Fast Detection and Tracking of Faces in Uncontrolled Environments for Autonomous Robots Using the CNN-UM

J. McRaven[*]*, M. Scheutz*[**]*, Gy. Cserey*[*]*, V. Andronache*[**] *and W. Porod*[*]

[*] Department of Electrical Engineering
[**] Department of Computer Science and Engineering
University of Notre Dame, Notre Dame, IN 46556, USA

ABSTRACT: We present a real-time system for the detection and tracking of faces in uncontrolled environments suitable for autonomous robots. The system is based on a CNN algorithm using a CNN-UM chip connected to a robot, which had to perform a face tracking task using a web camera mounted on a pan-tilt zoom unit. The experimental evaluation of the system demonstrates fast, reliable tracking of faces under uncontrolled lighting conditions and with temporary occlusions of the tracked face.

1. Introduction

The detection and reliable tracking of faces in real time is a difficult problem in dynamic environments, where lighting conditions are not controlled and faces can be temporarily occluded. The problem typically becomes intractable on autonomous robots where standard algorithms are not applicable because of their high computational demands.

In this paper, we tackle this problem at two levels: at the hardware level, we add a parallel processor–the CNN-UM ACE4K chip [1, 2, 3, 4]–to the robotic system, which is well-suited for image processing methods; at the software level, we integrate a novel face detection mechanism based on structural features and motion specifically targeted at the CNN-UM's processing architecture with a simple belief revision mechanism that allows the system to cope with detection errors.

The paper is organized as follows: after a quick review of current approaches using the CNN-UM for face detection and presenting an overview of our system setup, we describe the face-detection algorithm, followed by the results of evaluation experiments showing that the system can reliably track faces in adverse dynamic environments.

2. Background

The reliable detection and recognition of faces has been an ongoing research topic for decades and more than 150 different approaches have been reported in the literature (e.g., see [5] for a categorization and benchmark evaluation). Yet, there are no reliable real time detection and tracking algorithms of faces in unconstrained environments suitable for the implementing on autonomous robots with limited computational resources.

Recently, researchers have turned to the CNN-UM, a cellular neural/ non-linear network universal machine implemented in an array of programmable analog processing cells [2]. This

chip is particularly well-suited for real time image processing operations where local computations have desired global effects (e.g., filter operations on images).

One proposal for detecting faces using the CNN-UM chip [6] is based on finding eyes in a face in grayscale images by searching for features of the eyes and eye sockets. While the proposed algorithm can reliably and quickly detect faces from a face database, it has not been tested in uncontrolled environments and it is furthermore not clear if the approach would work in dynamic environments, where faces are not static.

Another project concerned with processing face information based on the CNN-UM attempted to identify of already detected faces [7]. This proposed algorithm first extracts features from the image to estimate the pose of the face, and then transforms the face to create a canonical position by filtering, rotating, and scaling it to a standard size. The standardized face was then compared to faces stored in a database and identification was based on the closest match.

Currently, there is no project that has attempted to detect and track faces in an uncontrolled environment using the CNN-UM on an autonomous robot. In the following, we describe our approach and solution to this problem, starting with a system overview followed by a detailed description of the CNN algorithm that was used to detect faces.

3. System Overview: Hardware and Software Framework

We used a simple web camera as video input source, mounted on the pan-tilt unit of an ActivMedia Pioneer Peoplebot robot. The camera was connected to a PC with the ACE4K PCI board (running the CNN algorithm) via a long USB cable. The PC, in turn, was connected to the robot's own PC via a wireless ethernet connection.[1]

The Aladdin Pro software development environment for the ACE4K chip captures Video frames from the camera using Directshow. The input image is a 160x120 pixel, 24 bit color image, of which a 64x64 pixel subimage is used for face detection on the CNN using the Image Processing Library included with Aladdin Pro, based on the estimation of the location of the face in the larger image. Data suited for digital processing is obtained from processing the subimage by the analog detection algorithm and sent up to the PC, where a program written in the Borland development environment processes the results to determine the likelihood of the presence of a face. The Borland program employs a number of control mechanisms to dynamically adjust the parameters of the CNN algorithm for the best results. If a face is detected, the Borland program sends the coordinates of the face to the camera control program on the robot, which uses a proportional controller to move the pan-tilt unit to keep the face centered. If no face is detected, a search process is initialized, in which other subimages are examined for the presence of a face. If no face is found, the search is started anew with the next frame. The following section presents the details of the employed face detection algorithm.

