Edge-based Crowd Detection from Single Image Datasets

Mike Wu, *Undergraduate, Yale University,* and Madhu Krishnan, *Undergraduate, UCSD*

***Abstract*—This paper describes the design of a crowd-based facial detection and recognition system using only optical features, allowing for robustness in tracking characterizations with applications in security and data extraction. Implementation is divided into three parts: packing information regarding a given image into edge pixels, segmentation into object groups, and circular segmentation. Detection is achieved by filtering the circles and characterizing those with features similar to that of a normal face. Preliminary facial recognition is described by matching feature vectors to each "facial region" and matching over subsequence image frames. Algorithms were implemented in MATLAB and testing was performed with a low-resolution video camera. Through a number of trials, results show good detection and tracking abilities given small to medium crowd sizes. Several limitations will be addressed.**

Keywords*—*Computer Vision, Sobel, Facial Detection, Ellipses

I. Introduction

Facial recognition is a growing field of computer vision. Closed source algorithms, such as Facebook's facial recognition system, are generally effective. However, because they are proprietary software, such software is, for the most part, unavailable to the public or faces a cost problem. Current research into open source algorithms is based on two schools of thought: a feature based approach or Gestalt based recognition, which takes a more holistic approach, given that a face is only recognizable as a whole, and not as individual elements. Feature based approaches have been dwelled into, but have many common problems. Many of the popular facial recognition systems are based on ellipse/circle detection algorithms, such as Hough Transform, and rely on eye-detection software. However, this focus on specific features is not robust enough to account for all scenarios of face-detection. As an example, a side view of the face often leaves the eyes too contiguous with the face to properly segment in all scenarios. Additional problems result from the sheer volume of information present in specific images. Given a large crowd, the resolution of each individual is blurry and most of the features are difficult to pinpoint. Thus, the feature-based approach needs to specialize the image based on outside specifications, defeating the purpose of an autonomous system.

A. Viola-Jones Detection

The most common type of Gestalt system is based on the Viola-Jones object detection framework, which is a rapid-calculation method based on Haar-objects. This method is generally effective, and as such this paper proposes another Gestalt system of facial recognition by subsection categorization using the summed differences of n by n square objects instead of the traditional Harr. This system takes a proposed image and uses a modified gradient operator thereby differentiating the image from the surroundings. These pixels are then enhanced to accentuate the outside of the human face, which can then be detected using a robust ellipse-generation algorithm. This system is not unique in its approach to facial recognition, but rather improves upon the methodology of choosing the pixels as well as optimizing the ellipse that will encompass the face.

M. Wu is a student under the Department of Computer Science, Yale University, New Haven, CT 06520 USA site: http://www.mikewumike.com.
M. Krishnan is a student under the Department of Computer Science, University of California, San Diego, CA 92130 USA e-mail: mvkrishn@ucsd.edu.
Manuscript received April 19, 2005; revised January 11, 2007.

II. Literature Survey

A. Simple and Fast Detection

Anila and Deverajan proposed a simple and fast edge-based face detection system. Pre-processing was done with a median filter for noise removal and histogram equalization for contrast adjustment. Edges were extracted with a sobel operator, creating rectangular objects that could be inputted into a back-propagation neural network. The system was capable of detecting single objects in environment of relatively little noise. While the method includes a well-developed pre-processing setup, little is done in terms of constraints for edge detection.

B. Still Image Crowd Detection

Arandjelovic's solution tackled crowd detection from still images using traffic management data. The system was situated for repetitive crowd features and uses SIFT features to handle detection. While more accurate than standardized sobel, SIFTing is an expensive procedure and shifts the focus of the problem to differentiating between background and objects-of-interest since so many potential objects are found. On top of that, a support vector machine was used to track classified objects, providing a robust but computationally heavy model.

