2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops

GPU-accelerated Human Detection using Fast Directional Chamfer Matching

David Schreiber, Csaba Beleznai, Michael Rauter
AIT Austrian Institute of Technology
Video- and Security Technology, Donau-City-Straße 1, 1220 Vienna, Austria
david.schreiber@ait.ac.at

Abstract

We present a GPU-accelerated, real-time and practical, pedestrian detection system, which efficiently computes pedestrian-specific shape and motion cues and combines them in a probabilistic manner to infer the location and occlusion status of pedestrians viewed by a stationary camera. The articulated pedestrian shape is approximated by a mean contour template, where template matching against an incoming image is carried out using line integral based, Fast Directional Chamfer Matching, employing variable scale templates (hybrid CPU-GPU). The motion cue is obtained by employing a compressed non-parametric background model (GPU). Given the probabilistic output from the two cues, the spatial configuration of hypothesized human body locations is obtained by an iterative optimization scheme taking into account the depth ordering and occlusion status of individual hypotheses. The method achieves fast computation times (32 fps) even in complex scenarios with a high pedestrian density. Employed computational schemes are described in detail and the validity of the approach is demonstrated on three PETS2009 datasets depicting increasing pedestrian density.

1. Introduction

The human detection task is a core issue in many applied fields of computer vision such as video surveillance, automotive safety and human-computer interaction. During the last two decades, the pedestrian detection problem has received a great amount of interest and various representations and detection schemes have been proposed. Despite the significant scientific achievements, the human detection task still remains a challenge. Its main complexity stems from the fact that image content comprises a daunting variability and it is ambiguous by objective measures. Images might contain a large number of humans, with possible interactions, occlusions and additional photometric and geometric variations. Variability also implies a large search space (position, scale, pose, etc.) typically leading to a high computational complexity. However, most practical applications require real-time performance thus calling for computationally powerful hardware, optimized implementation and novel insights.

Motivated by these challenges, we present a human detection framework which is capable of detecting humans in moderately crowded scenarios in real-time by computing shape and motion cues and estimating the spatial configuration of humans in the image in a Bayesian framework. Both cues are computed by algorithmic concepts having inherent parallelism, thus being good candidates for GPU implementation. The paper presents implementation details how shape and motion cues are used to build a pedestrian-specific detection scheme, utilizing concepts of shape-based representation [3, 4], shape-based matching [19], and compressed non-parametric background subtraction [21]. This paper extends previous work [3, 4], by replacing the oriented filters approach to contour-based template matching with the variable scale Fast Directional Chamfer Matching algorithm [19]. Detection results are presented for several publically available benchmark videos and demonstrated results exhibit state-of-the art performance at computation speeds reaching and even exceeding real-time.

The paper is organized as follows: Section 2 gives a concise overview on the most important human detection concepts, especially focusing on representations of shape and motion. Section 3 describes the overall concept of the proposed human detection method and provides details on the shape and motion-based detection modalities and their computation. Section 4 presents and discusses experimental results and their evaluation. Finally the paper is concluded in Section 5.

2. Related work

Due to the great practical interest in the human detection problem, significant amount of work has been published over the last decades, especially in recent years. Detailed reviews on various detection schemes, representations and evaluations can be found in [10], [8] and [12]. Proposed approaches target different sub-tasks, such as generic human detection, pedestrian detection in street-level scenarios, de-

978-0-7695-4990-3/13 $26.00 © 2013 IEEE
DOI 10.1109/CVPRW.2013.93

tection and pose estimation and visual surveillance. In our case we focus on the visual surveillance scenario where a single stationary camera observes a scene containing a varying number of humans.

Detection approaches can be grouped with respect to the spatial extent of the employed representation as monolithic (full-body) and part-based detectors. Monolithic detectors such as the popular HOG detector [6] and the Region Covariance detector [18] has produced promising results on challenging datasets, while significant occlusions, strong articulation still pose a problem. Part-based approaches or local representations, e.g. [14], avoid some of these problems by decomposing variable structure into simpler parts.

