IMPROVING FACE DETECTION WITH DEPTH

Gregory P. Meyer, Steven Alfano, Minh N. Do

University of Illinois at Urbana-Champaign
Department of Electrical and Computer Engineering

ABSTRACT

Face detection serves an important role in many computer vision systems. Typically, a face detector identifies faces within a grayscale or color image. Due to the recent increase in consumer depth cameras, obtaining both color and depth images of a scene has never been easier. We propose a technique that utilizes depth information to improve face detection. Standard face detection methods, such as the Viola-Jones object detection framework, detects faces by searching an image at every location and scale. Our method increases the speed and accuracy of the Viola-Jones face detector by utilizing depth data to constrain the detector's search over the image. Leveraging a Kinect camera, we are able to detect faces 3.5x faster, while greatly reducing the amount of false positives.

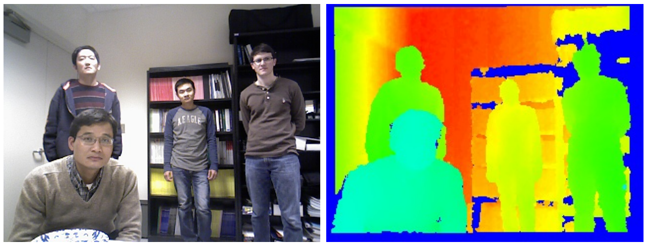
***Index Terms*—** Face Detection, Depth Cameras, Kinect, Real-time

1. INTRODUCTION

Detecting faces is an important initial step for many vision applications. Applications that typically require face detection are face analysis and biometrics, face recognition, face modeling, human-computer interaction, surveillance, etc [1].

Over the past few years, the availability of color images with corresponding depth images (Figure 1) has increased due to the popularity of low-cost depth cameras, notably Microsoft's Kinect. The goal of this work is to utilize the additional depth data to reduce the computational cost of face detection, and in doing so, enabling new real-time applications.

Face detection methods, such as the Viola-Jones object detection framework [2, 3], identifies faces by classifying sub-windows within an image as a face or non-face region. Without prior information, the size and position of a face within the image is unknown; therefore, the detector must exhaustively search the image at every position and scale.



(a) (b)

Fig. 1: Example (a) color and (b) depth images captured by a Kinect camera. In (b), blue indicates a small depth measurement where red represents a large depth measurement.

Our approach uses depth information to identify all positions and scales within the color image that may contain a face. As a result, the face detector is no longer required to classify every sub-window within the image, which is computationally expensive and prone to false detections.

Our proposed method begins by approximating the size of a face at each pixel location based on its measured depth value. With our estimated dimensions, we analyze the geometry of the region surrounding the pixel to decide whether or not it is possible for the pixel to lie on a face. Afterwards, we construct a list of sub-windows to be classified by the Viola-Jones face detector [3].

Our technique significantly reduces the amount of time required to detect faces, as well as, improves accuracy by greatly reducing the number of false detections. In the following sections, we review work related to our proposed method, describe our system in detail, and present experimental results.

2. RELATED WORK

Face detection is used in a wide variety of computer vision and robotic systems. Some of these systems contain sensors, such as stereo cameras, laser scanners, and depth cameras, that are capable of sensing depth. As a result, there have been a few methods proposed to accelerate face detection algorithms using depth information [4, 5].

M. Dixon *et al.* [4] describes a robotic platform where the relationship between a camera and its environment is known. Based on this knowledge, they are able to restrict the face detector's search to only geometrically feasible locations. For

This work was supported by the National Science Foundation under Grant CCF-1218682.

example, regions of the image that would require a person's face to be above the ceiling or below the floor are not classified by the detector. When a stereo camera or laser scanner is available, [4] further reduces their search space by eliminating sub-windows whose physical dimensions are significantly larger or smaller than the average human face.

H. Wu *et al.* [5] proposed a method to accelerate face detection using depth information computed by a pair of cameras. To reduce the cost of stereo depth estimation, [5] only computes a sparse set of depth values. Using nearby depth samples, they estimate the size of the sub-windows within the image. Similar to [4], they avoid classifying sub-windows that are too large or too small to contain a face.

