CO-OPERATIVE SURVEILLANCE CAMERAS FOR HIGH QUALITY FACE ACQUISITION IN A REAL-TIME DOOR MONITORING SYSTEM

Nurulfajar Abd Manap, Gaetano Di Caterina, Masrullizam Mat Ibrahim, John J. Soraghan

CeSIP, Electronic and Electrical Engineering Department, University of Strathclyde, UK

ABSTRACT

The increasing number of CCTV cameras in use poses a problem of information overloading for end users. Smart technologies are used in video surveillance to automatically analyze and detect events of interest in real-time, through 2D and 3D video processing techniques called video analytics. This paper presents a smart surveillance stereo vision system for real-time intelligent door access monitoring. The system uses two IP cameras in a stereo configuration and a pan-tilt-zoom (PTZ) camera, to obtain real-time localised, high quality images of any triggering events.

Index Terms— stereo vision, image matching, face detection, intelligent system, surveillance

1. INTRODUCTION

Modern video surveillance systems are employed in diverse scenarios. However, the very high number of CCTV cameras in place poses a problem of information overloading, since it is very difficult for human operators to monitor tens of video feeds simultaneously, with the same degree of attention and effectiveness. Smart technologies are adopted in video surveillance [1] in order to automatically analyze and detect events of interest in real-time. The analysis of video feeds on computers systems through image and video processing techniques is called video analytics. An example of a typical indoor video surveillance task is the detection of the face of people entering a room. When multiple views of the same scene are available, as in a multi-camera setup, 3D image processing can be used along with traditional 2D techniques, to better understand the environment surrounding the CCTV cameras. Disparity maps allow estimating the 3D structure of the scene and the geometry of the cameras in space. Many techniques have been proposed in order to determine the homologous points of the stereo pair as described in [2].

In this paper a smart surveillance stereo vision system in the context of a real-time door access monitoring application is presented. The proposed system can detect and record high quality face images of people entering a room, with no human supervision required. Two low resolution IP cameras are used in a stereo configuration, to obtain the 3D location of the object of interest, i.e. people's face, through stereo matching techniques. This positional information is used to control a high resolution pan-tilt-zoom (PTZ) camera, which can locate the object of interest in order to acquire high quality images of it. Starting from the ideas described in [3] for static object detection, the work presented in this paper applies a similar approach to moving targets, in the context of a smart surveillance system for real-time door access monitoring. Moreover, the proposed system uses a face detection algorithm based on Support Vector Machine (SVM) classification, and a stereo matching technique [4–6]. The remainder of the paper is organized as follows. Section 2 describes the system architecture, while a detailed description of the techniques used in the system is given in section 3. Section 4 describes the controller for the PTZ camera. Section 5 contains experimental results and discussion, and conclusions are provided in section 6.

2. SYSTEM ARCHITECTURE

The system presented in this paper has a centralized architecture as shown in Figure 1, with all the software running on a single machine, which can be a general desktop computer. The surveillance sensors used comprise two fixed Arecont AV1300 IP cameras of 1.3 megapixels, and a 5 megapixels ACTi IP Speed Dome (Cam6510), which is a pan-tilt-zoom (PTZ) camera capable of 360° panning, 180° tilting and zooming, with an angular speed of 400° per second. The main purpose of the system is to detect when a door opens and to subsequently acquire high resolution images of the face of the people entering the room. The two IP cameras are set up in a stereo configuration, with the door in their field of view, so that they acquire images from two different angles. Such images can be combined in stereo vision to compute the 3D location of the object of interest, which is the face of the people entering. This information is used to control the PTZ, which pans and tilts to point at the face location. The communication with PTZ and IP cameras and the acquisition of the video streams are entirely carried out over an IP network, ensuring high topological flexibility to the system layout. The system software is divided into two main parts namely (i) the video analytics block and (ii) the PTZ control block. The video analytics block is implemented in Matlab and contains the image and video processing techniques that will be described in section 3. The PTZ control block is implemented in both C and

Matlab, and acts as a controller for the PTZ, dealing with its hardware, firmware and communication protocols. Also the PTZ controller converts the 3D target location computed by the video analytics algorithms into commands suitable for the PTZ.

