IR-DEPTH FACE DETECTION AND LIP LOCALIZATION USING KINECT V2

[1]KATHERINE FONG, [2]JANE ZHANG

[1,2]Department of Electrical Engineering, California Polytechnic State University,
San Luis Obispo, CA 93407, United States
E-mail: [1]kkfong@calpoly.edu, [2]jzhang@calpoly.edu

Abstract— Face recognition and lip localization are two essential building blocks in the development of audio visual automatic speech recognition systems (AV-ASR). In many earlier works, face recognition and lip localization were conducted in ideal lighting conditions with simple backgrounds. However, such conditions are seldom the case in real world applications. In this paper, we present an approach to face recognition and lip localization that is invariant to lighting conditions. This is done by employing infrared and depth images captured by the Kinect V2 device. First we present the use of infrared images for face detection and highlight its improved performance over the traditional method. Second, we use the face's inherent depth information to reduce the search area for the lips by developing a nose point detection. Third, we further reduce the search area by using a depth segmentation algorithm to separate the face from its background. Finally, with the reduced search range, we present a method for lip localization based on depth gradients. Experimental results demonstrated an accuracy of 100% for face detection, and 96% for lip localization.

Keywords— Face Detection; Lip Localization; Audio-Visual Automatic Speech Recognition; Depth; Infrared (IR); Data Fusion; Kinect

I. INTRODUCTION

Automatic Speech Recognition (ASR) plays a pivotal role in human-computer interfaces with applications ranging from education, entertainment to communication and beyond [1]. Traditionally, ASR relies on acoustic-only information. Yet, this type of system suffers from performance degradation in noisy environments [2]. To remedy this, visual and audio modalities of the speech are combined to create a bimodal solution called audiovisual automatic speech recognizer (AV-ASR) [3]. The use of the additional visual information has open new possibilities and challenges. AV-ASR is primarily directed at two research areas – the design of a visual front end where visual speech features are extracted, and the development of an effective strategy to integrate audio and visual information sources. In this paper, we focus on the design of a visual front end consisting of face detection and lip localization. Numerous earlier works in the past decade have devoted to this topic. Crow et al. [4] proposed the use of selected color space and mean-shift algorithm to address the design of front end in visually challenging environments. However accuracy rates of 77% and 44% for face and lip detection only highlighted the difficulty of employing such a system in a real-world environment. Galatas et al. [3] proposed the use of Viola Jones Algorithm to detect the face followed by another Viola Jones Algorithm pass to localize the mouth region within the face. However, the system requires a constant distance between the speaker and the recording device, controlled illumination and a simple background. Similarly, Navarathna et al. [5] also uses the Viola Jones method to detect both the face and lips, but instead of utilizing the entire face image, the lower half of the face was used. Unlike the use of a simple background from Galatas et al, Navarathna et al. used images that were recorded within a car environment. Navarathna et al. achieved a detection rate of 96.92% and 92.36% for the face and mouth respectively. Navarathna et al. reported a false detection rate of 0.94% and 26.3% for the face and mouth respectively.

Although face detection has achieved great success through decades of research, challenges due to varying lighting conditions, presence of facial features such as beards, and glasses still remain [6]. One way to combat the problems is to use light invariant imaging methods. Amongst the various illumination invariant approaches, active near infrared (NIR) imaging technique [7] is promising. NIR is a source of electromagnetic radiation within the beginning of the infrared spectrum range, and borders the visible light spectrum. Its advantages include the ability to be reflected by objects, penetrate glass, and serve as an active illumination source [8]. Such NIR imaging technique can be found on the Kinect V2, an inexpensive motion sensing device from Microsoft that also provides depth data from the same IR sensor and color from an additional image sensor. Unlike the color images where the location and type of the light source, and extreme lighting and shadowing greatly affect the appearance of a face, IR images on the contrary remain the same despite changes in illumination. Even in the dark, IR images can capture distinct details of the face. To alleviate the challenges of face detection due to various lighting conditions, this paper will deviate from the traditional color video input and instead use infrared (IR) video.