Comparable to previous work, our proposed method uses depth information to approximate the physical size of a sub-window and only classifies windows that are approximately the size of a human face. Unlike other methods, we use the geometrical information contained within the depth image to avoid classifying regions that are unlikely to contain faces.

Furthermore, there has been techniques developed that utilize skin color to accelerate face detection [6]. However, skin color clustering can be affected by illumination. Depth cameras have the benefit of being robust to various lighting conditions.

3. METHOD

Our proposed method leverages depth information to identify regions within a color image that may contain a face. In addition, we use the depth data to estimate the size of the face within each region. As a result, only a small subset of windows within the image need to be classified by the face detector. Our approach avoids the exhaustive and computationally expensive search over the entire image at multiple scales.

Our technique begins by approximating the size of a face at each pixel location based on its measured depth value. Next, we analyze the geometry within the depth image to locate candidate face regions. Finally, we construct a list of sub-windows to be classified by the Viola-Jones face detector.

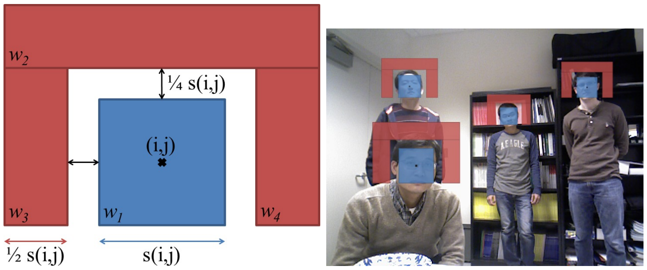
3.1. Approximating face dimensions

If we assume a pixel lies on a face, we can use the pixel's depth measurement to approximate the dimensions of the face. For the $(i,j)^{\text{th}}$ pixel in the image, we compute the size of the face in pixels, $s(i,j)$, using the following equation:

$$s(i,j) = \frac{f \cdot \bar{s}}{d(i,j)} \qquad (1)$$

where f is the depth sensor's focal length, $d(i,j)$ is the pixel's depth value in millimeters, and $\bar{s}$ is the average width of a human face in millimeters [7].

With only a pixel's depth measurement, we can avoid searching multiple scales at every pixel location, which reduces the amount of work required to detect faces, as shown in [4] and [5]. However, our proposed method goes further by analyzing the structure of a group of depth pixels to identify sections of the image that may comprise a face and eliminate regions that cannot contain a face.



(a) (b)

Fig. 2: (a) Adaptive template used to identify candidate face regions, and (b) an example illustrating the template at four different locations.

3.2. Identifying candidate face regions

We use an adaptive template to identify candidate face regions within the image. We compare the depth measurement at the $(i,j)^{\text{th}}$ pixel to the depth values in a local neighborhood surrounding (i,j), where the size of the neighborhood is based on $s(i,j)$. We exploit the fact that depth measurements on a face should have similar value, and depth measurements left, right, and above of the face should be significantly different.

Our adaptive template is illustrated in Figure 2. For the $(i,j)^{\text{th}}$ pixel to lie on a face, the depth values in w_1 should be similar to $d(i,j)$, and the depth values in w_2, w_3, and w_4 should be considerably different than $d(i,j)$. For each window w_k, we compute its average depth value,

$$\mu_{w_k}(i,j) = \frac{\sum_{(u,v)\in w_k} v(u,v) \cdot d(u,v)}{\sum_{(u,v)\in w_k} v(u,v) + \epsilon} \qquad (2)$$

where, $v(i,j)$ is a map of all valid depth measurements, and ϵ is a regularization constant to avoid division by zero. A depth value is considered valid if it is greater than zero and below a threshold T:

$$v(i,j) = \begin{cases} 1 & \text{if } 0 < d(i,j) < T \\ 0 & \text{otherwise} \end{cases}. \qquad (3)$$

We define $T = (f \cdot \bar{s}) \big/ s^*$, where s^* is the minimum size of a face in pixels that the face detector can classify. If a pixel's depth value is larger than T, the size of the face at this location will be smaller than s^*, and the face detector will not be able to identify the face. For our implementation, $s^* = 20$ pixels and T is roughly four and a half meters.

