Block-Based Search Space Reduction Technique for Face Detection Using Shoulder and Head Curves

Supriya Sathyanarayana, Ravi Kumar Satzoda,
Suchitra Sathyanarayana, and Srikanthan Thambipillai

Nanyang Technological University, Singapore

Abstract. Conventional face detection techniques usually employ sliding window based approaches involving series of classifiers to accurately determine the position of the face in an input image resulting in high computational redundancy. Pre-processing techniques are being investigated to reduce the search space for face detection. In this paper, we propose a systematic approach to reduce the search space for face detection using head and shoulder curves. The proposed method includes Gradient Angle Histograms (GAH) that are applied in a block-based manner to detect these curves, which are further associated to determine the search space for face detection. A performance evaluation of the proposed method on the datasets (CASIA and Buffy) shows that an average search space reduction upto 80% is achieved with detection rates of over 90% for specific parameters of the dataset.

Keywords: search space reduction, face detection, visual search, face localization, computational efficiency, head and shoulder curve.

1 Introduction

In the recent past, appearance based methods have been most widely adopted for face detection [1]. In such methods [2], a sliding window is used to scan the entire image to find faces of all possible sizes. However, computing the classifiers in every sub-window demands high computational power [3]. In [3], methods are proposed to reduce the search space where the sliding window method is applied for face detection.

Skin color is one of the commonly used attribute for reducing the possible search space for face detection [4]. However, this requires sensitive skin color models to accurately segment required regions of interest (ROIs). Upper body detection is explored in [5], to reduce the ROIs for face detection. Oriented Integration of Gradients (OIG) is proposed in [5] as a feature to describe the subparts of human head-shoulder curves, which are then detected using a classifier followed by Hough voting scheme to localize their position.

[2] propose a method based on an active testing framework, in which the image space is decomposed in a quad-tree fashion, and in each iteration of the

R. Klette, M. Rivera, and S. Satoh (Eds.): PSIVT 2013, LNCS 8333, pp. 385–396, 2014.
© Springer-Verlag Berlin Heidelberg 2014

algorithm, regions of the image space with a higher probability of presence of face are refined, while pruning the other regions.

Head and shoulder profile as a feature unique to humans has been used in face localization [6] and human detection [5]. In this paper, we propose a computationally efficient technique to reduce search space for detecting faces by extracting head and shoulder curves. The proposed technique is aimed at short-listing sub-windows that give a higher probability of presence of face. Gradient angle histograms [7] are used in the proposed method to effectively determine the ROIs for face detection. Our method processes the edge information of the image and being a block based approach, the method is scalable to a range of scales for a given image size. The scope of this paper is limited to front facing humans.

2 Proposed Method

In this section, we describe the proposed method in detail. As shown in Fig. 1(a), the proposed method detects the head and shoulder curves of the human being of a given scale. The scale is defined by the ratio $P_X : P_Y$, where P_X is the distance between the top of the head to the shoulders, and P_Y is the distance between the two shoulders in pixels. Given this scale, the proposed method detects possible right and left, head and shoulder curves as shown in Fig. 1, that satisfy the given scale.

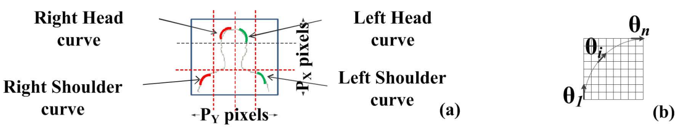
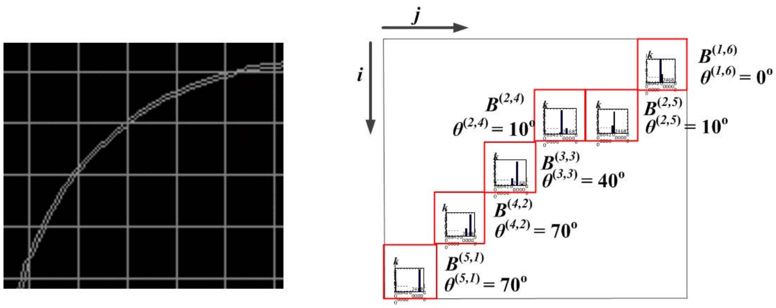


Fig. 1. (a) Head and shoulder curves of a human of scale defined by P_X and P_Y pixels. (b) Illustration showing the linear approximation of the curve [8].

2.1 Gradient Angle Histograms for Curve Detection

As shown in [9], a curve can be divided into smaller segments such that each segment can be approximated by the tangent passing through the mid-point of that segment. This is illustrated in Fig. 1(b). A curve C has a gradual change in tangential orientation from θ_1 to θ_n.