While most research studies on face detection have been conducted using color images, there has been a steady flow of research incorporating the use of NIR images in recent years. Li et al. [7] proposed the use

of local binary pattern (LBP) features extracted from NIR images for face detection. In a series of experiments, Li et al. achieve a detection rate of 91.9 percent by incorporating Adaboost with LBP, 32 percent for NIR image with PCA and 62.4 percent for NIR image with LDA. Based on these results, Li et al. concluded that the use of learning-based methods in conjunction with the NIR images offer a good solution for highly accurate face recognition. The biggest constraint reported was that this method is only suitable for indoor applications because NIR images degrade in the sunlight. In other works, Socolinsky et al. [9] conducted face recognition using the PCA algorithm on both NIR and color images as a function of light level. The illumination ranges from bright lighting to near complete darkness. As expected, color image face recognition performed poorly in low light conditions compared to the NIR images, but both perform well for bright light conditions. Hizem et al. [10] also conducted a performance comparison between color and NIR images, and concluded that NIR images perform better when it comes to illumination changes.

II. FACE DETECTION

The Viola Jones framework serves as the face detection algorithm for our design. We use this because the Viola Jones algorithm provides rapid and accurate detection, with accuracy rates that are well above 90% [11]. Although traditional face detection uses color images for face detection, we use IR images for this paper. To validate why we use IR images instead of color images, we first conduct an experiment between the two types of images. Afterwards, we implement the Viola Jones algorithm to detect the face in each IR video frame. Next, to minimize false detections, we applied a median filter on the face bounding box size after each frame. This allows us to discard any bounding box size that do not have a similar value as the median face bounding box size. Finally, the remaining face bounding box parameters can be used for reducing the region of interest (ROI) to the mouth region. Because both the IR data and depth data are created from the same IR sensor, both streams have the same coordinate system. Therefore, the location of the detected IR face can be directly applied to the depth image to create a facial depth image.

A. IR vs. Color Face Detection Experiment and Results

For this experiment, a test set consisting of eight subjects was collected in various backgrounds and lighting conditions from dark to light. In each session, the subject was asked to face directly at the Kinect V2 within 0.5-4.5 meters and talk in front of the Kinect V2 while individual images were captured. For each image capture, 1 frame of color, IR, depth and coordinate map were collected. A total of 108 images for each type of image are available and used in our study. Before we implement the face detection algorithm, image alignment had to be conducted between color and IR images since both are captured on sensors in different locations of the Kinect V2. The coordinate map aligns the two images together by generating a roadmap that maps the depth/IR space onto the color space. In addition to image alignment, the newly aligned IR image and the color image were downsampled to obtain an image size of 640 x 424. Finally, the resulting IR image was re-quantized from 16 bits to 8 bits since color image intensity were represented by 8 bits.

Post image alignment, we implemented the Viola Jones Face detection algorithm twice on MATLAB using an Intel Core i5 1.80GHz 64 bit processor, once for color and another for the IR image. True positives, represented as TP denotes a bounding box that was correctly placed on the face, while false positive, FP denotes incorrectly placed boxes. In addition, images with no bounding box correctly place on the face were regarded as false negative, FN. A summary of the test results is shown in Table I. The results reveal that the IR image is more capable of detecting the face in varying light conditions than color images.

TABLE I. VIOLA JONES FACE DETECTION RESULTS BETWEEN COLOR AND IR IMAGES

	Color			IR		
	FP	TP	FN	FP	TP	FN
Total	7	105	3	10	108	0
Percentage %	6.48%	97.22%	2.78%	9.26%	100.00%	0.00%

B. Face Detection for IR Images with vs. without Median Filter Experiment and Results

To study the effectiveness of our proposed face detection and lip localization algorithms, a database consisting of four subjects (two male and two female) was collected. Each subject participated in two to three sessions with various background and lighting conditions. In each session, the subject was asked to face directly at the Kinect V2 within 0.5-4.5 meters and recite the digits from 0-9 in English in front of the Kinect V2. For each digit, 30 frames of color, depth and IR data were captured simultaneously at 30 frames per second (fps). Therefore, a total of 2700 images for each image type are available and used in our study.