4. Image and Data Processing

As already mentioned, the CNN chip works together with the Borland program to find and track a face. Section 4.1 describes the algorithm used by CNN chip and section 4.2 describes

[1]While the ACE4K PCI board could be installed directly on the robot (thus making the robot completely autonomous), it was decided against it for the present setup for the simple practical reason that the robot is running RedHat LINUX 9.0, while the ACE4K programming environment requires Microsoft Windows.

the remaining computations performed by the Borland program to detect faces. The latter not only processes the results from the CNN computations, but also uses feedback control to adjust the parameters of the CNN algorithm for better results.

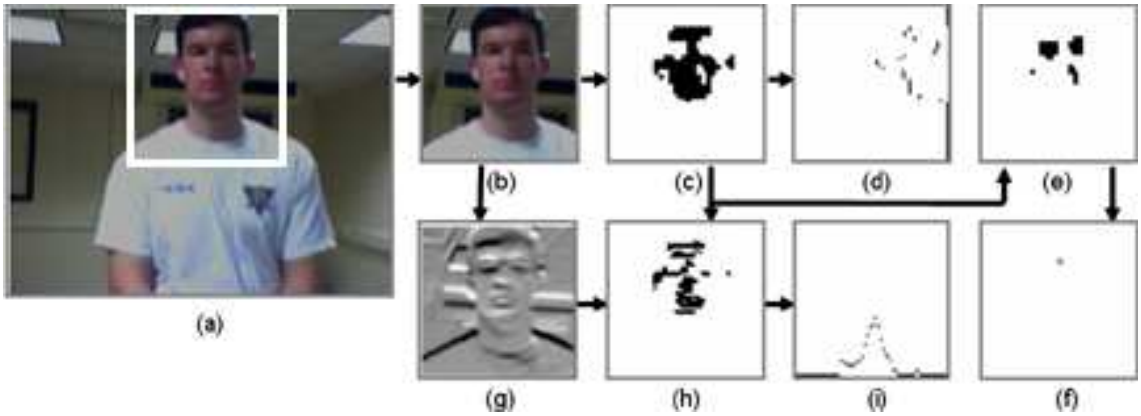

Figure 1: A diagram of the Analogical CNN Algorithm

4.1 Analogical CNN Algorithm

The face detection algorithm was developed for indoor settings, where faces have to be detected against backgrounds with varying color and uncontrolled lighting conditions (e.g., flickering lights). In such an environment, the variety of background objects makes it much harder for a computer to distinguish objects based on contours, hence color information was used for an initial estimation of whether a face could be located in the image.

We receive a three channel RGB image from our camera. To obtain reasonable color information relative to the overall brightness of the image, we factored out the influence of brightness by working only with the difference between the red and green channels. A threshold function with a parameter range between 0-255 (where 128 is an equal amount of red and green, and 255 is all red and no green) was applied. A typical range for facial color was 180-240, where the lower number was varied by the Borland program to obtain better results and make the color detection less dependent on the lighting cast on the face. (Figure 1: b-c)

To ensure that a given color patch was a face, we attempted to detect the eyes. We looked for gaps in the detected color blob (Figure 1: c) as an indication of the presence of eyes. To make sure the gaps were inside the face, shadow operators [8] were used to perform shadow operations independently up, and down, then left and right for the respective eyes. A right shadow operator, for example, adds a pixel to the right of every pixel present. A logical AND operation was performed on all shadowed images and the original color data was subtracted from the result to find the gaps inside the face (Figure 1: c-e). The left and right shadow operators were then independently combined via an AND operation with the result, from which we obtain a centroid for the set of eyes. (Figure 1: e-f)

A second method of eye detection was also employed by using a horizontal projection of the color information to determine the distribution of the color data. (Figure 1: d) A projection, in this case, is essentially pushing all of the pixels to one side, and stacking them as necessary. A ”histogram” of the number of pixels in each row is then formed by removing all but the leftmost pixel. The horizontal histogram was digitally analyzed to find the minima and check that the

position of the minima corresponded to the position of the eyes found in the earlier method. This horizontal histogram was also used to find the height of the face, which is used in a later processing step.