We propose to use gradient angle histograms (GAH) to identify curves of specific curvatures that can be associated with head and shoulder curves of a human, as shown in Fig. 1(a). GAH was shown to be an effective way in [7] to detect linear edges in a block-based approach. Given the edge map E_B of an image block I_B, GAH (represented by $\mathbf{h}$) is the histogram of gradient angles, where $h_i \in \mathbf{h}$ represents the count of edge pixels having the gradient angles in



(a) (b)

Fig. 2. A convex curve is divided into blocks $B^{(i_1,j_1)}$, $B^{(i_2,j_2)}$, ..., $B^{(i_N,j_N)}$. The curve appears as linear segments and GAHs show peaks corresponding to the gradient angles of these linear edges in each block.

the i-th bin. Given the edge block E_B and the gradient angles $\theta(x, y)$ of every edge pixel in E_B, GAH can be used iteratively in the following way to extract possible linear edges in E_B. In each iteration j, we get bin k with the maximum height in $\mathbf{h}$, which represents a possible linear edge in E_B [7]. Edge pixels with gradient angles in the range of $\theta_k - \epsilon \leq \theta \leq \theta_k + \epsilon$, centered around the k-th bin in GAH will result in $E_j \subset E_B$ with possible linear edges. This can be summarized as the following:

$$k = \arg\max_i \mathbf{h} \tag{1}$$

$$E_j = \{e(x, y) \in E_B | \theta_k - \epsilon \leq \theta(x, y) \leq \theta_k + \epsilon\} \tag{2}$$

The above equations are repeated after removing the k-th entry in the GAH $\mathbf{h}$, to get the next peak in $\mathbf{h}$. This is repeated until a termination condition T is reached. We denote this operation as

$$\{\mathbf{\Theta}^B, \mathbf{E}^B\} = \mathbf{\Phi}(\mathbf{h}^B, E^B, T) \tag{3}$$

where $\mathbf{\Theta}^B$ denotes the set of angles θ_ks in the GAH h^B for block B, and the boldfaced $\mathbf{E}^B$ denotes the set of edge maps that are obtained using θ_ks using equations (1). The termination condition depends on the application in which GAH is applied and we will define it later in this section. It is to be noted that GAH differs from HoG (Histogram of Oriented Gradients) in terms of how they are computed and used [1]).

The above GAH formulation can be used to detect a curve as follows. We divide a curved edge, with a known curvature, i.e. given it is a convex or concave curve, using a set of blocks $B^{(i_1,j_1)}, B^{(i_2,j_2)}, \cdots, B^{(i_N,j_N)}$, such that they are along the curve as shown in Fig. 2. It can be seen that the segments of the curve appear as linear edge segments in each block. GAHs are computed in each block as shown in Fig. 2. These GAHs are represented by the set $\mathbf{H}$, given by:

$$\mathbf{H} = \{\mathbf{h}^{(i_1,j_1)}, \mathbf{h}^{(i_2,j_2)}, \cdots, \mathbf{h}^{(i_N,j_N)}\} \tag{4}$$

Applying $\mathbf{\Phi}(\cdot)$ on each block of $\mathbf{H}$, we get $\mathbf{\Theta}^{(i,j)}$. If we consider that highest peaks in each GAH, $\Phi(\cdot)$ on $\mathbf{H}$ will give peaks at $\mathbf{\Theta} = \{\theta_k^{(i_1,j_1)}, \cdots, \theta_k^{(i_N,j_N)}\}$. As indicated in Fig. 2, these gradient angles correspond to the edge pixels of the line segments that form the curve. If these linear segments are approximated as the tangents of the curved segments in the blocks, then they should satisfy (5), i.e.,

$$\theta_k^{(i_1,j_1)} > \theta_k^{(i_2,j_2)} > \cdots > \theta_k^{(i_N,j_N)} \tag{5}$$

2.2 Detecting Shoulder and Head Curves Using Block-Based Gradient Angle Histograms

Given an $X \times Y$ sized image I, it is envisaged to detect shoulder-head region that can be captured within a window of size $P_X \times P_Y$ pixels, where X and Y denote the vertical and horizontal axes respectively. We first divide I into $b_s \times b_s$ sized blocks, such that a curve is decomposed into smaller linear segments in each block. In every block $B^{(i,j)}$, we apply Sobel filters [10] and compute the edge map denoted by $E^{(i,j)}$. For every edge pixel $E_{(x,y)}^{(i,j)}$ in the (i,j)-th block, its gradient angle $\theta_{(x,y)}^{(i,j)}$ is computed using the same Sobel kernels. In every block $B^{(i,j)}$, GAHs are computed for the constituent edge pixels. Therefore, for the $X \times Y$ image we have $\mathbf{H}$ which is set of $\frac{X}{b_s} \times \frac{Y}{b_s}$ GAHs corresponding to all blocks, i.e.,

$$\mathbf{H} = \begin{bmatrix} \mathbf{h}^{(0,0)} & \mathbf{h}^{(0,1)} & \cdots & \mathbf{h}^{(0,\frac{Y}{b_s}-1)} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{h}^{(\frac{X}{b_s}-1,0)} & \mathbf{h}^{(\frac{X}{b_s}-1,1)} & \cdots & \mathbf{h}^{(\frac{X}{b_s}-1,\frac{Y}{b_s}-1)} \end{bmatrix} \tag{6}$$