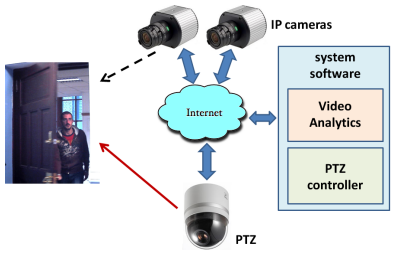


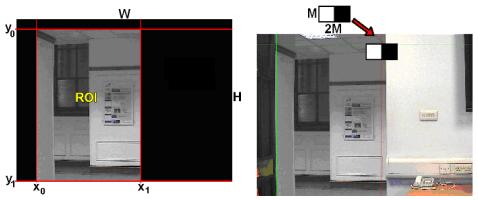
Fig. 1. System design with two IP cameras and a PTZ camera.

3. VIDEO ANALYTICS TECHNIQUES

The smart video analytics algorithms are implemented in Matlab and mainly consist of door opening detection, face detection, block stereo matching and location estimation, as described in the following sub-sections.

3.1. Door opening detection

The main objective of this sub-block is to detect in each new frame whether the door is open or closed. Door opening detection is performed only on one of the stereo image frames: the left image in this case. Since the two IP cameras are fixed, it is reasonable to select a region of interest (ROI) for the door in the $W \times H$ image, either manually or automatically [7], as shown in Figure 2, with y_0, y_1, x_0 and x_1 being the vertical and horizontal coordinates of the ROI, where the face of a person opening the door is expected to be. In this ROI, the vertical side where the hinges of the door are, is identified as 'hinge side', while the other vertical side is identified as 'free side'. In order to detect whether the door is open in the i^{th} frame, a $2M \times M$ binary mask resembling a Haar wavelet is overlapped across the free side at the top, in position $\mathbf{P}_{ref} = (x_1 - M, y_0)$, so that no object can ever occlude this part of the ROI. In usual video surveillance setups, cameras are mounted from the ceiling or at the very top of side walls, therefore the line of sight between camera and top edge of the door is never occluded. The pixel values in the binary mask are multiplied with the corresponding pixel values in the i^{th} frame and summed together to obtain the sum S_i, i.e. the binary mask is convolved with the door image, but only at position $\mathbf{P}_{ref}$. If the binary mask scans the 'door closed' and 'door open' images horizontally, with its position going from $\mathbf{P}_1 = (x_1 - 3M, y_0)$ to $\mathbf{P}_2 = (x_1 + 3M, y_0)$, the graph in Figure 3 is obtained. It is possible to see that in position $\mathbf{P}_{ref}$ the sum S_i can assume two very different values S_{open} and S_{closed}, when the door is respectively open and closed. The only assumption here is that the door, the wall beside it and the background behind it do not all have the same colour. A threshold T_{door} can be set as $T_{door} = |S_{open} - S_{closed}|/2$. For the i^{th} frame, S_i is computed and if $|S_i - S_{closed}| > T_{door}$, then the door is considered to be open and the face detection algorithm is run. The door opening detection step is used in the proposed system, instead of a general motion detector, because a motion detection technique would trigger also for people that are already inside the room, and are just passing in front of the cameras. Instead the proposed door opening detection technique triggers the face detection algorithm if and only if the door has been opened.



(a) (b)

Fig. 2. Door opening detection: (a) region of interest; (b) $2M \times M$ binary mask applied to the door image.