While information about the possible location of eyes in a color patch can significantly improve the detection of faces, background objects with colors similar to facial colors can still interfere enough to reduce the performance of face detection based on eye information. To reduce this interference, edge detection of the area, already determined to be of the correct color, was used to find more complicated structures. A vertical histogram of the edges was made, and the peak was determined to be the center of the face, in a method similar to that used by [7]. The vertical histogram was also used to find the width of the face. A ratio of the height, found earlier, to the width of the face was then used as a final decision making factor in the process. ((Figure 1: g-h-i)

4.2 Data Processing

The Borland program receives three sets of data from the CNN chip: a horizontal histogram of the color information, a vertical histogram of the edge information, and the eye centroid data. These histograms are actually an array of the pixels in the histogram of size 128x1, with x and y components for each of the 64 rows or columns of the histogram. The Borland program also receives a list of possible eye centroids. These histograms are analyzed to find maximums and minimums. For the horizontal histogram of the color, the first significant local minimum is decided to be the eyes. It is checked with the eye centroids to see if there is a correspondence. The vertical histogram of the edges is analyzed to find the center of the face, and to make sure there is enough complexity for it to be a face. The determination of whether it is a face is determined by the instantaneous probability. This instantaneous probability is computed every frame, independent of the other frames, based on the detection of eyes and the size of the color and complexity data.

If there is color information corresponding to the color we are looking for, but there is not determined to be a face, the Borland program will adjust the threshold values to look for a face. It also tries to center the 64x64 cutout area to center the color information.

4.3 The Belief Revision Mechanism

The belief revision mechanism is intended to provide to potential subsequent systems an estimate of how likely it is that a face is present in the image. It is intended to deal with temporary losses of faces that are being tracked due to imperfections of the detection algorithm (e.g., based on lighting conditions, background objects, etc.) as well as possible occlusions. The first step is to decide if what we currently see is a face. The belief revision mechanisms consists of a certainty factor C (between 0 and 1), which at any given time expresses the degree to which the system "believes" that it is seeing a face. C is initially 0 and subsequently update according to the following differential equation: $\Delta C = G*(1-C)*(positive) - D*(1-C)*(\neg positive)$, where $positive$ is a Boolean value that represents a positive identification of a face for the current frame as determined by the CNN and the Borland. G and D are constants determined experimentally for the best results, which influence the increase and decrease of the certainty that a face has been detected. Note that in the present setup the robot will begin a search for a face if C falls below a certain level.

5. Experiments

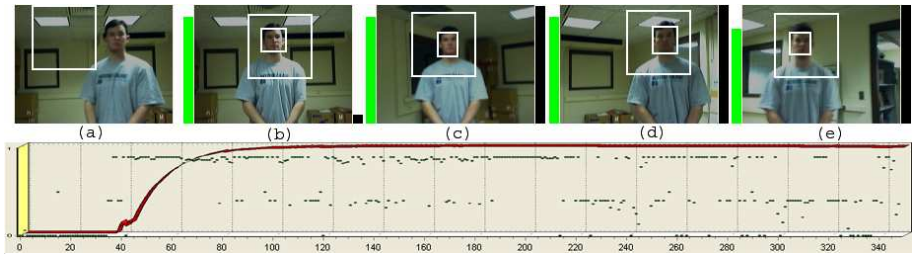

Figure 2: Experiment 1: Simple Tracking (a) Frame 31: Searching for a face. (b) Frame 36: Face found. (c-e) Face tracked through the room.

5.1 Simple Tracking

The first experiment tested whether the robot could find a face and track it across the room as it moved through varying backgrounds. A person entered the field of the vision of the camera. The system found the face, adjusted the threshold for best results, and quickly obtained a high C value. The subject then moved around the room. Figure 2 depicts the sequence of events that took place. Each picture shows the input from the camera. The first box drawn on the picture is the 64x64 frame that was processed by the CNN. Another box is drawn to indicate where the system believes the face is if the instantaneous probability is high enough. The accompanying chart shows a plot of the overall certainty and the instantaneous probability for each frame. The results show that the system was easily able to find and track a face throughout the room.