Applying $\Phi(\cdot)$ on $\mathbf{H}$, we get the following:

$$\mathbf{\Theta} = \mathbf{\Phi}(\mathbf{H}, E, T) = \begin{bmatrix} \mathbf{\Theta}^{(0,0)} & \mathbf{\Theta}^{(0,1)} & \cdots & \mathbf{\Theta}^{(0,\frac{Y}{b_s}-1)} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{\Theta}^{(\frac{X}{b_s}-1,0)} & \mathbf{\Theta}^{(\frac{X}{b_s}-1,1)} & \cdots & \mathbf{\Theta}^{(\frac{X}{b_s}-1,\frac{Y}{b_s}-1)} \end{bmatrix}$$

The termination condition T is defined as a simple threshold $b_s/4$. In other words, $\mathbf{\Theta}^{(i,j)}$ will have θ_ks corresponding to all bins in the GAH $\mathbf{h}^{(i,j)}$ which are higher than $b_s/4$. The selected gradient angles in each $\mathbf{\Theta}^{(i,j)}$ are further constrained in different ways to detect the shoulder and head curves. This will be explained next.

Referring to Fig. 1(a), we identify four curves which are curves of interest, i.e. right and left head curves, and right and left shoulder curves. In terms of the direction of convexity, we consider the right head and shoulder curves as right curves. Similarly, the left head and shoulder curves will hence forth be also called as left curves, unless otherwise stated explicitly.

In order to detect the left and right curves, we divided the GAH $\mathbf{h}$ into two ranges called the right and left angle ranges denoted by Δ_R and Δ_L respectively. Referring to Fig. 3, a peak in Δ_R represents a linear edge that slants diagonally

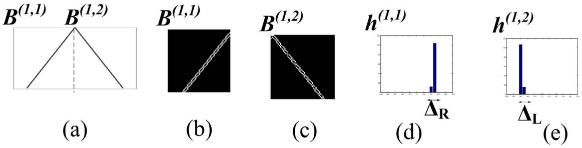


Fig. 3. GAH is divided into right and left ranges Δ_L and Δ_R that will be used to find the right and left shoulder and head curves

upwards, i.e. $\diagup$, whereas a peak in Δ_L of the GAH indicates a linear edge slanting diagonally downwards, i.e. $\diagdown$. It can be seen in Fig. 3 that the right and left slant edges, shown in Fig. 3(b) & (c), result in distinct peaks in different regions of their respective GAHs shown in Fig. 3(d) & (e).

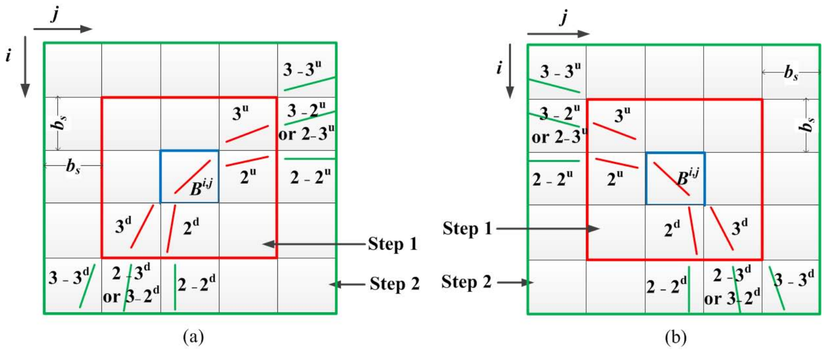


Fig. 4. Ranking of blocks for right and left curve detection

We will now explain the right curve detection process. The same can be applied for detecting the left curve by changing the different parameters like angle ranges etc. Referring to Fig. 4 (a), let us consider a block $B^{(i,j)}$, which is part of the right curve. The $\mathbf{\Theta}^{(i,j)}$ for this block is checked to find $\theta_k \in \Delta_R$, i.e. if there are any edge pixels that are forming a right slant linear edge. This step is the first and critical step because we consider that the right curve must necessarily have a segment that has a gradient angle in Δ_R. If no such θ exists, the next block is processed. If θ_ks exist in Δ_R, then we consider the block for further processing. The $\theta_k \in \Delta_R$ with maximum $\mathbf{h}_k$ is considered as a the anchor angle $\theta_a^{(i,j)}$ and block $B^{(i,j)}$ is considered as the anchor block. This anchor block will be used to check further if there are left and right shoulder curves.