C. High Resolution Detection

Mustafah et. al. developed a real-time face detection and tracking system using high resolution cameras. This is particularly interesting because the larger the crowd size, the higher resolution needed to differentiate faces. The "smart camera system" developed used simple rectangular haar objects with Adaboost to train the data set. Unfortunately using such a high resolution created more false positives and required much higher memory usage. While the memory usage is not unique, a more suitable algorithm may prevent a large increase in false positives. Interesting enough, this system also includes background subtraction and a skin contour detection, both of which narrow down the possible space of relevant objects early on.

The purpose of this paper is to delve deeper into the field of edge detection rather than the optimization of the process surrounding it. Additions such as HD footage, pre-processing, and statistical learning could still be used in a similar manner.

III. Individual Algorithms

A. Modified Edge Operator

Given the standard operators (Robert, Sobel, etc), the primary concern is to locate the most relevant edges to the objects in the image frame, and those given operators do a good job finding outlines of an image. However, once those edges are found, very little information is stored in the binary edge image. It then becomes very difficult to group edges and segment objects in the foreground effectively.

Instead, the modified edge operator focuses on locating edges in addition to storing local context to differentiate different parts of a continuous edge.Given each pixel P in the image frame, create a window around P of size N (this size is variable given a tradeoff between speed and accuracy). This window will be considered to sufficiently characterize the local image scene. Subtracting each of P's neighbors in the window of size N from pixel P and averaging the values over $N^2 - 1$ defines the delta value of P. Similar to the pixel-intensity sums defined by Harr objects, the delta value becomes a key definition of pixel P.

Given a p by q sized image, the delta mask created from calculating the delta value at each mathematically possible location will be of size p-n by q-n.Notice that increasing N captures more global context but reduces the emphasis of pixel P's initial value (losing local individuality). Plotting the delta mask as a heat map, the edges will by definition be the most prominent pixels seeing that their local window should possess the largest fluctuation. However, in addition separating edges from the remaining pixels, edges are uniquely identified: those with similar contexts will be similar but many intersection edges may have different local contexts, thereby allowing the delta value to be the distinguishing factor between them. If this holds true, then the comparison of delta values should isolate edges belonging to an area of the same local context, meaning an object.

- similar starting layout to convolution - neighbors of a pixel is defined as all the pixels encompassed with the window of size P

It's interesting to characterize the uses of the delta operator in both a single object instance and the multi-object scene. Given many objects in the image frame, the delta operator is a good estimator of segmentation since it is very likely that edges of different objects are defined by different local contexts. In a single object scene, the delta operator is useful in breaking the object into different components, allowing us to characterize the features of that object more accurately.

B. Thresholded Exponent Segmentation

Before facial detection is possible, the edges must be grouped in an appropriate manner that makes it easy to characterize similar but separate objects of interest: faces. The delta operator is very useful in picking out the "most interesting" pixels of an image frame (easily done by a sorting the delta mask and taking the top X pixels - these usually define object edges since they contain the largest contrast). But how are objects defined? The delta operator is designed with the assumption that edges of one object have the most similar local context, and this fact can be exploited to separate objects.

Optimally, plotting all the delta values with their appropriate locations in the image frame would give us a two dimensional contour plot with local extremas dictating individual objects (much like how color histogram separation works). However, there are difficulties in separating objects in the foreground because any edges located in the background will be significantly different than those defining the objects. Therefore, there is a tendency for the delta operator to always detect two objects, foreground and background. That is not nearly as useful as isolating objects for the purpose of this paper. It is then important to accentuate the smaller differences in the foreground with exponentiation.

Removal of the background is performed along with object segmentation. The difficulty is that the two local extremas separating foreground and background are never perfectly distinct and often take place in the form of multiple smaller peaks. It is not possible to determine ahead of time accurately which of the multiple peaks contain our pixels of interest, so all must be taken into consideration. Given the matrix of deltas, we can sort the values and organize them given intensity (merely grouping the curve segments into pieces of joined segments of peaks and troughs). After interpolation and smoothing, all the trial images were constrained to ¡ 6 peak/trough sections.