The face detection algorithm, which consist of the Viola Jones Face Detection and a median filter are implemented on MATLAB. False positives and true positives (Fig.1) are judged using the same criteria from the previous section. A summary of the test results is shown in Table II. The results are divided into two categories to compare the results before and after the median filter was applied to the algorithm. The results shown here indicates that the usage of prior knowledge from previous image frames can effectively reduce false positives.

TABLE II. VIOLA JONES FACE DETECTION WITH AND WITHOUT MEDIAN FILTER RESULTS

	Viola Jones IR Face Detection		Viola Jones IR Face Detection with Median Filter	
	FP	TP	FP	TP
Total	117	2700	0	2700
Percentage %	4.33%	100.00%	0.00%	100.00%

III. LIP LOCALIZATION



Fig. 1. True Positive Face Detection Result from various IR Images

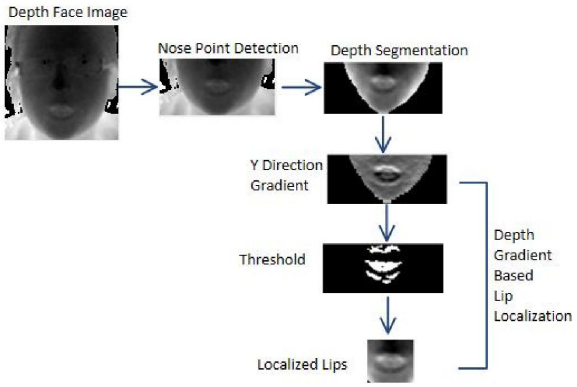


Fig. 2. General Lip Localization Block Diagram Overview

The general lip localization algorithm in Fig. 2 consist of three main stages. From the depth face image, the first stage is to find the nose point. Afterwards, we used the nose point location to reduce ROI from the face to the lower portion of the face. The resulting ROI may contain background pixels surrounding the face pixels, thus, the next step is to use depth segmentation to separate the face pixels with the background pixels. Lastly, we compute the gradient image of the depth face pixels and apply a threshold to create a binary image. From the binary image, we used several rules to find the boundaries of the lips. In the following sections, we will discuss each step in detail.

A. Nose Point Detection

To locate the nose from the detected face, we use the depth image obtained from the Kinect and make the following assumptions: 1. The nose is the closest point to the Kinect. Therefore, it has the smallest depth value within the depth image. 2. The nose lies in approximately the center of the face image. Hence, we can remove potential nose point candidates that may be caused by hair and hat from the border. 3. At nose point, the nose tip has a higher elevation than the bottom of the nose. Once a nose point based on these assumptions is found, its location can be used for ROI reduction.

Three main steps are required for a simple nose point detection algorithm. First, we remove all indices with values that are out the range of 500 to 4500 mm. One constraint of the Kinect is that accurate depth range is limited from 500 to 4500mm. Next, we find the minimum depth value within the depth face image. Finally, once the minimum depth is found, we find the pixel location of the minimum depth value. This simple nose point detection can accurately detect numerous nose points, however, false nose detection occasionally occurs in the eyeglasses area and the hair.

1. Median Filtered Nose Point Detection

From surveying the depth pixels surrounding the false nose points of the glasses area, it is observed that small groupings of sudden change in depth value occurred. Since such pixels act similarly to impulses, a median filter can be used to eliminate these sudden depth value changes. While the use of median filter can remove potential impulses, such filter can also remove true nose points. To ensure that the original minimum depth value is not discarded from the median filter pass, the neighbors surrounding the minimum depth value are considered by applying morphological operations. First, a binary image was created from the median filtered depth image with the minimum depth value as the threshold. Afterwards, dilation is used to expand the region. The resulting dilated binary image is then multiplied against the original depth face image to obtain an expanded search area for the nose point candidate. Finally, the resulting depth image is used to locate the new nose point and its corresponding pixel location.

