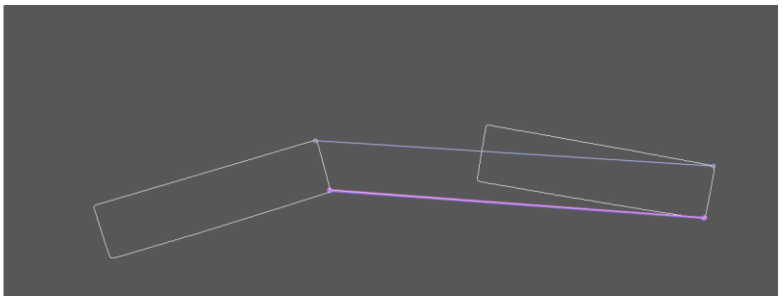

(H)

Fig. 2. *Example for the processing realized by the similarities finding algorithm for a polygonal shape. A) Input test image; B) Edges detection; C) Contours extraction; D) Characteristics points detection; E) Exterior contours extraction; F) Characteristics points comparison.*

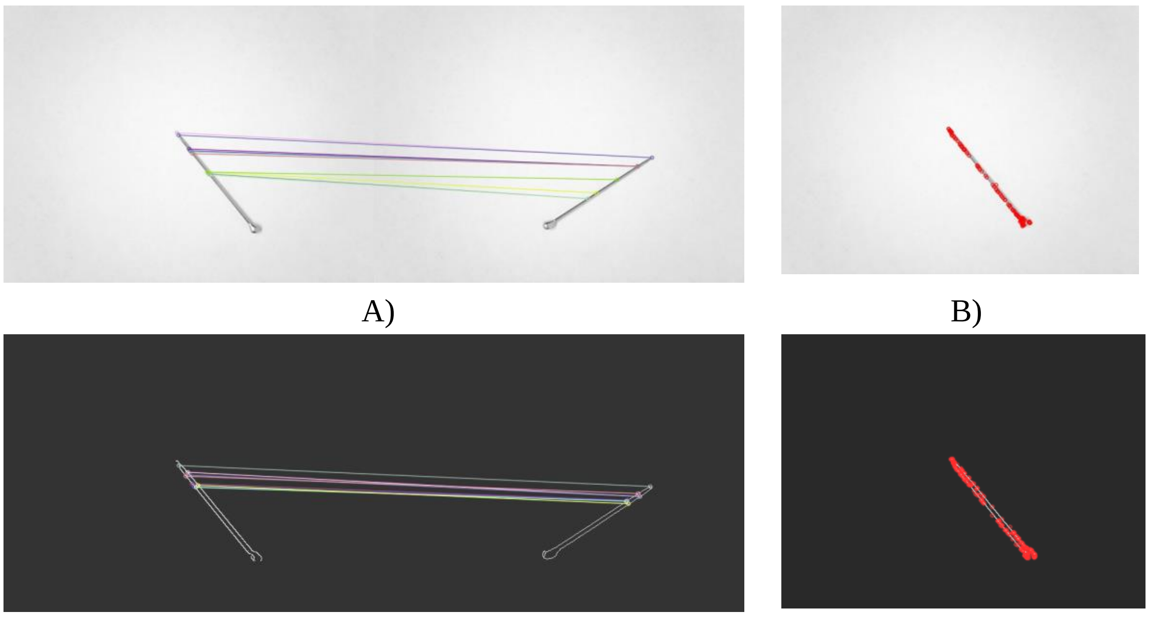

A) B)
C) D)

Fig. 3. *Example for the processing realized by the similarities finding algorithm for a thin shape. A) Filtered feature comparison on original image; B) Features from original image; C) Filtered feature comparison on exterior contour; D) Features from exterior contour.*

In Fig. 4 are synthesized the mains performances of the most relevant analyzed algorithms. The results presented in Fig 4 are associated for Filter 1 to Box filter with kernel size 3x3 and 5x5, Filter 2 to Median filter kernel size 5x5, Filter 3 to Bilateral filter with kernel size 3x3 sigma color=255 sigma space=255 and Filter 4 correspond to Bilateral kernel size 5x5 sigma color=255 sigma space=255.

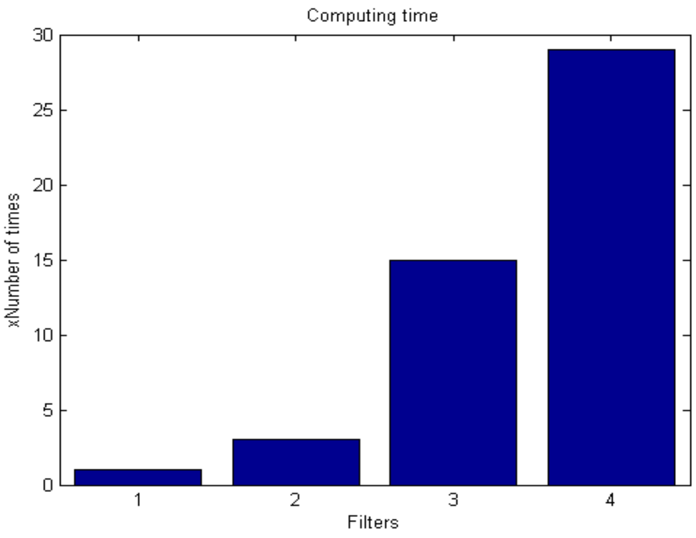


Fig. 4. *Computing performances by using OpenCV library.*

4. Conclusions

The main conclusions related with the similarity finding algorithms implemented on the tested embedded system development board assure a suitable time for software real-time implementations. The performances of the implemented algorithms depend on chosen filters and the order of the application. Some filters offer improvements for a class of applications but in case of similarity finding the results are affected in a negative way. This is a case of morphological operations where similarity obtained is at low level.