2.1. Shape-based matching and detection

Shape based matching makes use of the object's contour, rather than its appearance. Although proposed decades ago, Chamfer matching using Distance Transform [2] remains the preferred method for shape matching concerning speed and accuracy. There exist several new variants of Chamfer matching mainly to improve the cost function using orientation information [24]. Recently, [16] reported improving the accuracy of Chamfer matching while reducing its computational cost. They represent the edge image as a collection of line segments, and use line integral images to speedup matching of segments, namely achieving a sublinear complexity (linear in the number of line segments) for the matching part.

Most of the attempts so far to implement image matching on a Graphics Processing Unit followed the appearance based approach, e.g. template matching based on 2D correlation, or key-point matching (see [19] for a survey). In contrast, [19] was the first work to present a full implementation of a shape-based matching algorithm on a GPU. In particular, they have implemented a variant of the Fast Directional Chamfer Matching algorithm (FDCM) of [16]. While the algorithm of [16] employs model templates of fixed size, [19] extends the algorithm to handle templates with variable size, to account for perspective effects.

Shape-based models mostly rely on image features in form of gradients, edges, or use a computed segmentation (blob, color-based) to derive or validate a shape hypothesis. [11] uses a hierarchically structured human template tree to capture human shape variations and efficiently guide a chamfer distance based matching step. [29] also employ chamfer matching of a structured template set and tightly couple with a color-based foreground-background segmentation algorithm to form an iterative joint segmentation and detection procedure. [30] use a parametric human body model composed of elliptic shapes and probabilistically infer the most likely human configuration in the image using a computed motion segmentation. Blob-based motion segmentation is also used by [20] to implicitly capture local shape variations by learning a codebook of local descriptors. [15] decomposes the human body into parametric parallelogram-shaped parts and generate a compact shape tree for efficient model evaluation.

Shape-specific low-level image features are also used in discriminatively learned models. [28] introduce short straight and curved edge segments or edgelets which are selected in a boosted feature selection step to train several body part detectors. The individual part detector outputs are considered jointly to infer the most probable human configuration and occlusion status within the observed scene.

2.2. Motion-based detection

Motion is a strong cue for the pedestrian detection task. In the visual surveillance context, background subtraction plays an important role since it is capable of generating a coarse motion segmentation at a low computational cost. Most background modeling techniques build a pixel-based model. However, the temporal evolution of a given pixel's intensity values typically does not follow a single unimodal distribution. Irregular lighting changes, repetitive background motion or scene changes generate more complex distributions requiring statistical models with more detail. The Mixture of Gaussians (MoG) approach was introduced [23] to model a distribution by a fixed set of Gaussians. However, MoG exhibits certain disadvantages due to its parametric nature. Kernel density estimation [9] as a non-parametric technique can deal with multi-modality in the distribution of background intensities, nevertheless, the method is very memory and time consuming. To overcome computational limitation of this technique, [13] proposed a compressed non-parametric representation using a codebook model. Samples at each pixel are clustered into a set of code-words based on a color distortion metric together with brightness bounds. In [21], a background modeling algorithm for a practical surveillance system was proposed. It utilizes a compressed non-parametric representation, significantly simplifying the work of [13]. It was conceived such that a GPU implementation becomes rather straightforward, achieving ultra real-time performance.

3. The detection approach

3.1. Outline of the method

Given a digital image we would like to estimate the spatial configuration of humans (c^*) such that the hypothesized configuration best describes the observed image features I. Hence the detection task is postulated as a *maximum a posterior* (MAP) estimation problem:

$$c^* = \arg\max_{\mathbf{c}} P(\mathbf{c}|I), \tag{1}$$

A *configuration* encompasses a set of human hypotheses $c = \{h_1, h_2, \ldots, h_n\}$, where n denotes the number of hu-