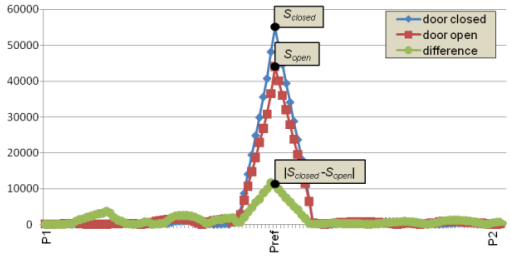


Fig. 3. Behaviour of the sum S_i in both 'door open' and 'door closed' images.

3.2. Face detection

This surveillance system requires a robust face detection algorithm on multi-pose of face. The face entering the door can have a variety of poses depending on the angle view of camera and the way a person enters the door. This face detection algorithm comprises a combination between a knowledge based method and appearance based method in order to develop a robust system. There are three main stages in this algorithm: skin colour segmentation and morphological operation, rectangle bounding formation and specification, and Support Vector Machine (SVM) classification. The skin colour segmentation discriminates between regions that contain the colour of face skin and regions of non-face skin. The challenges of skin segmentation in this application are sensitivity to illumination changing, ethnicity colour skin and the different characteristic of cameras [8]. The combination of three colour spaces RGB,

YCbCr and HSV is applied. The skin region is segmented out using the following rule:

$$\begin{aligned} &\text{if } (r > 95 \wedge g > 40 \wedge b > 20) \\ &\quad \wedge ((\max(r,g,b) - \min(r,g,b)) > 15) \\ &\quad \wedge (|r - g| > 15) \wedge (r > g) \wedge (r > b) \\ &\quad \wedge (140 < c_b < 195) \wedge (140 < c_r < 165) \\ &\quad \wedge (0.01 < hue < 0.1) \\ &\text{then } \textit{(pixel is a skin pixel)} \end{aligned} \tag{1}$$

This rule is based on experiments from three different works on skin detection and segmentation [9–11], in order to provide acceptable performance. In [9] the RGB colour space is used with illumination adaption values; in [11] YCbCr colour space is applied with modulated range, while in [10] HSV colour space is employed. In order to obtain an acceptable segmentation result, morphological operators are used to remove noise and to fill small holes in the skin regions. Normally the small holes in non-skin regions and the noise in skin regions are due to illumination effects. Bounding rectangles are then formed by using a connected components labelling operator. The connected component for bounding area is represented from eight distinct points: leftmost bottom, leftmost top, rightmost bottom, rightmost top, topmost left, topmost right, bottommost left and bottommost right. Each bounding rectangle is then examined in terms of size and pattern shape of the rectangle. The size of rectangle bounding $B(x, y)$ range must comply to the following rule:

$$\begin{cases} \eta < B(x,y) < \varphi, & \Rightarrow B(x,y) \text{ retained} \\ \text{otherwise}, & \Rightarrow B(x,y) \text{ discarded} \end{cases} \tag{2}$$

The smallest rectangle bounding η is defined based on minimum pixels that can represent the features of face. In this algorithm the smallest size η has been chosen as 19×19. For the largest rectangle bounding φ, the value is equal to the image size. Another aspect that needs to be checked on rectangle bounding is the pattern shape. The pattern shape of a rectangle describes the rectangle bounding whether it bounds a face or a non-face object, and it is measured by the ratio of width to height of rectangle defined as follows:

$$0.83 < \frac{width}{height} < 1.27 \tag{3}$$

The limits in (3) have been determined experimentally based on 98 images that contain 561 faces. After all rectangle boundings have been checked in terms of size and pattern, the remaining rectangles are classified whether the rectangle bounding denotes a face or non-face. SVM classifies the bounding rectangles based on horizontal projection features. The horizontal projection of a face has a distinctive pattern (Figure 4) that is used as features for SVM in training and classifying operation. Figure 4 shows three different poses of faces and horizontal projection of eyes, nose and mouth. Such projection is used as features to differentiate between face and non-face objects. The human face identified will be used as the main target object for the stereo matching algorithm discussed in the next part.