2. Clear Border Nose Point Detection

Although the median filtered nose point detection algorithm reduced false detection occurrences on glasses, another undesirable location occurs around the outer facial area: hair. To remedy this, another method was created to remove any minimum depth value that is within a certain border proximity threshold. First, a border mask was created with the desired amount of border pixels to clear from each side of the image. Next, the border mask is multiplied to the depth face image, element by element. With the borders of the depth face image cleared out, the out of range filtering, minimum and find blocks are used to locate the nose point depth value and location.

3. Final Nose Point Detection

In this section, we incorporate aalgorithms together to form the final nose point algorithm. The depth face image first passes through the median filtered nose point detection algorithm. From there, a minimum depth mask is made, where all nose point pixels are declared as 1. Afterwards, the border mask is

multiplied with the minimum depth mask. If the sum of the resulting matrix is greater than (for cases with multiple nose point detections) or equal to 1, then the original nose point from the median filtered nose point detection is indeed the true nose point. If the sum was 0, then we assume that the nose pointed detected was around the border, and it would go through the clear border nose point detection to find the new nose point depth and location.

4. Normalization Based on Nose Point

Before we implement algorithms on the facial depth image, we normalize the depth image such that the normalized depth map is independent of the distance between the subject and the Kinect. The normalization procedure begins with applying the out of range filtering on the facial depth image. Afterwards, the detected nose point value is subtracted from the filtered result. This computation results in shifting the overall depth data range to begin at 0, where 0 corresponds to the nose point. In creating this normalization algorithm, the dependency between the distance of the Kinect and subject is removed.

5. ROI Reduction Based on Nose Point

Following the depth normalization, the next step is to reduce the current face ROI based on assumption 3. First, we use the row corresponding to the nose point to discard the upper half of the original face ROI. Afterwards, we find the bottom of the nose row by searching for the first sign change from positive to negative of the corresponding nose column within the y directional gradient image. Finally, we use bottom of the nose row to further discard the all the nose rows within the ROI. What remains of the current ROI is the lower portion of depth face below the nose. Although most of the ROI may contain face pixels, background pixels may exist, thus the next step is to separate the face pixels from the background pixels by using depth segmentation.

B. Depth Based Segmentation

There are many approaches to image segmentations. These techniques include: edge detection, threshold-based, and etc. [12]. Edge detection is based on discontinuities in intensity. Such discontinuities can be found by taking the first and second order derivatives of the image [13]. Threshold-based segmentation uses one or more thresholds to divide an image [12]. Such thresholds are usually chosen based on the image's corresponding histogram.

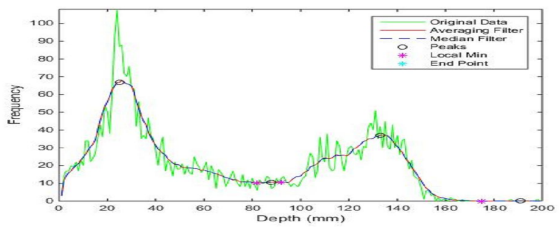


Fig. 3. Histogram Based Depth Segmentation

Within the depth based images, their corresponding histogram have distinct peaks, thus threshold-base segmentation is chosen as the focus of this section. In the previous section, the normalized depth image begins at the depth value of 0, where 0 is the nose point. Based on that knowledge, the first histogram peak will include depth within the facial area. Thus, the first local min after the first local max (peak) of the histogram will serve as the threshold value. Although the histogram have distinct peaks, there can be a lot of little peaks throughout the curve. To remedy this, a smoothing filter, and a median filter are applied to the curve. Upon adding the two filters, vast improvements can be seen in Fig.3. However, with a fixed size for both filters and varying sizes of facial depth images, a safeguard is included into the algorithm to ensure that most of the face is included in the segmentation. This precaution ensures that at least 40% of the image pixels should be segmented as a face. Starting with k=1, the next kth local min depth value will be used until this condition is met.