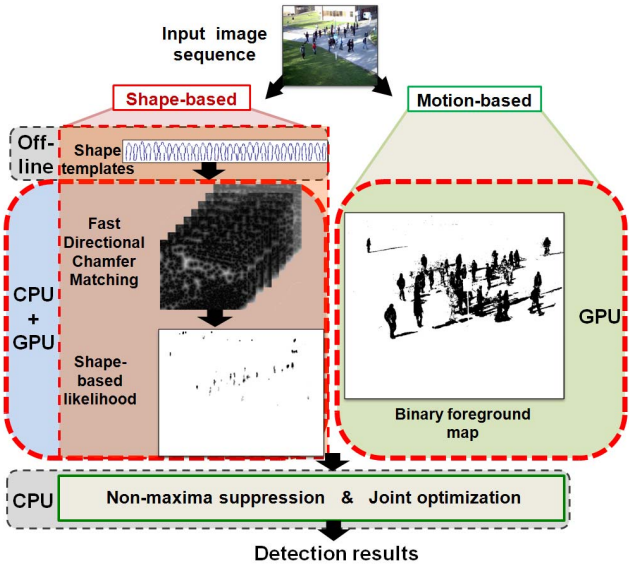

Figure 1. Illustration showing the overall scheme of our human detection approach. Processing steps performed off-line, on the GPU and CPU are indicated as an overlay. All intermediate result images are inverted for better visibility.

mans forming the configuration. A given human hypothesis h_i is characterized by a foot position $\mathbf{x}_i = (x_i, y_i)$ in the image and a corresponding shape C_i represented by a contour template: $h_i = \{\mathbf{x}_i, C_i\}$. According to Bayes theorem the posterior probability is proportional to:

$$P(\mathbf{c}|I) \propto P(I|\mathbf{c})P(\mathbf{c}), \tag{2}$$

where $P(I|\mathbf{c})$ is the joint image-based likelihood and $P(I|\mathbf{c})$ denotes the prior probability of a configuration. All spatial arrangements of individual human models are considered equally probable, therefore the prior depends on individual human model parameters (C) only. We assume that pedestrians stand upright on a common ground plane. We perform an off-line calibration step estimating a linear model $H(y)$ of the projected 2D human height in the scene, as a function of the vertical position y. The human shape is represented by a mean shape consisting of 13 line segments, generated from 120 pedestrian images (plus their mirror images) of the INRIA dataset [6], annotated manually.

The employed cues are shape and motion. Shape models are matched against the obserevd directional Distance Transform images, and motion probabilities are computed from a binary foreground/background segmentation generated by the codebook-based background model. Assuming independence between the two cues, the image-based likelihood can be written as :

$$P(I|\mathbf{c}) = P(I_c|\mathbf{c})\, P(I_m|\mathbf{c}), \tag{3}$$

where $P(I_c|\mathbf{c})$ and $P(I_m|\mathbf{c})$ denote the shape-based and motion-based likelihoods, respectively.

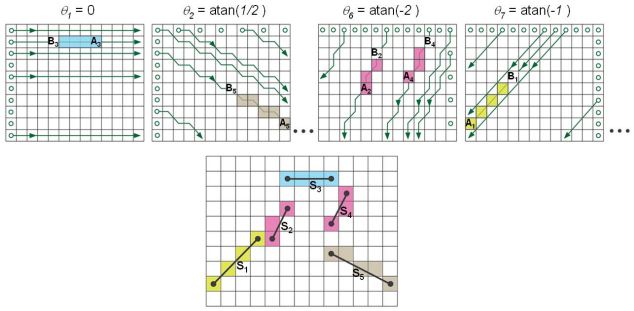

Figure 2. Illustration showing the construction of integral images by oriented string scans for different orientations. Dots represent starting locations of individual string scans. The bottom right image depicts an example contour template consisting of five line segments, where line integrals (sum of values at pixels color-coded according to orientation) can be efficiently computed based on the integral images.

The overall human detection scheme can be concisely described as follows: Both cues are computed on the GPU in separate stages (shape cue computed on a hybrid CPU-GPU). A final joint optimization step (computed on the CPU) estimates the configuration maximizing the posterior given the computed shape and motion-based probabilities. Figure 1 illustrates the overall human detection concept with the required inputs and generated intermediate and final outputs.

3.2. Shape-based detection

The integral image concept [5, 22] has been widely used to speed up the computation of region-based statistical measures, such as area sums [26], covariance [25] and co-occurrence [27]. [3] describes the construction of multiple integral images by oriented string scans over the entire image in order to efficiently compute integrals along oriented linear contour segments. Efficient integration permits fast evaluation of contour-based features. Figure 2 shows the rasterization of scanlines for some orientations. The bottom image shows an example for a contour-based template consisting of five line segments. Using the precomputed integral images, the sum of pixel values along each of the line segments can be computed using a single arithmetic operation, independent of location and scale, by simply subtracting the value of the start-point from the value of the end-point, of the corresponding line integral image.