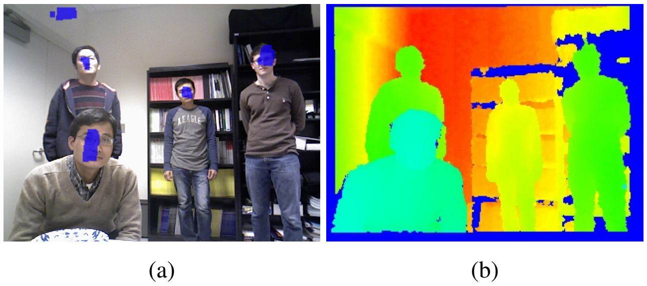

(a) (b)

Fig. 3: (a) Mask generated by our proposed method based on the depth information in (b). Blue pixels in (a) indicate $m(i,j) = 1$.

We generate a mask, m, of candidate face regions by comparing $d(i,j)$ to the average depth values in the surrounding windows:

$$m(i,j) = \begin{cases} 1 \quad \text{if} & \begin{array}{l} |\mu_{w_1}(i,j) - d(i,j)| < \tau_1 \text{ and} \\ |\mu_{w_2}(i,j) - d(i,j)| > \tau_2 \text{ and} \\ |\mu_{w_3}(i,j) - d(i,j)| > \tau_3 \text{ and} \\ |\mu_{w_4}(i,j) - d(i,j)| > \tau_4 \end{array} \\ 0 \quad \text{otherwise} \end{cases} \tag{4}$$

where, $\tau_1 = 100\,mm$ and $\tau_2 = \tau_3 = \tau_4 = 200\,mm$ (these values where determined empirically). To reduce noise in the mask, we perform a open morphological operation [8]. Additionally, we expand the candidate face regions to nearby pixels with similar depth values to increase our chances of detecting faces. To generate m efficiently, we use integral images [2] to compute the window averages, $\mu_{w_k}(i,j)$, as well as, to perform the morphological operations. Figure 3 depicts the mask of candidate face regions produced by our method.

Afterwards, we use m and s to construct a list of sub-windows to be classified by the face detector. For every pixel (i,j) where $m(i,j) = 1$, we add to the list one sub-window of size $s(i,j)$ centered on (i,j). This list is considerably smaller than the list of all possible sub-windows within the image. As a result, our proposed method not only reduces the time it takes to detect faces, but also, reduces the number of false detections.

4. RESULTS

We evaluate the performance of our proposed method on the Cornell Activity Dataset (CAD-120), which features 120 videos captured by a Kinect camera [9]. The sequences contain one of four subjects performing some type of activity, such as, preparing and eating food, picking up and arranging objects, cleaning, etc. Each frame consists of a color and depth image with a VGA resolution (640 × 480), as well as, the positions of the subject's joints within the images. Using the annotated location of the subject's head we can determine whether a detection is a true or false positive.

We use the Viola-Jones face detection algorithm [3] as a baseline for comparisons. The recall of our approach is bounded by the Viola-Jones technique, since we utilize it to perform the final classification of the sub-windows. For this reason, we assume that the set of true positives detected by the Viola-Jones method contains all the faces within the CAD-120 dataset. We use this assumption to analyze the accuracy of our proposed method.

We would also like to compare our method to the techniques proposed by [4] and [5]. It is difficult to accurately contrast our methods as they have different setup requirements, and they use different 3D sensors. However, both methods use depth information in one way or another to estimate the scale at each pixel location. For comparison, we approximate [4] and [5] with our implementation by setting $m(i,j) = 1$ for every pixel (i,j).