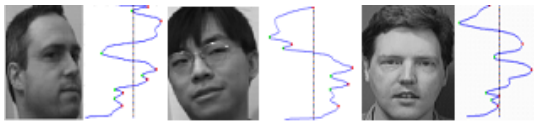


Fig. 4. The profile horizontal projection of face.

3.2.1. Face detection test

The face detection algorithm was tested with the CMU face colour images database [12] that contains a variety of faces in normal room lighting condition. 346 face images with a variety of skin colour tones and different facial poses were used, as shown in Figure 5. The face detection described in this paper correctly detected human faces in 327 images (94.5%), with 19 images (5.5%) erroneously detected. The main cause of the errors was due to pieces of clothing classified as skin.

Fig. 5. Testing face images with different skin colour tones.

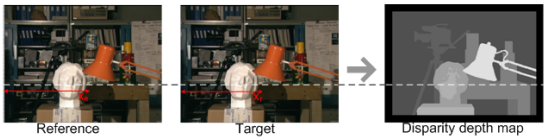


Fig. 6. The disparity map.

3.3. Stereo matching algorithm

The main aim of stereo matching algorithms is to find homologous points in the stereo pair [13]. In stereo matching, two images of the same scene are taken from slightly different viewpoints using two cameras that are placed in the same lateral plane. For most pixels in the left image, there is a corresponding pixel in the right image in the same horizontal line. The disparity is calculated as the distance of these points when one of the two images is projected onto the other. The disparity values for all the image points produce a disparity map. A disparity map is typically represented with a greyscale image, where the closer points are brighter, as shown in Figure 6. The correspondence pixels can be found by searching the element in the right image, which is the most similar (according to a similarity metric) to a given element in

the left image (a point, region or generic feature). Stereo correspondence is conventionally determined based on matching windows of pixels, by using similarity metrics such as sum of absolute differences (SAD), sum of square differences (SSD) or normalized cross-correlation (NCC) techniques. In this research, the SAD metric is selected for faster execution and low computation. In order to determine the correspondence of a pixel in the left image, the SAD values are computed for all candidate pixels in the right image within the search range. Assuming the stereo pair is in the same epipolar line, the disparity estimation is performed by using a fixed-size window. The SAD function is used as a matching cost as follows:

$$SAD(x, y, d) = \sum_{i,j=-n}^{n} D(x, y, i, j, d) \tag{4}$$

$$D(x, y, i, j, d) = |\mathbf{I}_L(x + i, y + j) - \mathbf{I}_R(x + d + i, y + j)| \tag{5}$$

where $\mathbf{I}_L$ and $\mathbf{I}_R$ are the grey-level left and right images respectively, with window size of $n \times n$, and d is the disparity. The best disparity value is determined using the minimum SAD value. As illustrated in Figure 7 the algorithm firstly sets one particular fixed value for d for all the points, and the matching costs are calculated for each image row. Then by varying d, the cost calculation is repeated until the value of d has iterated through the complete disparity range. Consequently a two-dimensional matrix containing the SAD values for each image row is obtained. The width of the matrix is the same as the length of image row, and the height of the matrix is the disparity range.

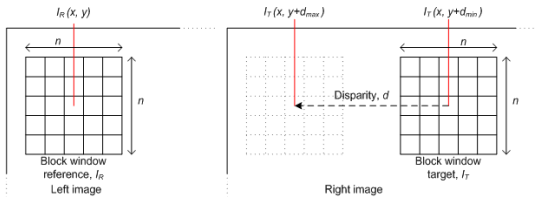
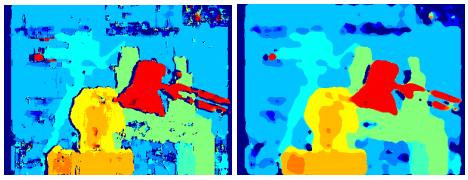


Fig. 7. Matching costs computation based on window size $n \times n$ and disparity range d, with left image as reference and right as target image.