*The peak/trough sections were created from the zeroes of the interpolated and smoothed curve.

This curve is broken down into [inflection point, local max/min, inflection point]. Given these boundaries, each pixel in the delta array is tagged with its proper group.

From each of those groups, the "important pixels" are picked out. But because each of the pixels are only slightly different from each other, it is desirable to exaggerate the differences. Performing $e^{(1/x)}$ M iterations for the delta array in each section with x being a single pixel's value, a seemingly continuous curve is broken down into more peaks and troughs. Plotting each of these chosen pixels from each section, most if not all represented an object of interest or part of one. Using hierarchical clustering, the similar objects are grouped.

C. Robust Ellipse Detection

After segmentation, objects are grouped by themselves to some degree but are merely composed of individual pixels. To detect faces is a more difficult task.

In this paper, we chose to implement a similar ellipsis algorithm, but because each of the segmented sections is combined with a limited amount of pixels, there is leeway for a more computationally expensive solution. Given each section, every possible 4 points defines a pair of major and minor axis, thereby creating an ellipse. Although there are fewer pixels, this is still a costly procedure and contributes to most of the processing time.

Given all ellipses, it is important then to locate the ones associated with faces. This paper proposes a two-staged implementation to sort the ellipses in order of relevance.

Stage One: Each ellipse is given a feature vector, defining a number of attributes, such as the ratio of major axis to minor axis, the number of pixels outlining (or close to) the ellipse, the general ellipse size in relation to the average, etc. These different features were weighted and the weighted sum represented a categorized ”score” for the ellipse. Naturally, these attributes were chosen to conform to those representing a human head. Picking the top M percent, the remaining ellipses all identified very face-like structures, although many objects can be disguised as faces given this criteria. Possible improvements include much more work with the attributes in the feature vector.

Stage Two: Given the trimmed set, the usual case depicts many of the circles overlapping. To reduce redundancy, a blobbing algorithm is used to merge similar circles. Regions with overlapping circles are prioritized to regions with one circle. More than not, the face in crowds are encompassed within one of these finalized circles.

IV. Feature Recognition

The work proposed above is interesting for recognition because given a face, the delta operator gives a lot of information that is unique to that particular face (with exception of lighting). There is a lot of detail around the nose and eye regions depicting not only relative position, but relative local delta maximas that are stable between frames as well. We suspect that after all the objects are segmented and facial objects detected, they could be once again be recognizable by image factors, one of which being the delta value.

It is also apparent that all of the image processing presented above was performed given one frame at a time. Given a series of image frames, much of the noise and false detections may be eliminated through repeated exposure of the facial objects. Future work may look at SIFT-ing found objects through frames and incorporating a support vector machine to decide what is worth showing as a facial object and what isn’t based on cross-frame data.

Combining the two ideas presented in this section, given video footage and feature recognition, it might also be possible to create a self-updating 3D representation of each found face based on accentuated delta matrices.

V. Future Work

Pre-processing was neglected since the work was a proof of edge detection. However, the addition of background subtraction, skin contour differentiation, etc to generate multiple fine-tuned regions of interest would greatly benefit the amount of processing needed.

In addition, because the following algorithm is for single image frame data sets, it would be interesting to see the improvement in accuracy given a series of image frames, allowing for the tracking of features between frames.

VI. Sample Data

Below represents detailed graphical walkthrough of a trial image frame. This image frame was chosen because of its large variance in color and large amounts of potential objects.

Further testing showed comparable capabilities. Often ”facial objects” were found that did not correspond to faces but to some similar oval-structured object. More trials are to be performed to fully gauge the robustness of the algorithm, for example, its effectiveness at night or its accuracy in footage.

Figure 1. This image is chosen to serve as an example due to the large amount of noise in the background and abundance of potential facial objects in the scene.

A. Step One: Applying the operator

Using the modified edge operator, categorize each pixel in the image in terms of its neighbors. It is an interesting problem to choosing the best N for the operator. This should dependent on the resolution of the camera as well as the distance away from the crowd of interest. The lower the resolution and the closer the crowd is, the larger N should be.