Table 1: Evaluation of the Viola-Jones face detection algorithm [3] (Baseline), M. Dixon *et al.* [4] and H. Wu *et al.* [5] (Scale-only), and our proposed method on the CAD-120 dataset [9]

Technique	Recall	Precision	Runtime
Baseline	1.00	0.57	107.31 ms
Scale-Only	0.95	0.73	55.10 ms
Proposed Method	0.94	0.91	30.64 ms

As shown in Table 1, using depth information to avoid searching multiple scales at every pixel location alone slightly improves the performance of the face detector. [4] obtains additional speed up by using the known calibrating between the camera and its environment to avoid classifying sub-windows that would require the face to exist above the ceiling or below the floor. However, this information is not available in the CAD-120 dataset. [5] claims better performance than what is depicted in Table 1. It is likely they obtain their speed up by only using a spare set of depth samples, as a result, they classify a smaller set of sub-windows. Although, it is probable this will affect the accuracy of the face detection.

Our proposed method does not require prior knowledge about the environment, nor does it sacrifice accuracy by using a subset of pixels. Our technique uses the geometrical information contained within the depth images to achieve its performance. As shown in Table 1, our approach accelerates face detection by 3.5x; in addition, it profoundly improves the precision without significantly impacting the recall. The total overhead incurred by our proposed method is 3.96 ms, which is included in the runtime recorded in Table 1. All the techniques were profiled on a Intel Core i7 CPU.

Example results from the CAD-120 dataset are shown in Figure 4. In order to demonstrate that our proposed method is robust to multiple subjects, we captured a few sequences with a Kinect camera, and the results are shown in Figure 5.

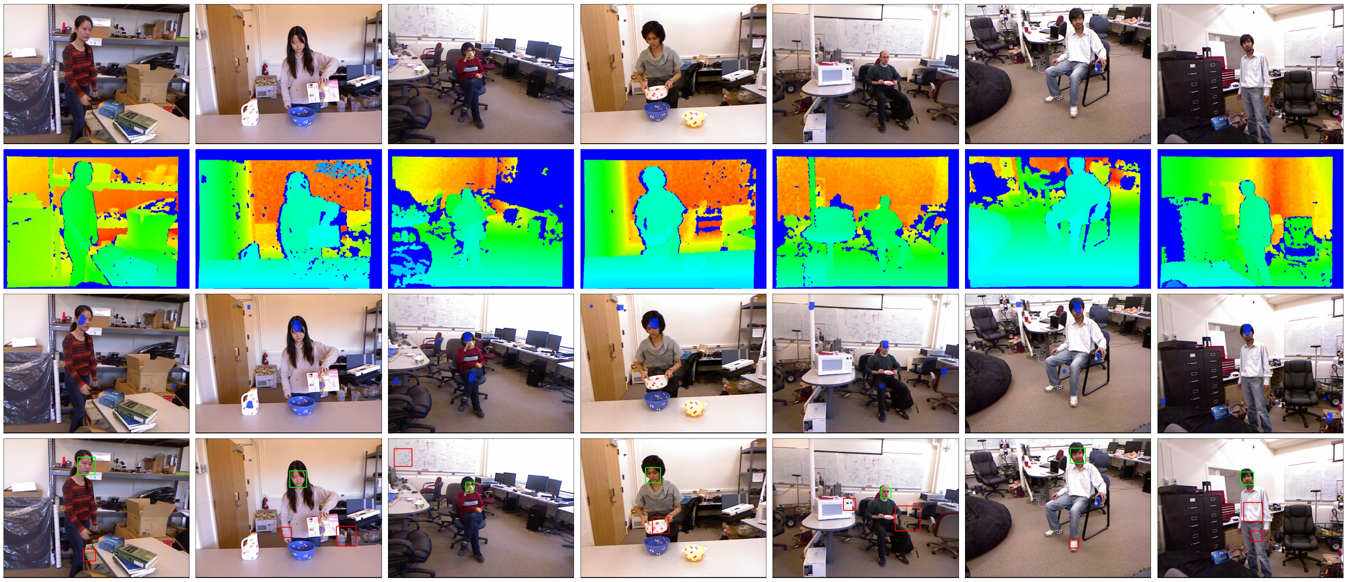


Fig. 4: A set of RGB images (first row) and corresponding depth images (second row) from the CAD-120 dataset [9]. The mask generated by our proposed method is visualized in the third row. In the last row, faces detected by our approach are shown in green; also, false positives detected by the Viola-Jones face detector but removed by our technique are shown in red.