To detect human shapes, we follow [19] and use a variant of the Fast Directional Chamfer Matching algorithm [16], which is an extension of the Distance Transform algorithm incorporating the line-integral approach [3] to speed up computation time. [19] extends the method of [16] to incorporate the scale variation of objects in the image. The algorithm first computes 8 directional gradient images. The next step is to compute the Distance Transform for each direc-

tion. Unlike the original implementation proposed in [16], they propagate costs only in 2D (image plane) but not in a third dimension (quantized edge orientations), since they have found that this does not give any substantial improvement, and the additional computational costs may be saved. The distance values are limited by a threshold which depends on the size of the human template as follows. Denoting the linear dependence of a persons height on its footpoint vertical position y as (see [19]):

$$\cos(\theta) \cdot \mathbf{H}(y) - \sin(\theta) \cdot y - \rho = 0, \tag{4}$$

the maximal distance value, as a function of the vertical position, is defined by (a, b and k are additional parameters):

$$f(y) = \begin{cases} a, & y \leq \frac{1}{\tan\theta}\left(\frac{a}{k} - \frac{\rho}{\cos\theta}\right) \\ \mathbf{H}(y) \cdot k, & \frac{1}{\tan\theta}\left(\frac{a}{k} - \frac{\rho}{\cos\theta}\right) < y < \frac{1}{\tan\theta}\left(\frac{b}{k} - \frac{\rho}{\cos\theta}\right) \\ b, & y \geq \frac{1}{\tan\theta}\left(\frac{b}{k} - \frac{\rho}{\cos\theta}\right) \end{cases} \tag{5}$$

Next, they compute the line integral images out of the distance images. Matching is done by computing products of line segments of the human model with the corresponding line integral images, for each pixel of the incoming image. To compensate for the change of scale of the model over the image, the distance values are normalized with respect to the maximal distance values, defined by Eq. 5. Figure 3 illustrates the overall FDCM algorithm concept.

In this work we use an identical GPU implementation as in [19]. However, our full human model consists of 13 line segments instead of 16. In addition, in order to handle occlusions, the contour-based likelihood at a given image location $\mathbf{x}$ is computed by matching both head-shoulder (HS) and full-body (FB) templates in a dense scan:

$$\begin{aligned} P(I_c|\mathbf{x}) = & w_1 P_{HS}(I_e|\mathbf{x}, T^*_{HS}(\mathbf{x})) \; + \\ & w_2 P_{FB}(I_e|\mathbf{x}, T^*_{FB}(\mathbf{x})), \end{aligned} \tag{6}$$

where $T^*_{HS}(\mathbf{x})$ and $T^*_{FB}(\mathbf{x})$ denote the locally best matching head-shoulder and full-body templates, w_1 and w_2 are importance weights.

3.3. Motion detection

In this section we briefly describe the computation of the motion cue given by a binary foreground/background segmentation map. The background subtraction algorithm is based on a pixel-based codebook model, as in [21]. The distribution of temporally aggregated pixel intensities is captured by a compressed non-parametric representation, namely a few codewords. A pixel-specific codebook $CB = \{c_1, c_2, \ldots, c_N\}$ of a given pixel contains N codewords, where the i^{th} codeword is denoted by c_i. A codeword $c_i = \left\{\mathbf{v}_i = (\bar{Y}_i, \bar{C}b_i, \bar{C}r_i), S_i\right\}_{i=1\ldots N}$ consists of a vector of three color channels $YCbCr$. S_i is a counter representing the significance of the codeword c_i. For efficiency's sake, the codewords are sorted in descending order with respect to S_i. The sorted order is maintained whenever adding or deleting a codeword.