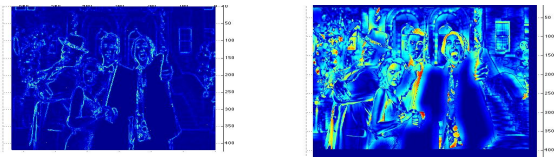

Figure 2. The left image shows the modified sobel operator with a N of size 3. The right image shows a congruent operator with N of size 25. Notice that increasing N increases the general magnitude of brightness in pixels and blurs (”spreads out”) the regions of interest.

B. Step Two: Pulling out the important points

This section seeks to demonstrate the ”Thresholded Exponent Segmentation”. Initially, a generous raw threshold is implemented; only the bottom percentages are removed.

However, notice that the image contains a large amount of noise. To counteract, we will exponentiate the results to differentiate the more significant ”red points” from the background. Notice that this step will not be able to remove all existing noise.

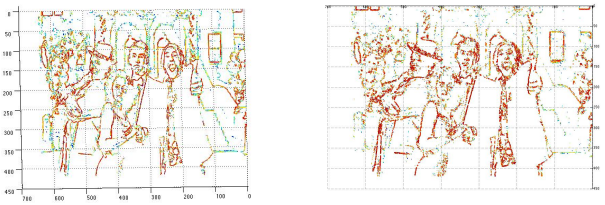

Figure 3. Left: The color in the points represents the absolute value of the modified sobel operator's value assignment. The spectrum runs from green to red: small to large. Right: Notice that there is far less green dots and while this is difficult to represent visually, there is actually more of a noticeable difference between the formerly all "red" points, meaning that even in the higher magnitudes, values are differentiated. This is very useful for distinguishing bewteen similar objects.

Finally, it is good practice to segment the objects into separate groups such that individual processing is easier. From the previous image, it is possible to separate the image into sections where "likely facial objects" are. The segmentation performed splits the image into rectangular regions where regions are slowly recombined based on similarity. After all iterations, the remaining number of boxes represent the number of objects.

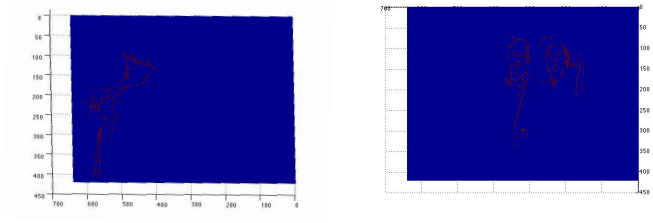

Figure 4. The left and right images indicate the two segmented "main objects" in the region. Notice that it is hard to conclude in the right image that there are two bodies. Later processing will continue segmentation.

C. Step Three: Ellipse Generation

Notice the scarcity of points remaining in these image frames. We can take advantage of that and construct all possible ellipses (with some heuristical constraints) to evaluate possible facial objects. The two-person group will be used as an example.

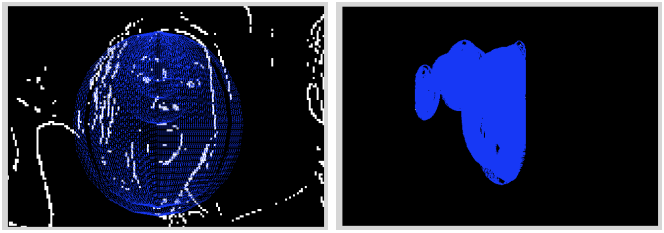

Figure 5. The left demonstrates the sheer quantity of circles generated given an ellipsis center. The right image shows all of the ellipsis generated for the given group. More heuristics shoudl be added to trim the initial set.

After applying heuristics and the simple decision-making algorithm to choose which circles to keep, we get the following remaining. circles. Much of the algorithm here is based on human features like eye presence, facial proportions, etc.