B. Depth Gradient Based Lip Localization

Once the ROI, the below nose face depth image is used as the input into the depth segmentation algorithm, the resulting segmented depth image should contain only pixels of the face. To locate the lips from the segmented depth image, we further assume the following: 4. There are sudden incline and decline of slope within the mouth depth region as shown in Fig. 4. However, there are no sudden change of incline and decline from either side of the mouth to the edge of the face (towards the chin area). Hence, there should be a gradual increase of depth to the edge of the face, where the gradual increase of depth is caused by the TOF sensor. 5. From the bottom of the lip to the edge of the face, there are no sudden change of incline or decline. Likewise, there should also be a decrease of depth from the bottom of the lip to the edge of the face. Based on these assumptions, the regions with depth declines within the current face region belongs to the mouth. If we can find the depth region with declines, we can use this information to find the lip boundaries.



Fig. 4. Depiction of how the open and closed mouth should decrease and increase in depth value in certain areas of the mouth region

To find the depth regions with declines, we compute an image gradient image from the segmented depth image. Ideally, once we apply a threshold -1 to the y gradient image, only mouth pixels should have a value of 1 within the binary image. From that ideal binary

image, we can implement the algorithm, MouthColFinder to find the vertical (right and left side) boundaries of the lips. The MouthColFinder algorithm begins by summing up individual columns starting from the nose point column. Afterwards, the search continue with the respective direction until a pattern where one column has a sum greater than 0, follow by two consecutive columns with the sum of 0 are found. Next, to find the horizontal boundaries of the lips, we use the algorithm called MouthRowFinder that begins by extracting the nose point column from the binary image. Starting from the bottom row and ascending, the search continues until the row value pattern where one row is 0, and the row above it is 1 are found. However, both algorithms rely on having a clean binary image that clearly depicts the mouths. Sometimes, the resulting binary image may contain clusters that do not belong to the mouth. In the following sections, we use different methods to filter out non lip objects from the binary image for the lip boundaries.

1. Vertical Lip Boundaries

The proposed algorithm for finding the vertical lip boundaries are shown in Fig.5. Up until now, we have used the Sobel operator for calculating the y directional gradients. The Sobel mask for the y direction have smoothing capabilities that can remove noisy depth pixels. However, it also have the capability to remove potential mouth pixels. When such situations happen, it leads to false detections for the lip algorithm due to its inability to encompass the whole mouth. In addition, because we are focusing on the change of depth pixels value in the y direction, we consider the use of gradient operators that uses only 1 dimensional mask, since 2 dimensional mask considers diagonal direction. Gradient methods that use 1 dimensional mask include the central difference gradient operator shown in equation (1) and the intermediate difference gradient operator shown in equation (2).

$$\frac{dI}{dy} = (I(y+1) - I(y-1))/2 \quad (1)$$

$$\frac{dI}{dy} = I(y+1) - I(y) \quad (2)$$

While the central difference gradient operator in equation (1) can remove depth pixel noise, the intermediate difference gradient operator in equation (2) is more accurate as it preserves more detail of the original image. Since the MouthColFinding algorithm is based on summations of columns, the y gradient vector can afford the loss in detail, thus we use the central difference method for the vertical lip boundaries. For the MouthColFinding method, more processing is required since small binary clusters can cause false detection. To minimize the possibilities of small clusters buildup that doesn't belong to the mouth, we first consider gradient pixels with strong edges. This can be done by applying the threshold of -1 to the central difference gradient image. Any y gradient pixel value that is equal or less than -1 are denoted as 1 within the binary image, while are all other pixels become 0. From the binary image, we used the hair border removal method to remove any binary hair pixels that are connected to the border within the hair border threshold. Afterwards, we used morphological operations to remove small binary objects. The resulting binary image is now cleared of any clusters that are not part of the lips. However, the actual mouth may also include weaker edges, thus, we reintroduce the neighboring weaker edges by applying a weaker threshold of 0 to the gradient image to form a new binary image. This new binary image is then multiplied with the dilated image. Afterwards the result is inserted into the MouthColFinder algorithm to find the vertical boundaries of the lips.

2. Horizontal Lip Boundaries

For the horizontal lip boundaries, we use intermediate difference gradient operator from equation (2) since the MouthRowFinding algorithm relies on more precise depth information. In the case of the MouthRowFinding method, once we implemented the intermediate difference gradient method, threshold of -1 to the resulting gradient image. Any y gradient pixel value that is equal or less than -1 are denoted as 1 within the binary image, while are all cases become 0. Afterwards, we use the resulting binary image for the MouthRowFinding algorithm to find the horizontal boundaries.