Initially, each pixel of the background model is initialized with a codeword corresponding to the pixel's $YCbCr$ intensities and the significance value is set to a small value (the learning rate parameter γ_+). During the next frames, intensities of an incoming pixel $\mathbf{u} = \{Y, Cb, Cr\}$ are matched against all codebook entries using the following rule,

$$|\mathbf{u}(1) - \mathbf{v}_i(1)| + |\mathbf{u}(2) - \mathbf{v}_i(2)| + |\mathbf{u}(3) - \mathbf{v}_i(3)| \leq d, \tag{7}$$

where d is a sensitivity threshold. If a match is found, the codeword is updated using an update factor α according to:

$$\mathbf{v}_i(k) = (1-\alpha)\mathbf{v}_i(k) + \alpha\mathbf{u}(k), \quad k = 1, 2, 3. \tag{8}$$

For matching codewords, the significance value S_i is incremented by γ_+, otherwise it is decremented by γ_- (forgetting rate parameter).

The foreground map is generated according to the folowing criterion (see [21]):

$$FG(\mathbf{x}) = \begin{cases} \text{foreground} & \text{if } \sum_{i=i_m}^{N} S_i < T \cdot \sum_{i=1}^{N} S_i \\ \text{background} & \text{otherwise} \end{cases} \tag{9}$$

where T is a parameter defining the width of the distribution's tail. If none of the codewords from a pixel's codebook matches the current RGB triplet of the incoming pixel, a new codeword is added to the codebook. If the number of codewords N exceeds a predefined maximum N_{max}, the codeword with the smallest significance is removed from the codebook to make room for a new codeword. As in [21], the maximal number of allowed codewords per codebook is $N_{max} = 6$, which is sufficient to model outdoor videos.

3.4. Cue computation on the GPU using CUDA

CUDA was used to implement core functions of the contour-based detection scheme and the background subtraction algorithm. GPU with CUDA exhibits performance strength for algorithms or algorithmic parts which can be highly parallelized. Such parts are implemented in a so called computation kernel. Parallel computation kernels were implemented for the shape-based detection carrying out gradient computation and line integral based matching, while kernels for motion detection involved image conversion, computation of the foreground image and the background model update. In general, CUDA takes care of the parallel execution of a thread (an instance of the kernel), where the only thing to be predefined is the number of threads to be processed, split into a block of threads and a grid of blocks. By careful implementation one minimizes data access to global device memory as much as possible, thus avoiding time-costly data transfer.