With $B^{(i,j)}$ as the center, we first consider a 3×3 neighborhood of blocks (as shown in Fig. 4 (a)), which shows the possible linear edges that can form the right curve with the center being $B^{(i,j)}$ block. We rank these blocks as p^u or p^d where $p = 2, 3$, and u and d indicate up and down (with respect to $B^{(i,j)}$). p indicates the rank of the block, i.e. the order in which it will be processed when going up or down to form the right curve. These ranks were decided based on the manual inspection of human images in datasets like Buffy dataset [11], CASIA dataset [12]. It was seen that the right shoulder tends to be flat (or horizontal) in block $B^{(i,j+1)}$ as compared to slanting further up in block $B^{(i-1,j+1)}$. Therefore, $B^{(i,j+1)}$ is given as higher rank, i.e. 2^u, as compared to $B^{(i-1,j+1)}$ (ranked 3^u). Similar observations can be made about the blocks below $B^{(i,j)}$ to rank them as shown in Fig. 4 (a).

Therefore, given $B^{(i,j)}$ and the anchor angle $\theta_a^{(i,j)}$, we first go to 2^u ranked block ($B^{(i,j+1)}$) and check for θ_ks in $\mathbf{\Theta}^{(i,j+1)}$ such that the following condition is satisfied: $\theta_k^{(i,j+1)} \leq \theta_a^{(i,j)} - \delta_2$ where δ_2 is the expected change in the gradient angle that should occur if the linear edge segment in the anchor block curves as we go towards the outer blocks, i.e. $B^{(i,j+1)}$. If there are multiple $\theta_k^{(i,j+1)}$ that satisfy the above condition, we consider the θ_k which has the highest count in the GAH $\mathbf{h}^{(i,j+1)}$. These conditions will ensure the curvature condition defined in (5) is satisfied. If $B^{(i,j+1)}$ does not satisfy any of these conditions, then we consider the next ranked block, i.e. $B^{(i-1,j+1)}$ which is ranked 3^u. This must also satisfy the same condition for θ_k but has a smaller δ_3 such that $\delta_3 < \delta_2$. This is because the linear edge in $B^{i-1,j+1}$ is slanting upwards more than $B^{(i,j+1)}$ and hence, it is expected to have a lesser gradient angle variation than $B^{(i,j+1)}$ (based on observation from datasets that was described above).

The same is repeated for blocks $B^{(i+1,j)}$ and $B^{(i+1,j-1)}$ that are below the anchor block, which are ranked 2^d and 3^d respectively. The angles must meet similar conditions as above but with a positive δ_2 and δ_3 because the gradient angles in blocks lower than the anchor block are higher than the anchor block (according to (5). If any of these conditions are not met in the 3×3 neighborhood of $B^{(i,j)}$, no further processing for $B^{(i,j)}$ is done and the next block is processed for identifying the anchor block.

After identifying the block in the 3×3 neighborhood of the anchor block, we perform another check with the blocks that surround this neighborhood. This is the second stage of processing. This is an optional step depending on user requirement in terms of the curvature constraint one wants to ensure. In our experiments, we found that going for one more layer of blocks helped to reduce the false positives. Therefore, we cover a 5×5 neighborhood around the anchor block to ensure that we have captured a curve in it.

The ranks of the blocks in the outer ring of blocks in the 5×5 neighborhood of the anchor blocks are marked as $p - q^u$ or $p - q^d$, where $p = 2, 3$ indicates the rank of the origin block in 3×3 neighborhood, $q = 2, 3$ indicates the rank of the current block. This is shown in Fig. 4 (a). For example, $B^{(i,j+2)}$ has the rank $2 - 2^u$, which implies that it could have a edge from block $B^{(i,j+1)}$ which was previously ranked as 2. Block $B^{(i-1,j+2)}$ has two ranks: $2 - 3^u$ and $3 - 2^u$. If the right curve is detected in $B^{(i,j+1)}$ in the first stage of processing, then we process $B^{(i-1,j+2)}$ after processing $B^{(i,j+2)}$. If the right curve is detected in $B^{(i-1,j+1)}$ in the first stage of processing, then we process $B^{(i-1,j+2)}$ first and then we look into $B^{(i-2,j+2)}$. A similar approach is taken to rank the blocks in the lower half of the neighborhood as shown in Fig. 4.