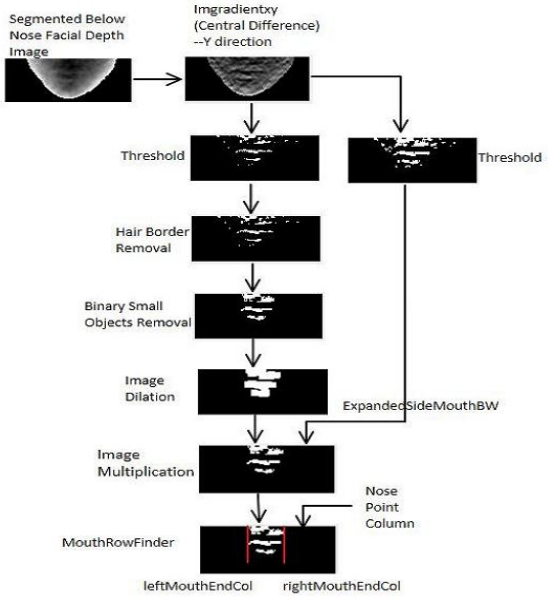


Fig. 5. Vertical Lip Boundary Detection Block Diagram

D. Experimentation and Results

The lip localization algorithm mentioned above are implemented on MATLAB processor. We tested our algorithm against all depth images in our database, which consist of 2700 depth images. A summary of the test results are shown in Table III. For the nose detection, true positive denotes any red points that touches the nose, while false positive denotes red points were not located in the nose area. Sample

images of false positive and true positive of the nose detection are shown in Fig.6. Lastly for the mouth localization algorithm, true positive (Fig.7) denotes a mouth that is fully enclosed within the bounding box, and false positive denotes any bounding box that does not enclose the mouth. In addition, false positive(Fig.9) also include bounding boxes with partially enclosed mouths as well as bounding boxes that include the nose.

TABLE III. NOSE AND LIP LOCALIZATION ALGORITHM RESULTS

	Nose		Lip Localization	
	FP	TP	FP	TP
Total	8	2692	99	2601
Percentage %	0.30%	99.70%	3.67%	96.33%

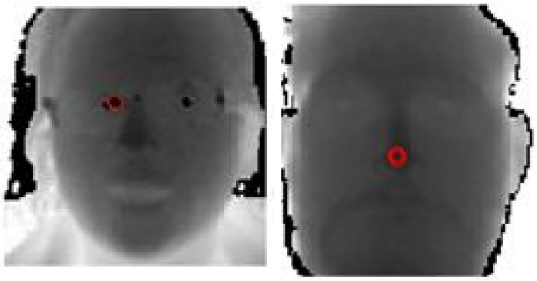


Fig. 6. Nose detection result on depth images: (left) false Positive, (right) true positive

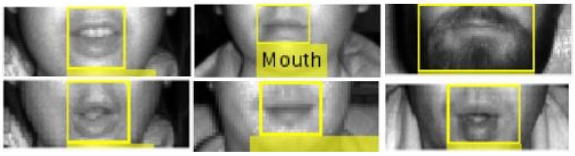


Fig. 7. True Positive Lip Localization Result Projected onto its respective IR images

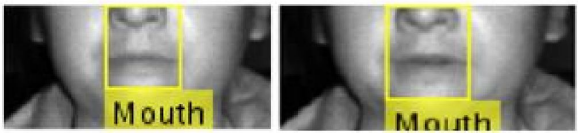


Fig. 8. False Positive Lip Localization Result Projected onto its respective IR images

CONCLUSION

This paper presented an approach to face recognition and lip localization that is invariant to lighting conditions. Through experimentation, the results indicated that IR image produced more accurate face detection results than color image when it comes to varying light conditions. With infrared images, we achieved a face detection system with an accuracy of 100%. From the depth face image, we successfully reduce the search area for the lips by developing a nose point detection that achieved an accuracy of 99%. Finally, for the depth gradient based lip localization, we achieved an accuracy of 96%.