The matching process is performed in both directions in order to ensure consistency and accuracy of the disparity map. At the first stage, the left image is selected as the reference image and right image as the target. The disparity map for this matching is referred to as left to right disparity map d_{LR}. Then a similar process is performed by having the right image as reference and left one as target. In this case, the disparity map is known as right to left disparity map d_{RL}. The result from both matches is used in the next stage that comprises the left-right consistency check. In a stereo pair occlusions can create points that do not belong to any corresponding pixels. In many cases occlusions occur at depth discontinuities, where the occlusions on one image correspond to disparity jumps on the other. In the human visual system occlusions can help to detect object boundaries. However in computational stereo processing it is a major source of errors. Left-right consistency check is performed to reduce the half-occluded pixels in the final disparity map. This can be performed by taking the computed disparity value in one image and re-projecting it in the other image. If the disparity is computed following (6) with threshold $\tau = 1$, then the new disparity map keeps its computed left disparity and defined as $d_{LRC} = d_{LR}(x)$, otherwise it is marked as occluded [6]. The value of τ is set to 1 to ensure that there are exact pixel similarities between the left-right and right-left disparity depth maps.

$$|d_{LR}(x) - d_{RL}(x + d_{LR}(x))| < \tau \tag{6}$$

The disparity maps are refined by using image filtering techniques without explicitly enforcing any constraint about the underlining disparity maps. A common image filtering operator used is the median filter due to the fact that it preserves edges whilst removing noise [5]. The filtering of the disparity map can improve the results in weakly textured regions, where the signal to noise ratio is low and often some pixels are rejected although the disparity can correctly be estimated in the neighbourhood. Figure 8(a) shows the disparity depth map without the filtering process. As indicated in Figure 8(b) the depth map after the filtering process significantly reduces the noise while smoothen out the depth map. The next section explains how to estimate the 3D location from the disparity depth map.



(a) (b)

Fig. 8. Disparity refinement with image filtering: (a) original disparity depth map; (b) disparity depth map after filtering.

3.4. 3D location estimation

Basic geometry is used in order to calculate the 3D location or the range field of the scene. The projection of a 3D physical point on the two image planes requires finding the exact location of the object. The simplest geometry of a stereo system is formed by two parallel cameras with horizontal displacement as shown in Figure 9. The stereo configuration is derived from the pinhole camera model [13]. The disparity can be determined by finding the difference between the X-coordinates of two correspondence points. Referring to Figure 9, $\mathbf{O}_L = (U_L, V_L)$ is the reference camera (left camera) centre point, while $\mathbf{O}_R = (U_R, V_R)$ is the target camera

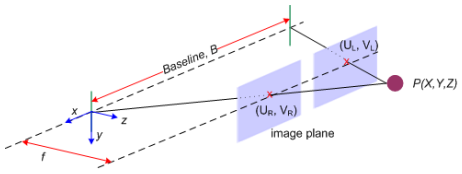


Fig. 9. Stereo camera configuration.

centre point. The implementation of this system is based on parallel cameras, which are shifted along the same horizontal line or X-coordinate, known as the epipolar line, where $V_L = V_R$. The symbol f is the focal length of cameras' lens (the distance from camera centre point to the image plane) and B is the baseline distance (distance between two optical centers, $\mathbf{O}_L$ and $\mathbf{O}_R$). The disparity of the stereo images is obtained as difference between the two corresponding points, U_L and U_R:

$$d = U_L - U_R = \left(f\frac{x}{z}\right) - \left(f\frac{x-B}{z}\right) \tag{7}$$

The location of correct projections of the same point P on the two image planes can determine the exact depth of P in the real world. From (7), the depth z is defined as

$$z = (Bf)/d \tag{8}$$

From (8), the equations used to calculate the exact 3D location of $\mathbf{P} = (x, y, z)$ with respect to the stereo cameras are:

$$x = \frac{Bx_1}{d}, \quad y = \frac{By_1}{d}, \quad z = \frac{Bf}{d} \tag{9}$$

The next section describes the PTZ controller, which uses the information obtained from the stereo matching and 3D location estimation algorithms.

