RESEARCH ARTICLE OPEN ACCESS

Approach To Design and Develop Highly Efficient Face Detection System

Sonia Chhetri
P.G Scholar, Department of Computer Science and Engineering,
Uttarakhand Technical University,
Dehradun- India

ABSTRACT

In the field of biometric and video coding, face detection and tracking has been an important and active research area. We have proposed a system to implement a real time face detection system to detect and track the human face. Image filtering and color based skin segmentation will be used in our proposed face detection algorithm. We would be working on various methods to determine the location of face in an image, one way can be by calculating the centroid of the detected region. We would be implementing the software version of this system in MATLAB and will test the system on still pictures. By this approach, an attempt is made to improve the accuracy and effectiveness of the real time system under various facial poses and, skin colors and varying light conditions. The application will be tested in real time with minimum computation effort, hence power limited applications may find our proposed approach very useful and efficient.

Keywords:- MATLAB, Image processing, Color segmentation, morphological Operation, face detection

I. INTRODUCTION

Face detection and tracking is the process of determining whether or not a face is present in an image. Unlike face recognition—which distinguishes different human faces, face detection only indicates whether or not a face is present in an image. In addition, face tracking determines the exact location of the face. Face detection and tracking has been an active research area for a long time because it is the initial important step in many different applications, such as video surveillance, face recognition, image enhancement, video coding, and energy conservation. The applicability of face detection in energy conservation is not as obvious as in other applications. However, it is interesting to learn how a face detection and tracking system allows power and energy to be saved. Suppose one is watching a television and working on other tasks simultaneously, the face detection system check whether or not the person is looking directly at the TV. If the person is not directly looking at the TV within some time period (i.e. 15 minutes), the TV's brightness is reduced to save energy. When the person turns back to look at the TV, the TV's brightness can be increased back to original. In addition, if the person looks away for too long (i.e. more than one hour), then the TV will be automatically turned off.

Different approaches to detect and track human faces—including feature-based, appearance-based, and color-based have been actively researched and published in literature. The feature-based approach detects a human's face based on human facial features—such as eyes and nose. Because of its complexity, this method requires lots of computing and memory resources. Although compared to other methods this one gives higher accuracy rate, it is not suitable for power-limited devices. Hence, a color-based algorithm is more reasonable for applications that require low computational effort. In general, each method has its own advantages and disadvantages. More complex algorithm typically gives very high accuracy rate but also requires lots of computing resources.

For face detection techniques two broad categories are identified that separate the various approaches, appropriately named feature-based and image-based approaches. Each category will be explained, and the work completed will be presented, providing a brief yet thorough overview of the various face detection techniques.

The rest of the paper is organized as: Section II describes the different feature based approaches viz. low-level analysis, feature analysis and active shape model. Section III covers the design and implementation which proposes two algorithms for face detection. . Section IV concludes the paper.

II. FEATURE BASED APPROACH

Here, we are trying to define an approach to locate the face in an image. Feature-based systems are divided into three subcategories: low-level analysis, feature analysis, and active shape models.

A. Low-Level Analysis

Low-level Analysis deals with the segmentation of visual features using the various properties of the pixels, predominantly gray-scale or color. Edge representation (detecting changes in pixel properties) was first implemented by *Sakai et al [6]* for detecting facial features in line drawings. Gray Information can be used to identify various facial features. Generally Eyebrows, pupils and lips appear darker than surrounding regions, and thus extraction algorithms can search for local minima. In contrast, local maxima can be used to indicate the bright facial spots such as nose tips. Detection is then performed using low-level GRAY-scale thresholding. Motion Information (where available) can be used to assist in the detection of human faces, using the principle that, if using a fixed camera, the “background clutter” will remain somewhat static, relative any “moving object”. A straightforward way to achieve motion segmentation is by frame difference analysis. Thresholding accumulated frame differences is used to detect faces or facial features. Another way to measure motion is through the estimation of moving image contours, a technique that has proven to be more reliable, particularly when the motion is insignificant.

B. Feature Analysis

Low-level analysis introduces ambiguity which can be solved by high level feature analysis, often through the use of some additional knowledge about the face. There are two approaches for the application of this additional knowledge (commonly face geometry). The first involves sequential feature searching strategies based on the relative positioning of individual facial features. Initially prominent facial features are determined which allows less prominent features to be hypothesized. The facial feature extraction algorithm is a good example of feature searching, achieving 82% accuracy with invariance to gray and color information, failing to detect faces with glasses and hair covering the forehead.

C. Active Shape Model

Active shape models represent the actual physical and hence higher-level appearance of features. These models are released near to a feature, such that they interact with the local image, deforming to take the shape of the feature.