However, some limitations from this paper includes a small database size. In this work, 4 subjects were involved, and all but one session were captured indoor. One session was captured outdoor with a roof above the subject. The system proposed here provides a proof-of-concept and should be tested against a larger data set to study its effectiveness in face detection and lip localization. Additionally, the current system required that the subject directly faces the Kinect to ensure that the nose is the closest point to the sensor. If the subject looks away from the Kinect while talking, the algorithm will not work. Once an accurate and robust visual front end is built, the next step is to develop strategies to extract useful visual speech features followed by audio-visual integration to perform automatic speech recognition.

REFERENCES

[1] Potamianos, A.; Narayanan, S., "Robust recognition of children's speech," Speech and Audio Processing, IEEE Transactions on , vol.11, no.6, pp.603,616, Nov. 2003

[2] R. Stern, A. Acero, F.-H. Liu, and Y. Ohshima, "Signal processingfor robust speech recognition," in Automatic Speech and SpeakerRecognition. Advanced Topics, C.-H. Lee, F. K. Soong, and Y.Ohshima, Eds. Norwell, MA: Kluwer, 1997, ch. 15, pp. 357–384.

[3] Galatas, G.; Potamianos, G.; Makedon, F., "Audio-visual speech recognition incorporating facial depth information captured by the Kinect," Signal Processing Conference (EUSIPCO), 2012 Proceedings of the 20th European , vol., no., pp.2714,2717, 27-31 Aug. 2012.

[4] B. Crow and J. Zhang, "Face and lip tracking in unconstrained imagery for improved automatic speech recognition," Proc. SPIE 7257, Visual Communications and Image Processing, Jan. 2009

[5] Navarathna, R.; Lucey, P.; Dean, D.; Fookes, C.; Sridharan, S., "Lip detection for audio-visual speech recognition in-car environment," *Information Sciences Signal Processing and their Applications (ISSPA), 2010 10th International Conference on* , vol., no., pp.598,601, 10-13 May 2010.

[6] Ming-Hsuan Yang; Kriegman, D.; Ahuja, N., "Detecting faces in images: a survey," Pattern Analysis and Machine Intelligence, IEEE Transactions on , vol.24, no.1, pp.34,58, Jan 2002.

[7] Li, S.Z.; RuFeng Chu; Shengcai Liao; Lun Zhang, "Illumination Invariant Face Recognition Using Near-Infrared Images," Pattern Analysis and Machine Intelligence, IEEE Transactions on , vol.29, no.4, pp.627,639, April 2007

[8] Xuan Zou,; Kittler, J.; Messer, K., "Illumination Invariant Face Recognition: A Survey," Biometrics: Theory, Applications, and Systems, 2007. BTAS 2007. First IEEE International Conference on , vol., no., pp.1,8, 27-29 Sept. 2007

[9] Socolinsky, D.A.; Wolff, L.B.; Lundberg, A.J., "Image Intensification for Low-Light Face Recognition," *Computer Vision and Pattern Recognition Workshop, 2006. CVPRW '06. Conference on* , vol., no., pp.41,41, 17-22 June 2006

[10] Hizem, W.; Allano, L.; Mellakh, A.; Dorizzi, B., "Face recognition from synchronised visible and near-infrared images," *Signal Processing, IET* , vol.3, no.4, pp.282,288, July 2009

[11] Viola, P.; Jones, M., "Rapid object detection using a boosted cascade of simple features," Computer Vision and Pattern Recognition, 2001. CVPR 2001.

Proceedings of the 2001 IEEE Computer Society Conference on , vol.1, no., pp.I-511,I-518 vol.1, 2001

[12] Gonalez,R , “Image Segmentation” in Digital Image Processing, Third ed. Upper Saddle River, NJ, USA: Prentice Hall, 2008, ch 10, sec 1-3, p. 689-761

[13] Raut, S.; Raghuvanshi, M.; Dharaskar, R.; Raut, A., "Image Segmentation – A State-Of-Art Survey for Prediction," Advanced Computer Control, 2009. ICACC '09. International Conference on , vol., no., pp.420,424, 22-24 Jan. 2009.

★★★