3.4.1. Location estimation test

For the location estimation test, the system is fed with the 22 stereo images, to evaluate the accuracy of the target location estimated by the proposed system ($\mathbf{p}_{estimate}$), with respect to the exact target location ($\mathbf{p}_{exact}$) in the 3D space. The error between each set of estimated and exact locations is computed $e = |\mathbf{p}_{exact} - \mathbf{p}_{estimate}|$. Table 1 shows means ($\mu$) and standard deviations (σ) of the absolute differences between exact and estimated values, for each coordinate axis. The error in X and Y coordinates are very small, while the error in Z coordinate is slightly higher.

4. PTZ CONTROLLER

The PTZ controller module deals with the PTZ hardware, firmware and communication protocols. First, it applies a homogeneous transformation to compute the 3D location $\mathbf{P}_{PTZ} = (x_{PTZ}, y_{PTZ}, z_{PTZ})$ of the target, with respect to the PTZ. If $\mathbf{T}$ is a transformation matrix that transforms from the stereo cameras coordinate frame to the PTZ coordinate frame, the location $\mathbf{P}_{PTZ}$ is computed as:

$$[x_{PTZ}, y_{PTZ}, z_{PTZ}, 1]^T = \mathbf{T}\,[x, y, z, 1]^T \tag{10}$$

The PTZ controller converts the target location $\mathbf{P}_{PTZ}$ into pan and tilt angles, and zoom factor for the PTZ. These values are incorporated into commands for the PTZ, in the form of standard HTTP requests, over the network. The panning angle θ and the tilting angle β are calculated as:

$$\theta = \tan^{-1}\left(\frac{z_{PTZ}}{L - x_{PTZ}}\right) \tag{11}$$

$$\beta = \tan^{-1}\left(\frac{y_{PTZ}}{\sqrt{(L - x_{PTZ})^2 + z_{PTZ}^2}}\right) \tag{12}$$

where L is the distance between IP cameras and PTZ along the X-axes. The zoom ratio instead is proportional to the Euclidean distance between PTZ camera and target object.

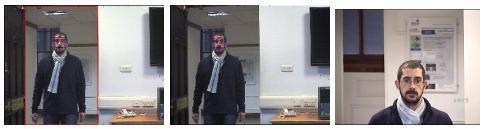
Table 1. Mean and standard deviation of absolute differences.

AXES	X	Y	Z
μ	0.047 m	0.099 m	0.357 m
σ	0.027 m	0.011 m	0.077 m

5. EXPERIMENTAL RESULTS

The system has been tested using different test vectors, i.e. by placing the cameras at different locations with respect to the PTZ and different people as targets. The two fixed IP cameras face directly towards the door that is to be monitored. Images of 640×480 pixels are acquired. The PTZ camera is placed in a different position with respect to the IP cameras, and their relative position is known. During the system setup, the IP cameras need to be calibrated to ensure the images captured are in the same epipolar line. This stage is quite important to ensure accurate distance and depth estimation of the target location. Figure 10 shows a typical image result captured by the presented system. Both left and right detected faces are in the same epipolar line. The searching area for face detection in the left image is minimized to the region of interest, as described in section 3.1. The searching area for the stereo matching algorithm in the right image is limited to a small neighbourhood around the face position in the left image. With this approach, the execution of stereo matching and face detection is ensured to be computationally efficient. The face detection result is processed in the stereo matching and location estimation blocks, to obtain depth and position of the detected object. With this information, the coordinates of the

object are calculated and transmitted to PTZ controller. The coordinates are converted into pan and tilt angles, and zooming factor. The PTZ camera points at the target object and a high resolution image is acquired, as shown in Figure 10(c).