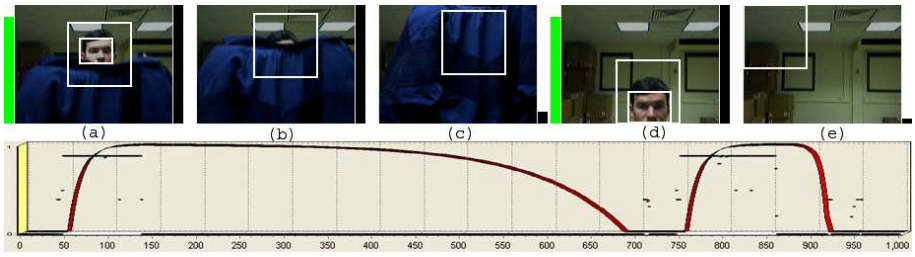

Figure 3: Experiment 2: (a-c) Occlusion Experiment. (a) Frame 137: Last frame in which face recognized. (b) Frame 140: Face Occluded. (c) Frame 673: Still looking in the same place. Begins searching again in the next frame. (d-e) Movement Experiment. (d) Frame 860: Last frame in which face is recognized. (e) Frame 915: Searching for a new face.

5.2 Occlusion vs. Movement

In the second experiment, the ability of the system to distinguish between an occluded face and a face moving out of the view of the system was tested. The objective is to have the certainty factor decay much more slowly if the face is only blocked, than if it disappears from the view of the camera (as in the former case, it is still present in the environment, although not visibly). A person moved into the view of the camera, and allowed the certainty to become very high. The subject's face was then covered with an object, until the certainty decayed close to zero. The object covering the face was then removed to increase the certainty again. This time, the

subject moved downward out of the range of the camera, until the certainty again decayed to zero. Figure 3 shows that the decay rate is much faster in the second case, when the subject moved out of the view of the camera.

This test is important, because the system must realize that if the face is lost, but has not moved, it may reappear in the same position. Thus, the system will continue to look in that location for a while before it gives up. If the face was lost, and it was moving, the system must quickly realize that it should begin a new search for a face.

6. Discussion and Conclusion

In this paper, we proposed a novel system based on the CNN-UM for the detection and tracking of faces on autonomous robots. Our experimental results demonstrate that the system is capable of fast, reliable tracking of faces under uncontrolled lighting conditions and despite temporary occlusions of the tracked faces. Specifically, one complete cycle took 38.4 ms on average, which can be broken down into 2.5 ms (6.5%) for getting frame, 35 ms (91.1%) for processing it on the CNN chip, and 0.9 ms for post-processing (2.4%). Future work with the system will include more extensive tests in indoor and outdoor environment, as well as extensions to allow it to track multiple faces.

7. Acknowledgements

Partial support was provided by the Office of Naval Research through a MURI grant.

References

[1] L. O. Chua and L. Yang, “Cellular neural networks: Theory,” IEEE Trans. on Circuits and Systems, Vol.35, pp. 1273-1290, 1988.

[2] L. O. Chua & T. Roska, Cellular neural networks and visual computing, Foundations and applications, Cambridge University Press, 2002.

[3] A. Zarandy, Cs. Rekeczky, I. Szatmari, and P. Foldesy, “Aladdin Visual Computer,” IEEE Journal on Circuits, Systems and Computers, Vol. 12(6), 2003.

[4] G. Linan, S. Espejo, R. Dominguez-Castro and Rodriguez-Vazquez, “ACE4k: An analog I/O 64 x 64 Visual Microprocessor Chip With 7-bit Analog Accuracy,” Intl. Journal Of Circuit Theory and Applications, Vol. 30, May-June 2002, pp. 89-116.

[5] M.-H. Yand, D.J. Kriegman and N. Ahuja, “Detecting Faces in Images: A Survey,” IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 24. pp. 34-58, 2002.

[6] D. Balya and T. Roska, “Face and Eye Detection by CNN Algorithms,” Journal of VLSI Signal Processing, Vol. 23., pp. 497-511, 1999.

[7] Z. Szlavik and T. Sziranyi, “Face Identification with CNN-UM,” in Proc. ECCTD ’03, Krakow 2003.

[8] Aladdin Professional, Reference Manual, Version 2.5, Budapest 2003.