III. DESIGN AND IMPLEMENTATION

A. First Algorithm

General design stages are illustrated in Figure 1.

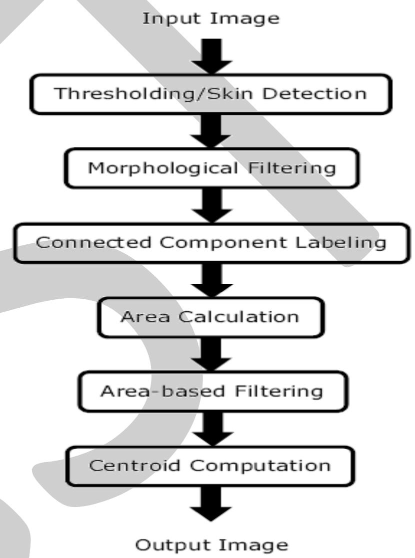

Fig. 1 Connected Component Labeling and Area Calculation

Color segmentation has been proved to be an effective method to detect face regions due to its low computational requirements and ease of implementation. Compared to the featured based method, the color-based algorithm required very little training.

1) *Thresholding/Skin Detection:* Skin pixels will be converted to the modified YUV space[10], the skin pixels can be segmented based on the experimented threshold values which will be calculated during experimentation.

2) *Morphological Filtering*: Realistically, there are so many other objects that have color similar to the skin color. There are lots of false positives present in the raw segmentation result. Applying morphological filtering—including erosion and hole filling would, firstly, reduce the background noise and, secondly, fill in missing pixels of the detected face regions, MATLAB provided built-in functions—imerode and imfill for these two operations[11][12].

```
outp = imerode(inp, strel('square', 3));
```

The command imerode erodes the input image inp using a square of size 3 as a structuring element and returns the eroded image outp. This operation removed any group of pixels that had size smaller than the structuring element's.

```
outp = imfill(inp, 'holes');
```

The command imfill fills holes in the binary input image inp and produces the output image outp. Applying this operation allowed the missing pixels of the detected face regions to be filled in. Thus, it made each face region appear as one connected region.

After each group of detected pixels became one connected region, connected component labeling algorithm will be applied. This process labels each connected region with a number, allowing us to distinguish between different detected regions. The built-in function bwlabel for this operation was available in MATLAB. In general, there are two main methods to label connected regions in a binary image—known as recursive and sequential algorithms.

```
[L, n] = bwlabel(inp);
```

The command bwlabel labels connected components in the input image inp and returns a matrix L of the same size as inp. L contains labels for all connected regions in inp. N contains the number of connected objects found in inp. The command regionprops can be used to extract different properties, including area and centroid, of each labeled region in the label matrix obtained from bwlabel.

```
face_region = regionprops(L, 'Area');
face_area = [face_region.Area];
```

The two commands above performed two tasks (1) extract the area information of each labeled region (2) store the areas of all the labeled regions in the array face_area in the order of their labels.

3)*Area-based Filtering:* Note that morphological filtering only removed some background noise, but not all. Filtering detected regions based on their areas would successfully remove all background noise and any skin region that was not likely to be a face. This can be done based on the assumption that human faces are of similar size and have largest area compared to other skin regions, especially the hands. Therefore, to be considered a face region, a connected group of skin pixels need to have an area of at least 26% of the largest area. This number was obtained from experiments on training images. Therefore, many regions of false positives could be removed in this stage.

4) *Centroid Computation:* The final stage is to determine face location. The centroid of each connected labeled face region can be calculated by averaging the sum of X coordinates and Y coordinates separately. The centroid of each face region can be denoted by the blue asterisk. Here the centroid of each connected region can be extracted using" regionprops".

5) *Image Enhancement:* Under hardware constraints, a simple non-linear and hardware-friendly sharpening filter that proved its efficiency will be implemented and employed in YUV color space. The filter properties will be dynamic in nature and can be adjusted from time to time as per the requirement of the system. For example during low light conditions the filter need to be adjusted to process dull images, whereas, during proper lightening conditions the filter settings may be altered to get the desired results.

B. Second Algorithm

The second algorithm is based on template matching system. A template is chosen for reference and that is then compared with the given input images. Certain steps are carried out as shown below which decide whether the given input images are same as the template or not.

- Template data
- Down-sample the image by factor 2:1
- Calculate the face-probability image by color
- Remove hands/arms
- Template match with skin-probability image
- Eliminate false positive hits on necks of large faces
- Remove detections at the edge of the image
- Remove patterned hits

1) Template data**:** The template data matrix for the skin/not skin color-space separation will be pre-computed and stored in file, which is then loaded at the beginning of each classification execution. This data is used to calculate the skin- probability image. The template data will consist of many scaled versions of the two basic templates along with their associated thresholds.