Figure 6. Only a narrow set of circles remain. Notice most of them are overlapping, representing the same object. Note that there are noisy ellipsis as well (See top right of the left face.)

Additionally, a blobbing method is applied to remove redundancy, joining ellipses that overlap and tossing away ones that deviate from the norm. The blobs are then overlayed with the initial image.

Figure 7. The top image represents the result of blobbing from the 2-person group while the bottom image represents the result of blobbing from the remaining group. Note that there is potential for error: some ellipses may persist through the safety nets and obscure itself as a "facial object".

VII. Limitations

There exist limitations with the algorithms that do not allow for effective results in large crowded environments. In the

segmentation step, when repeated exponentiation occurs with the delta mask, each cycle results in homogenization of data and loss of (important) outliers. This means that while we obtain a "big picture" of the surroundings, any specific details or secondary objects could be at risk of removal.

Furthermore, while the ellipse-creation step is very robust, it is exhausting in terms of computation expense. A significant portion of the ellipses drawn are either redundant or improbable to the human viewer. If a good pre-creation filter were to be implemented, the algorithm would be much less costly and perhaps make real-time. Besides processing power, further testing revealed that two stages of filtering for ellipses may not be enough to remove poorly accentuated circles. Often there are circular objects that indeed share similarity with a human's facial features (especially when the jurisdiction rests upon a vector of image-based features). Without context of movement, no additional filter is included to remove these facial imposters.

Overall, the results from this algorithm contain provable error and may not be the most accurate solution to the facial detection problem. However, it provides a fairly good estimate given very limited data - a single image frame. The benefits here are position in terms of resourcefulness and efficiency. If a longer stream of input data were to be included, our extended hypothesis is that the proposed algorithm could be applied for accurate measurements.

Figure 8. Notice that given poor parameters, the ellipse-finding technique will not only find improper circles but miss a large proportion of them.

VIII. Conclusion

This paper presents a new methodology for facial detection in crowd scenarios given limited data samples. The goal of the research is to improve quick and efficient estimation of faces. The most important features of this approach are: the generation of the matrix of an image frame from a sobel operator prioritizing local context, segmentation based on peak splitting, ellipsis creation, and ellipsis blobbing/filtering. We expect increased speeds in algorithms, and more applications given larger samples of data.

Appendix

A. Single Object Detection

While this paper is focused on detection in crowds, a lot of interesting information can be provided by the modified edge operator in terms of analysis of a single facial object in the image frame. Given a solidary object, the operator is useful in classifying different human attributes on the human face like the eyes, ear, mouth, etc – seeing that similar objects have similar delta values.

Figure 9. The graphs shown above represent the effect of 3 different types of deltas matrices applied to the image (top left). A normal deltas matrix of size 10 is shown (top right) - notice the accents on the eyes, noise, and mouth. Much of the background and empty facial regions are easily removable and distinct characteristics are given to the remaining objects (represented by color). A deltas matrix of size 3 and a deltas matrix of size 25 are shown (bottom left, bottom right). Notice that lowering the size increases the amount of lines found, adding detail and noise. Increasing the size blurs objects together but thoroughly separates the background from the primary object. It is a challenge to find the optimal size.

Acknowledgment

The authors would like to thank Professor Ross Walker from the University of California, San Diego for advice throughout the development.

References

[1] S. Anila and N. Devarajan, *Simple and Fast Face Detection System Based on Edges* , 1st Vol. 2nd Iss. International Journal of Universal Computer Sciences.

[2] O. Arandjelovic, *Crowd Detection from Still Images*. BMVC. 2008.

[3] Y. Mohd-Mustafah et. al., *Real-Time Face Detection and Tracking for High Resolution Smart Camera System*. DICTA 2007.

[4] P. Viola and M Jones, *Robust Real-Time Face Detection*. International Jounral of Computer Vision.

[5] D. Reisfeld and Y. Yeshurun, *Robust Detection of Facial Features by Generalized Symmetry*. IEEEXplore.