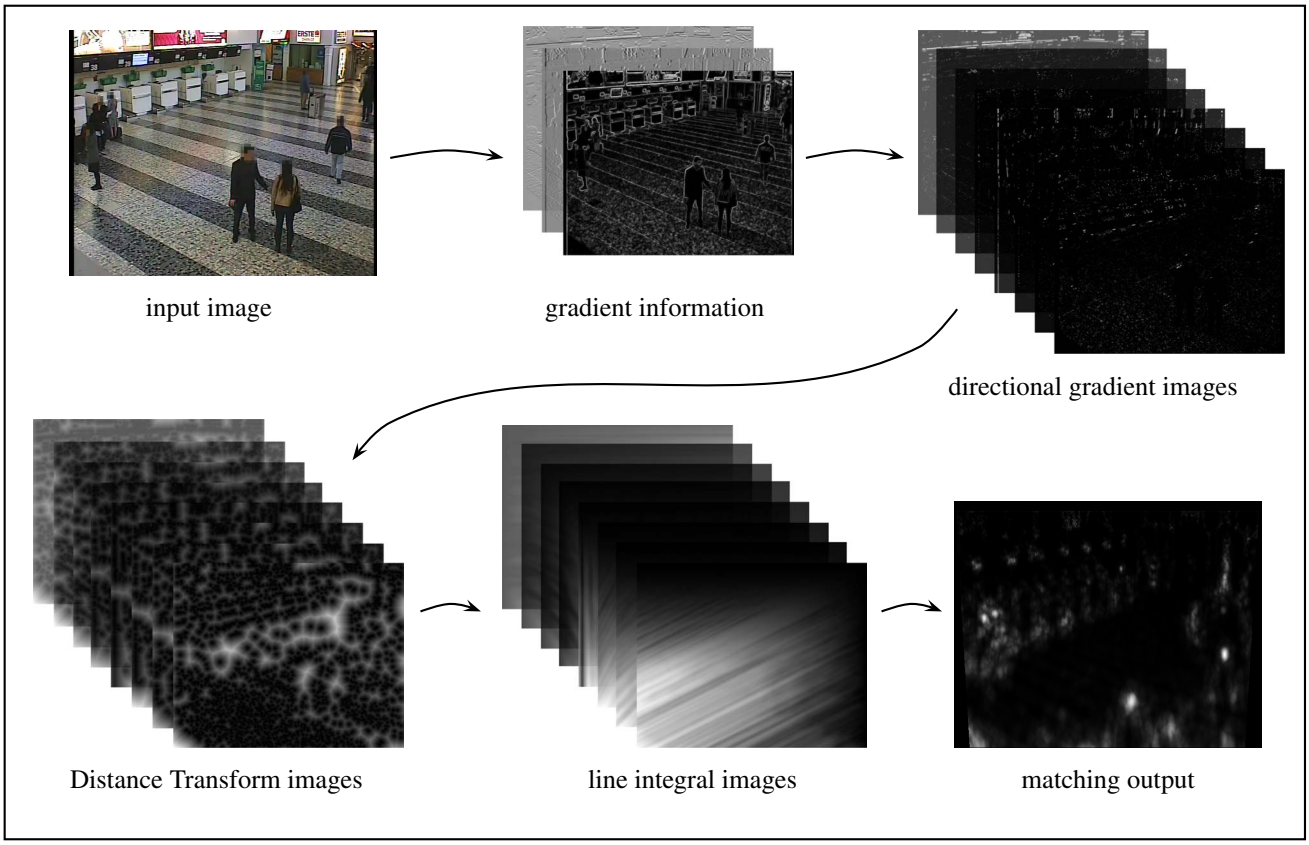

Figure 3. FDCM algorithm overview

Hardware setup: Our test system used for all computations and timing measurements consists of an Intel Xeon CPU with 4 physical and 4 virtual cores @ 2.93 GHz, 12 GB RAM and a NVIDIA GeForce GTX 460 running on Windows7 64-bit.

Cue computation:

The GPU-based parallelization of the Fast Directional Chamfer Matching algorithm was done in [19] for all the algorithmic steps, namely: computation of 8 directional gradient images, computation of the corresponding 8 Distance Transform images, computation of 8 line integral images and the line-based template matching using a variable size human templates. They have also implemented a highly optimized CPU version (via handwritten multi-threading and SSE2), as well as a hybrid CPU-GPU version. Eventually, for typical PC architecture and image resolution used, the hybrid CPU-GPU implementation version turned out to be the fastest. In the hybrid version, the Distance Transform computation on CPU was optimized using multi-threading and inclusion of IPP functions; the line integral computation on CPU was optimized by partly using OpenMP and partly by using hand-written multi-threading.

Gradient computation on GPU: All pixels can be processed independently from each other, leading to a great parallelization potential. However, memory access to global GPU memory is expensive. Therefore, one needs to compute as many arithmetic operations involved in the gradient computation as possible on the fly, and to avoid storing intermediate results in global device memory. Since every *CUDA* thread has its own set of private registers, all intermediate results are stored in GPU registers.

Line-based matching on GPU: The line-based template matching is well suited for parallelization. By using shared memory, memory access time is minimized, since shared memory has much lower latency than global GPU memory. The shared memory is used to store the line-based variable size model templates. This data is shared between threads, since every pixel needs its own copy of model data for the computation of its matching costs. Every thread is loading its slice from the entire model data. Such fraction is commonly the size of a few bytes. By using shared memory, loading times of the model data is reduced. Texture memory is used for the look-ups in the line integral images. This gives a performance increase as long as the scaled line segment models are reasonable small. When line segments are small, the corresponding lookup positions of the line segment's start- and end-point are spatially near. This results in texture memory cache hits giving one lower latency and

thus better performance.

Background subtraction on GPU: For the background subtraction algorithm used, kernels are the image conversion, computation of the foreground image and the background model update. First, image data is transferred from CPU/main memory to GPU. Next, the $YCbCr$ 4:2:0 planar image is converted to $YCbCr$ 4:4:4 interleaved image. The interleaved image has the advantage that data fragments prcessed in successive time steps lie locally near to each other and can be loaded faster. Data access to global device memory for background model data is minimized by loading and storing only the number of currently used codes per pixel. Additional steps performed in the CUDA kernel include loading the current pixel of the background model to shared memory (fastest but limited GPU memory which can contain only part of the image at once), and loading the updated pixel of the background model from shared to global memory. Finally, the foreground image is transferred back to the host from the device. We use an identical GPU implementation as in [21].