2) Down-sample** the image by 2:1[13]:** As the image is extremely high resolution the first thing we do to it is to down-sample it to half the horizontal and vertical size by simply throwing out every other sample. We may not use an anti-aliasing filter, as we are not concerned with the fine detail of the image; the resolution is sufficiently high that we do not see any degradation in our outputs from this approach. Yet, by decreasing the number of pixels by a factor of four, the computational time for subsequent operations can be decreased greatly.

3) Calculate the face-probability image by color: This technique will do a table lookup on the pre-computed and pre-loaded conditional probability of a pixel being a face pixel given its color.

4) Remove hands/arms**:** The skin-probability image is thresholded and smoothed with a median filter to generate a binary image of skin blobs. Each of the blobs below a certain threshold in the vertical dimension will be examined and removed if it is either too small (a hand) or has too high of an eccentricity value (an arm).

5) Blob detection through template matching: The actual implementation simply consists of taking the correlation with the original image of a given template, finding the peaks above that template's threshold, marking those as faces, and subtracting the template from the remaining face-probability image to prevent future detections of that particular face.

6) Remove no-neck heads below neck-heads**:** In general the faces in the bottom row (the ones closest to the camera) are larger than those in the back. If these faces are rotated they will not match our template and the larger neck templates leave a large piece of the original rotated image behind. This is often detected incorrectly in subsequent iterations as a face by the smaller no-neck templates. As all the faces towards the bottom of the image are assumed to be closer to the camera we can be quite certain that there should be no small, obscured faces in this region, and we will remove any false hits of this type.

7) Remove edge points: This step simply removes any points within 4% of the edge of the image.

8) Remove patterned hits: This step is necessary to remove the tile area on the roof seen in several of the images. This area is both large and has a very face-like color, resulting in several strong hits with many of the medium-sized templates. This may help us in improving overall efficiency.

IV. CONCLUSION

As discussed in this paper, two algorithms are proposed for face detection. The first one is based on color segmentation and the other one is on template matching method. However, both of them may be implemented in real time and will be checked for correctness and efficiency. Additionally, other face detection approach has been also discussed in this paper. It depends on the functional requirements that which one is appropriate and useful. In end by using these algorithms a proposed highly efficient and fast face detection system can be implemented in MATLAB.

REFERENCES

[1] I.Craw, H.Ellis, and J.R. Lishman,, *Automatic extraction* of *face features,* Pattern Recognition Letters, 5:183-187, February.1987.

[2] R. Hoogenboom and M. Lew, *Face detection using local maxima*, IEEE Proc. of 2nd Int. Conf. on Automatic face and gesture recognition, 334-339,1996.

[3] M. Ooi, *Hardware Implementation for Face Detection on ...* Component Transformation Color Space, in Third IEEE International Workshop on Electronic Design, Test and Applications, Washington, DC, 2006.

[5] S. Paschalakis and M. Bober, *A Low Cost FPGA System for High Speed Face Detection and Tracking*, in Proc. IEEE International Conference on Field Programmable Technology, Tokyo, Japan, 2003.

[6] M. Sakai, Nagao, and T. Kanade, *Computer analysis* and *classification* of *photographs* of *human faces*, in Proc. Of First USA—Japan Computer Conference, pp. 2.7, 1972.

[7] J.C Terrillion, M. Shirazi, H. Fukamachi, and S. Akamatsu, *Comparative performance of different skin chrominance models and chrominance spaces for the automatic detection of human faces in colour images*, Proceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition, 2000.

[8]V. Govindaraju, *Locating human faces* in *photographs*, The International Journal of Computer Vision, Kluwer Academic Publishers, 19(2): 129-146, 1996.

[9] G. Yang and T.S. Huang, *Human Face Detection* in *Complex Background*, Pattern Recognition, vol. 27, no. 1, pp. 53-63, 1994.

[10] About YUV Space retrieved from Princeton University Tutorial http://www.princeton.edu/~achaney/tmve/wiki100k/docs/YUV.html

[11]Tutorials on morphological operations taken from http://www.mathworks.in/help/images/morphological-filtering.html

[12]Tutorials from CYUT University http://www.cyut.edu.tw/~yltang/course/image%20 processing/notes-ip09.pdf

Sonia Chhetri completed her B.Tech(IT) from Uttarakhand Technical University, Dehradun in the year 2011. Currently, she is doing her M.Tech(CSE) from the Tula's Institute, Dehradun. Her research interests include Artificial Intelligence, Image Processing, Wireless communication,wired communication.