Now, given the blocks identified in the 3×3 neighborhood of the anchor block $B^{(i,j)}$, which could potentially be having the right curve, we consider these blocks in the 3×3 neighborhood as the new anchor blocks. Therefore, we will have two new anchor blocks B_a^u and B_a^d corresponding to the blocks above and below the main anchor block $B^{(i,j)}$. We now repeat the above process of checking the GAHs with B_a^u and B_a^d, using the ranks for the blocks in 5×5 neighborhood (discussed above). The δ_a parameter is increased or decreased depending on the block that is being processed in a similar approach as described earlier for the first stage so as to satisfy (5).

If we are able to trace a curve within the $k\times k$ window ($k=5$ in our case), then a right curve is considered to be detected, which is anchored at block $B^{(i,j)}$. If there was any discontinuity while traversing from $B^{(i,j)}$ to its $k\times k$ neighborhood, further analysis of $B^{(i,j)}$ is terminated and the next edge block is considered for the entire analysis described above. This is repeated for all blocks in $\mathbf{\Theta}$ to detect anchor blocks that have either the right or the left curve. We generate maps $\mathbf{R}$ and $\mathbf{L}$, such that: $\mathbf{R}\in\mathbb{R}^{\frac{X}{b_s}\times\frac{Y}{b_s}}$ & $\mathbf{L}\in\mathbb{R}^{\frac{X}{b_s}\times\frac{Y}{b_s}}$ where an element in $\mathbf{R}$ say $\mathbf{R}(i,j)$ is set to 1 if we find an anchor block for a right curve at index (i,j). Similarly $\mathbf{L}$ is defined.

2.3 Associating Left and Right Curves for Face Localization

Once all the valid left and right curves are recorded in $\mathbf{L}$ and $\mathbf{R}$, the next level of association is performed to identify regions in the image that could possibly have a front facing human. Recalling that our aim is to detect shoulder-head region which is defined within a $P_X\times P_Y$ pixels sized window, we construct an association window by grouping $u\times v$ blocks, where $u=P_X/b_s$ and $v=P_Y/b_s$. This is illustrated in Fig. 5 (a).

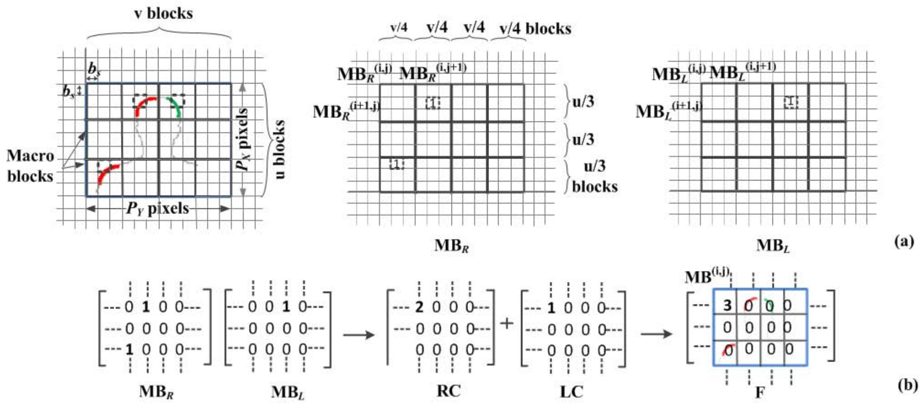


Fig. 5. (a)Association of Right and Left Curves to detect possible shoulder-head regions. (b)Kernels $\mathbb{K}_R$ and $\mathbb{K}_L$ applied on the Macroblocks $\mathbf{MB}_R(i,j)$ and $\mathbf{MB}_L(i,j)$ resulting in Matrices $\mathbf{RC}$ and $\mathbf{LC}$ respectively, and eventually matrix $\mathbf{F}$

Having determined the number of blocks, i.e. $u \times v$, that fit the $P_X \times P_Y$ pixels window, we generate what we call as macroblock left and right arrays, denoted as $\mathbf{MB}_L$ and $\mathbf{MB}_R$ respectively. In order to compute every element in $\mathbf{MB}_R$ and $\mathbf{MB}_L$, denoted by $\mathbf{MB}_R(i,j)$ and $\mathbf{MB}_L(i,j)$, we consider a group of $u/3 \times v/4$ elements in $\mathbf{R}$ and $\mathbf{L}$, in a rastor scan method, as shown in Fig. 5 (a). Therefore, the dimensions of these macroblock arrays are given by: $\mathbf{MB}_R \in \mathbb{R}^{\frac{X}{\frac{u}{3}\times b_s} \times \frac{Y}{\frac{v}{4}\times b_s}}$ and $\mathbf{MB}_L \in \mathbb{R}^{\frac{X}{\frac{u}{3}\times b_s} \times \frac{Y}{\frac{v}{4}\times b_s}}$. Each element $\mathbf{MB}_R(i,j)$ and $\mathbf{MB}_L(i,j)$ is assigned the value 0 or 1 using the following equations:

$$\mathbf{MB}_R^{(i,j)} = \begin{cases} 0 \text{ if } \sum\limits_{r=i\frac{u}{3}}^{(i+1)\frac{u}{3}} \sum\limits_{s=j\frac{v}{4}}^{(j+1)\frac{v}{4}} \mathbf{R}^{(r,s)} = 0 \\ 1 \text{ otherwise} \end{cases} \quad \mathbf{MB}_L^{(i,j)} = \begin{cases} 0 \text{ if } \sum\limits_{r=i\frac{u}{3}}^{(i+1)\frac{u}{3}} \sum\limits_{s=i\frac{v}{4}}^{(j+1)\frac{v}{4}} \mathbf{L}^{(r,s)} = 0 \\ 1 \text{ otherwise} \end{cases} \tag{7}$$

In other words, if any block in the group of $u/3 \times v/4$ blocks (which is one macroblock array element) has an anchor block (either left or right slant edge), we assign 1 to that group (macroblock array element). Fig. 5 (a) & (b) shows an example of macroblock array assignment. It can be seen from Fig. 5 (a) & (b) that if a group of $u/3 \times v/4$ blocks has an anchor block, we assign its corresponding position in macroblock array to 1.

We now define 3×4 kernels $\mathbb{K}_L$ and $\mathbb{K}_R$ in the following way:

$$\mathbb{K}_L = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} ; \mathbb{K}_R = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{8}$$

which are convolved with $\mathbf{MB}_R$ and $\mathbf{MB}_L$ respectively as shown in Fig. 5 (d). Considering the $\mathbb{K}$s are not square-odd matrix, which can be centered at the center of the kernel, the top left element of $\mathbb{K}$s, i.e. $\mathbb{K}(0,0)$s are aligned with every element of $\mathbf{MB}_L$ and $\mathbf{MB}_R$. The convolution output is placed on the top left corner element over which the convolution kernels are moved as shown in Fig. 5(b). The convolution operation results in $\mathbf{RC}$ and $\mathbf{LC}$ matrices given by:

$$\mathbf{RC}(i,j) = \sum_{k=1}^{k=3} \sum_{l=1}^{l=4} \mathbf{MB}_R(i+k-1, j+l-1)\mathbb{K}_R(k,l) \tag{9}$$

$$\mathbf{LC}(i,j) = \sum_{k=1}^{k=3} \sum_{l=1}^{l=4} \mathbf{MB}_L(i+k-1, j+l-1)\mathbb{K}_L(k,l) \tag{10}$$

where $1 \le i \le \frac{X}{\frac{u}{3}b_s}$ and $1 \le j \le \frac{Y}{\frac{v}{4}b_s}$. The matrices $\mathbf{RC}$ and $\mathbf{LC}$ capture the number of right and left curves in the left and right halves of the $u \times v$ window respectively, as shown in Fig. 5 (b). We then get the position of face localization window by generating $\mathbf{F}$ matrix given by: $\mathbf{F} = \mathbf{RC} + \mathbf{LC}$. If $\mathbf{F}(i,j) \ge 3$, then we consider a window with its top left corner positioned at (i,j)-th position to have a face. This condition allows us to check for the presence of at least 3 of the 4 curves that form the head and shoulder curves. This is shown in Fig. 5 (b).

3 Results and Discussion

In this section, we present the evaluation of the proposed algorithm for reducing the search area of face detection. We will first evaluate the detection rate of the proposed algorithm on two different datasets. The first dataset is the Biometric Database (CASIA Face Image Database - CASIA-FaceV5 300-399) with 500 images of 100 subjects [12]. This dataset contains five different images of each subject; the subject being a front facing human in a constrained background set up. The variations among the images include angular movements of the person with respect to the camera, changes in illumination, and imaging distance. Considering that we are demonstrating the algorithm for one particular scale, we have taken 480 images from this dataset, which are of similar scale by manual inspection. The second dataset is a subset of the Buffy Dataset [11]. This contains images with unconstrained backgrounds. 108 images of front facing humans from this dataset were considered for the evaluation. The second performance metric we will evaluate is the amount of search space reduced using the proposed algorithm.