The main conclusions related with the Canny edge detection algorithm implementation based on OpenCV library for embedded system assure low error rate related with the detection of edges; good localization, minimized distance between edge pixels detected and real edge pixels.

The future research direction will consists in improvement in efficiency related with the accuracy of edge detection, similarity finding and more improvement of the computational time.

Evolutionary algorithms are used for many computational hard problems solving [18, 19], that doesn't have a more appropriate solution. Another prospective aspect approach in the research will consists in finding of the optimal parameters using an evolutionary algorithm.

References

[1] Umbaugh, Scott E., *Digital image processing and analysis: human and computer vision applications with CVIPtools* (2nd ed. ed.). Boca Raton, FL: CRC Pres, 2010.

[2] Zhang, W., Bergholm, F., "Multi-scale blur estimation and edge type classification for scene analysis", International Journal of Computer Vision, vol 24, issue 3, Pages: 219 – 250, 1997.

[3] Ziou, D., Tabbone, S., "Edge detection techniques: An overview", *International Journal of Pattern Recognition and Image Analysis,* 8(4):537–559, 1998

[4] Lindeberg, T., "Edge detection and ridge detection with automatic scale selection", *International Journal of Computer Vision*, 30(2):117-154, 1998 http://www.nada.kth.se/-cvap/abstracts/cvap191.html

[5] Wu, Ying Nian, Si, Zhangzhang, Gong, Haifeng, Zhu, Song-Chun, Learning Active Basis Model for Object Detection and Recognition, Int. J. Comput. Vis., 90: 198–235, 2010.

[6] Linguraru, M.G. Richbourg, W.J. Liu, J. Watt, J.M. Pamulapati, V. Wang, S. Summers, R.M., Tumor Burden Analysis from CT Data of Diseased Patients via Automated Liver and Tumor Segmentation. IEEE Transactions on Medical Imaging, Vol. 31(10):1965-7 2012. https://sites.google.com/site/mglinguraru/publications

[7] Markonis D., Holzer M., Dungs S., Vargas A., Langs G., Kriewel S., Müller A., “A survey on visual information search behavior and requirements of radiologists”, *Methods of information in Medicine*, 51(6):539-548, 2012.

[8] Foncubierta A., Müller H., Depeursinge A., “Retrieval of High-Dimensional Visual Data: current state, trends and challenges ahead”, *Multimedia Tools and Applications special issue,* 2013.

[9] Hang Su, Zhaozheng Yin, Seungil Huh, Takeo Kanade, “Cell segmentation in phase contrast microscopy images via semi-supervised classification over optics-related features”, *Medical Image Analysis*, 17: 766–778, 2013.

[10] Canny, J., “A Computational Approach To Edge Detection”, *IEEE Trans. Pattern Analysis and Machine Intelligence,* 8(6):679–698, 1986.

[11] Deriche R., Using Canny's criteria to derive a recursively implemented optimal edge detector, Int. J. Computer Vision, 1, 167–187, April 1987

[12] Bardosi, F.I., Grif, St.-H., Ratoi, O., “Mouse Controlled by Hand Gestures Using Digital Image Processing ”, *In Proc. of. The 4th International Conference on Interdisciplinarity in Engineering INTER-ENG 2009,* 2009; 163-167

[13] Alt, H., Scharf, Ludmila, Scholz, S., “Probabilistic matching and resemblance evaluation of shapes in trademark images”. *In Proc. of CIVR '07 Proceedings of the 6th ACM international conference on Image and video retrieval,* 2007, 533-540.

[14] Pele, O., Werman M., “Robust Real Time Pattern Matching using Bayesian Sequential Hypothesis Testing”. *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* 2008, 30(8): 1427 - 1443.

[15] Kamal, R., Embedded Systems - Architecture, Programming and Design. Tata McGraw-Hill, 2008.

[16] Nguyen, T.-A., Hong, M.-C., “Filtering-based Noise Estimation for Denoising the Image Degraded by Gaussian Noise”, Advances in Image and Video Technology: 5th Pacific Rim Symposium, PSIVT 2011, Ed. Y.-S. Ho, Part II. LNCS 7088, Springer-Verlag Heidelberg, 2001, 157-167.

[17] Pitas, I., Digital Image Processing Algorithms and Applications. John Wiley & Sons, 2000, 139-140

[18] Back, T, Fogel, D. B. and Michalevitz Z., Handbook of Evolutionary Computation, Oxford University Press, Oxford, 1997

[19] Dumitrescu, D., Lazzerini, B., Jain, L. and Dumitrescu, A., Evolutionary Computing, CRC Press, Boca Raton, 2000.

[20] Paris, S., Kornprobst, P., Tumblin, J. and Durand, F., “Bilateral Filtering: Theory and Application”, Foundationd and Trends in Computer Graphics and Vision, 2008, 4(1):1-73.

[21] German-Sallo, Zoltan. Filter bank synthesis for adapted wavelet analysis, Acta technica Napocensis, 51(3), 2010, 63-66.