(a) (b) (c)

Fig. 10. Images acquired after face detection and location estimation: (a) left camera view; (b) right camera view; (c) target object image captured by the PTZ.

5.1. Execution time

The mean and standard deviation profile of the recorded execution times are presented in Table 2. The results show that face detection, stereo matching and location estimation steps accounts for less than 50% of the total execution time. The high image acquisition time is due to the transmission of both left and right images over the network, from the IP cameras. The average frame rate is about 8 fps, for the system which is currently implemented in Matlab. It is expected that a dedicated DSP board would significantly speed up the total execution time.

Table 2. Average execution times, in seconds.

	μ	σ
ACQUISITION	0.090 s	0.004 s
FACE DETECTION	0.032 s	0.002 s
STEREO MATCHING	0.028 s	0.001 s
LOCATION ESTIMATION	0.001 s	0.000 s
TOTAL	0.133 s	0.007 s

6. CONCLUSION

A fully automated smart surveillance system using stereo images has been designed and developed. It can automatically detect and zoom in on objects of interest, using image processing techniques. The system processes two stereo images acquired from fixed IP cameras, to calculate the 3D location of the face of people entering the room. This information is used to control a high resolution PTZ camera. The features of this system include door access detection, face detection and high quality images acquisition using the PTZ camera. The system is robust and reliable, and it can be easily integrated with other smart surveillance systems. As future work we plan to implement the described smart video analytics algorithms on a multimedia DSP board, for fast 'in-camera' execution.

7. REFERENCES

[1] A. Valera and S. A. Velastin, "Intelligent distributed surveillance systems: a review," *IEE Proc. - Vision, Image and Signal Processing*, vol. 152, no. 2, pp. 192–204, 2005.

[2] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *Int. J. of Computer Vision*, vol. 47, pp. 7–42, 2002.

[3] N. A. Manap, G. Di Caterina, J. J. Soraghan, V. Sidharth, and H. Yao, "Face detection and stereo matching algorithms for smart surveillance system with IP cameras," in *EUVIP*, 2010, pp. 77–81.

[4] L. Di Stefano, M. Marchionni, and S. Mattoccia, "A fast area-based stereo matching algorithm," *Image and Vision Computing, Elsevier*, vol. 22, pp. 983–1005, 2004.

[5] K. Muhlmann, D. Maier, J. Hesser, and R. Manner, "Calculating dense disparity maps from color stereo images, an efficient implementation," *Int. J. of Computer Vision*, vol. 47, pp. 79–88, 2002.

[6] A. Fusiello, V. Roberto, and E. Trucco, "Efficient stereo with multiple windowing," in *IEEE CVPR*, 1997, pp. 858–863.

[7] X. Yang and Y. Tian, "Robust door detection in unfamiliar environments by combining edge and corner features," in *IEEE CVPR Workshops*, 2010, pp. 57–64.

[8] P. Kakumanu, S. Makrogiannis, and N. Bourbakis, "A survey of skin-color modeling and detection methods," *Pattern Recognition, Elsevier*, vol. 40, pp. 1106–1122, 2007.

[9] J. Kovac, P. Peer, and F. Solina, "Human skin color clustering for face detection," in *EUROCON*, 2003, pp. 144–148.

[10] J. M. Chaves-Gonzalez, M. A. Vega-Rodriguez, J. A. Gomez-Pulido, and J. M. Sanchez-Perez, "Detecting skin in face recognition systems: a colour spaces study," *Digital Signal Processing*, vol. 20, no. 3, pp. 806–823, 2010.

[11] Y. T. Pai, S. J. Ruan, M. C. Shie, and Y. C. Liu, "A simple and accurate color face detection algorithm in complex background," in *IEEE ICME*, 2006, pp. 1545 –1548.

[12] CMU, "Image data base: face," http://vasc.ri.cmu.edu/idb/html/face/ frontal_images/.

[13] A. C. Bovik, *Handbook of image and video processing*, Elsevier, Academic Press, 2 edition, 2005.