3.1 Accuracy Analysis

As discussed in Section 2, the proposed algorithm being a block based approach, parameters that can influence the detection accuracy for a given scale are block size (b_s), number of bins in the GAH, block-wise GAH threshold setting (T), maximum gradient angle change allowed between blocks while detecting the curves (δ_2 and δ_3 in Section 2). It was found that the block size (b_s) is the most critical parameter that impacts the detection rate, and other parameters were tuned based on observations with respect to the scale. For example, T is set to a value proportional to b_s, $|\delta_2|$ and $|\delta_3|$ are set to 50° and 20° with respect to the scale of the human we have considered for detection. However, these parameters can also be varied and their impact on detection rate can be studied. In the scope of this paper, we will evaluate and discuss the effect of change in block size b_s on the detection rate.

We first generated the ground truth which includes marking a bounding box around the face in each image of the two datasets. During evaluation phase, if the association window resulting from the proposed algorithm has detected a true shoulder-head region and the window encloses the ground truth window, then it is considered as a True Positive (TP) window. In addition to TP windows, there can be false positive (FP) windows. If the proposed algorithm gives at least one TP window for an input image, then we consider that the head-shoulder curve is correctly detected. Therefore, if we have n_{TP} images with at least one TP window, then the detection rate is given by n_{TP}/n_{total}, where the dataset contains n_{total} number of images.

Table 1 gives the detection rates for each dataset and different block setting. It can be seen that the detection rate is 99.5% for CASIA dataset at a block size of 6 and it is 94% for Buffy dataset at a block size of 8. The detection rates reduce for both datasets as the block sizes are increased. This shows that for a

Table 1. Accuracy analysis results

	CASIA				*Buffy*		
	$P_X \times P_Y = 150 \times 200$ $X \times Y = 240 \times 320$				$P_X \times P_Y = 150 \times 150$ $X \times Y = 405 \times 720$		
Block Size	6	8	12	16	8	12	16
Detection Rate	99.5	85.1	79.9	70.7	94.4	74.0	79.6
False Positives per frame	1.78	0.44	0.27	0.08	6.37	2.71	2.36
Number of windows per frame	3.32	1.44	1.30	0.71	8.15	3.74	3.66

given scale of the humans in a dataset, a particular block size gives the highest detection rates.

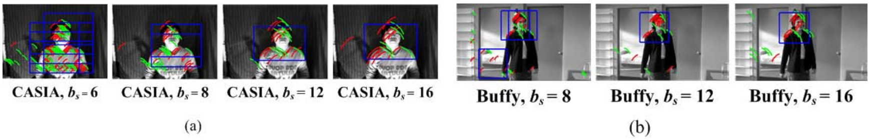


Fig. 6. Detection windows resulting from the proposed algorithm as the block size increases for a specific image in CASIA dataset (a) and Buffy dataset (b)

In Table 1, we list the average false positives per frame (FPPF) for each dataset under varying block sizes. It can be seen from Table 1 that the FPPF is less than 2 and 6.5 for CASIA and Buffy datasets respectively. In other words, an average of 2 false positive windows are detected per frame in the case of CASIA dataset. This number is about 6 in the case of Buffy dataset. The number of false positive windows increases in Buffy because of the unconstrained backgrounds which have more variations as compared to CASIA dataset. In both cases, the FPPF is highest for the smallest block size. This is because, at smaller block sizes, there is a higher probability for curves to be detected from the edge pixels, which are further combined to form curves. In case of higher block size setting, this probability is lesser.

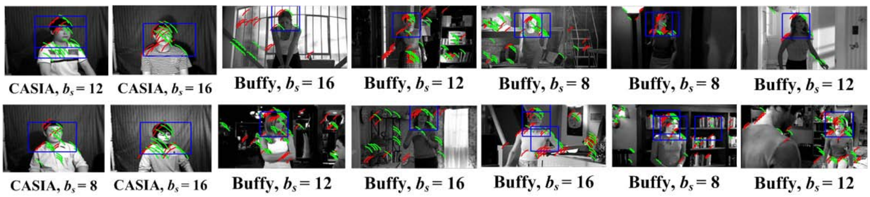


Fig. 7. Detection results under varying background conditions and complexities

In Table 1, we include average total number of windows per frame for each dataset. For each input image, we count the total number of windows, i.e. all

TP windows and FP windows. Table 1 gives the average number of windows over the entire dataset. This metric is particularly important to determine the amount of image area that needs to be searched by the face detection algorithm. Fig. 6 (a) & (b) show that there are more number of windows detected when the block sizes are smaller as compared to higher block sizes and that the FP windows reduce as the block sizes increase. Fig. 7 shows more examples of correct detection windows by the proposed algorithm under varying backgrounds and complexities of the input images.

3.2 Amount of Search Space Reduction

We determine the percentage savings in search area for face detection. If an image has at least one TP window, we take the total search area for face detection as the union of all the detection windows, which include both TP and FP windows. The ratio between the total image size $(X \times Y)$ and the total number of pixels in the search area enclosed by this union of detection windows is used to determine the percentage savings for each image. This percentage savings in search area directly corresponds to a proportional decrease in sliding window based face detection techniques such as [2].