5. CONCLUSION

We presented a method for improving face detection with depth. Our approach utilized the geometrical information within a depth image to identify regions within a color image that may contain a face. As a result, we avoid the exhaustive and computationally expensive search over the entire image at multiple scales. Our method enables us to detect faces 3.5x faster, and it greatly reduces the amount of false detections. By providing additional post-processing time, our technique enables new real-time applications.

Our proposed method is integrated into the OpenCV library [10] through its mask generator feature, which allows us to specify the sub-windows that should be classified without modifying the face detector itself. Consequently, it is trivial to add our method to an existing system that utilizes a Kinect camera and OpenCV. Our algorithm is open source, and the code is available at `http://gregmeyer.info`.

6. REFERENCES

[1] Erik Hjelmas and Boon Kee Low, "Face detection: A survey," *Computer Vision and Image Understanding*, vol. 83, no. 3, pp. 236 – 274, 2001.

[2] Paul Viola and Michael Jones, "Rapid object detection using a boosted cascade of simple features," in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*. IEEE, 2001, vol. 1, pp. I–511.

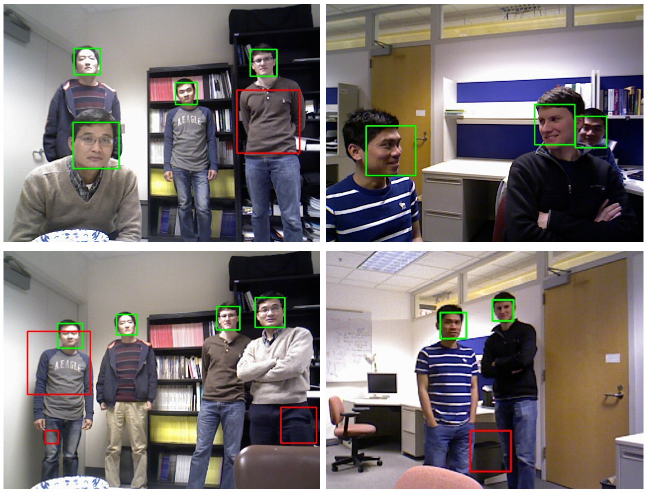


Fig. 5: A set of results from videos captured by a Kinect that contain multiple subjects. Faces detected by our proposed method are shown in green; in addition, false detection classified by the Viola-Jones algorithm but eliminated by our approach are shown in red. Note that our approach works well even in the situation where the faces are close together.

[3] Paul Viola and Michael J. Jones, "Robust real-time face detection," *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137–154, 2004.

[4] M. Dixon, F. Heckel, R. Pless, and W. D. Smart, "Faster and more accurate face detection on mobile robots using geometric constraints," in *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, Oct 2007, pp. 1041–1046.

[5] Haiyuan Wu, Kazumasa Suzuki, Toshikazu Wada, and Qian Chen, "Accelerating face detection by using depth information," in *Advances in Image and Video Technology*, vol. 5414 of *Lecture Notes in Computer Science*, pp. 657–667. Springer Berlin Heidelberg, 2009.

[6] Jure Kovac, Peter Peer, and Franc Solina, *Human skin color clustering for face detection*, vol. 2, IEEE, 2003.

[7] Paolo De Leva, "Adjustments to zatsiorsky-seluyanov's segment inertia parameters," *Journal of biomechanics*, vol. 29, no. 9, pp. 1223–1230, 1996.

[8] Robert M. Haralock and Linda G. Shapiro, *Computer and robot vision*, Addison-Wesley Longman Publishing Co., Inc., 1991.

[9] Hema Swetha Koppula, Rudhir Gupta, and Ashutosh Saxena, "Learning human activities and object affordances from rgb-d videos," *The International Journal of Robotics Research*, vol. 32, no. 8, pp. 951–970, 2013.

[10] Itseez, "OpenCV," `http://opencv.org/`.