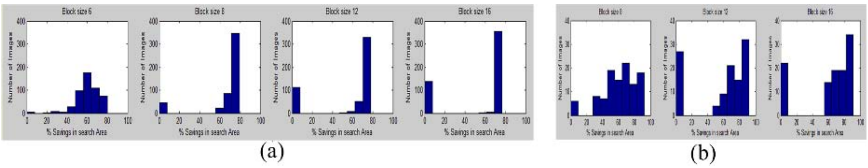


Fig. 8. Distribution of search space savings: (a) CASIA (b_s = 6, 8, 12 and 16 (from L to R)) (b) Buffy (b_s = 8, 12, 16 (From L to R)). x-axis: % savings in search area, y-axis: Number of images.

We show the distribution of percentage savings for each dataset under different block size settings in Fig. 8 (a) & (b). There is a cluster of distribution around 0%, corresponding to the missed detections, i.e. no TP windows. The second cluster is seen at a higher percentage for images that have at least one TP window.

Referring to Fig. 8 (a) which shows the distributions for CASIA dataset, we observe that the percentage savings in search area is spread across bins ranging from 40% to 80% for a block size setting $b_s = 6$. This shows that with this block setting, there are false positives along with the true positive windows, but overall detection rate is high since there are very few missed-detections (the cluster around zeroth bin the histogram is less than 10). Block sizes of 8, 12, 16 show a high concentration of the count in the histogram at 80% savings, which means that they result in very low false positive rates and high precision. But, the bin at 0% is also populated for the three block sizes, which accounts for the cases that are missed-detections under these block size settings. Similar observations can be drawn from the histograms for the Buffy dataset also as shown in Fig. 8 (b) for different block size settings.

4 Conclusions

We have proposed an effective strategy for search space reduction for face detection. The block-based nature of the approach allows for performance gains through parallelism. Evaluation of our algorithm on two standard datasets, CASIA frontal face and Buffy Stickmen was shown to yield a reduction of search space by upto 80% of the image area. It was established that optimal block settings can be derived for a given scale of humans in the image, depending on the required accuracy and savings in search areas. It was shown that the method can cater to varying scales of human faces by deploying it iteratively or by combining GAH information in a hierarchical manner. The method was shown to perform well for profile view of persons on the datasets considered.

References

1. Zhang, C., Zhang, Z.: A survey of recent advances in face detection. Technical Report Microsoft Research (2010)
2. Viola, P., Jones, M.: Robust real-time face detection. In: IEEE ICCV, vol. 2, pp. 747–747 (2001)
3. Sznitman, R., Jedynak, B.: Active testing for face detection and localization. IEEE Trans. PAMI 32(10), 1914–1920 (2010)
4. Xu, D., Chen, Y.L., Wu, X., Ou, Y., Xu, Y.: Integrated approach of skin-color detection and depth information for hand and face localization. In: 2011 IEEE Intl. Conf. on Robotics and Biomimetics (ROBIO), pp. 952–956 (2011)
5. He, F., Li, Y., Wang, S., Ding, X.: A novel hierarchical framework for human head-shoulder detection. In: 4th Intl. Cong. on Img. & Sig. Proc., vol. 3, pp. 1485–1489. IEEE (2011)
6. Sun, Y., Wang, Y., He, Y., Hua, Y.: Head-and-shoulder detection in varying pose. In: Wang, L., Chen, K., S. Ong, Y. (eds.) ICNC 2005. LNCS, vol. 3611, pp. 12–20. Springer, Heidelberg (2005)
7. Satzoda, R.K., Suchitra, S., Srikanthan, T.: Gradient angle histograms for efficient linear hough transform. In: 16th IEEE Intl. Conf. on Img. Proc. (ICIP), pp. 3273–3276 (2009)
8. Kolesnikov, A.: Constrained piecewise linear approximation of digital curves. In: 19th Intl. Conf. Pat. Rec (ICPR), pp. 1–4. IEEE (2008)
9. Zucker, S.W., David, C., Dobbins, A., Iverson, L.: The organization of curve detection: Coarse tangent fields and fine spline coverings. In: 2nd Intl. Conf. on Comp. Vis., pp. 568–577 (1988)
10. Gonzalez, R.C., Woods, R.E.: Digital image processing, vol. 2 (2009)
11. Ferrari, V., Eichner, M., Marin-Jimenez, M., Zisserman, A.: Buffy stickmen v 3.01 dataset
12. Chinese Academy of Sciences Institute of Automation, C.A.: CASIA-FaceV5 dataset