3.5. Joint optimization

Following the computation of shape-based and motion-based likelihood images, a joint optimization step is performed, estimating the spatial configuration of humans. This optimization step is performed in a greedy manner, considering the observation likelihoods and the occlusion status of individual hypotheses at the same time.

First we perform non-maxima suppression on the shape-based likelihood image to generate a set of human hypotheses $h = \{(\mathbf{x}_i, C_i)_{i=1...K}\}$ characterized by their foot location and the appropriately scaled contour model. Our optimized non-maxima suppression algorithm is an extension of a recent work reported in [17], presenting two solutions for the non-maxima suppression problem that use fewer than 2 comparisons per pixel with little memory overhead. The first algorithm locates 1-D peaks along the images scan-line and compares each of these peaks against its 2-D neighborhood in a spiral scan order. The second algorithm selects local maximum candidates from the maxima of non-overlapping blocks of one-fourth the neighborhood size. Both algorithms are reported to run considerably faster than previous best methods in the literature when applied to feature point detection. We extend the first solution of [17] by, first, using a rectangular object window rather than a squared one; second, we enable a variable scale object window (due to perspective effects); finally, we detect plateaus in addition to regular peaks. Hypotheses are sorted according to their vertical coordinate component in the image, to generate a plausible depth ordering. Starting with an empty configuration, individual human hypotheses are added incrementally while evaluating shape-based and motion-based likelihoods jointly after each addition step. Given the depth ordering estimate and occlusion status, individual human hypotheses can be evaluated independently. Shape-based likelihoods are evaluated by computing matching probabilities along unoccluded contour segments. Next, these likelihoods are multiplied by motion-based probabilities, computed by measuring the amount of foreground regions covered by human hypotheses forming the configuration. The final configuration estimate is typically reached efficiently given the greedy nature of approximation.

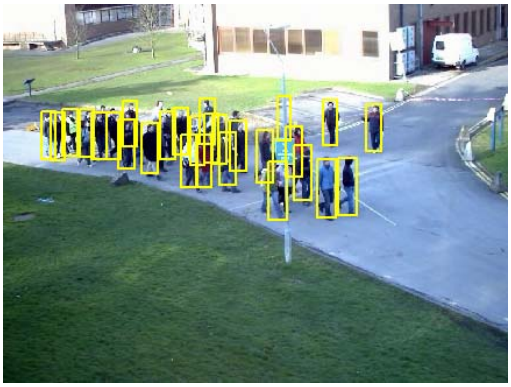

Figure 4. Sample detection results for the most challenging S2-L3 PETS'09 scenario.

4. Experiments and discussion

The proposed contour-based human detection framework employing the Fast Directional Chamfer Matching algorithm was compared to previously published framework [3] using oriented gradient images as cues for shape-based matching. We annotated three sequences (S2-L1, S2-L2, S2-L3) of the PETS 2009 dataset [1], which depict scenarios with an increasing pedestrian density. The annotation consists of the bounding box coordinates of each individual for the entire length of the sequence. Persons at the image borders are annotated when they fully enter the scene.

Evaluation procedure: We denote our proposed detection framework using Chamfer Matching by CM, and the previous work [3] based on Oriented gradient Filters by OF. An examplary video frame with detection results by the CM framework is shown in Figure 4 for the S2-L3 sequence. Spatial bounding box coordinates generated by the CM and OF methods are compared separately to the ground truth data based on an overlap criterion. If two bounding boxes - one generated by the detector, one given by ground truth - have an overlap of more than 50% with respect to their joint area (union), then a potential match is declared. To eliminate multiple matches, a one-to-one match is enforced between all ground truth and detection instances. This latter constraint is especially relevant in crowded scenarios where typically many detection results exhibit an overlap to a given ground truth region at the same time. Based on the bounding box associations, we derive quality measures of *detection rate* and *false alarm rate*,

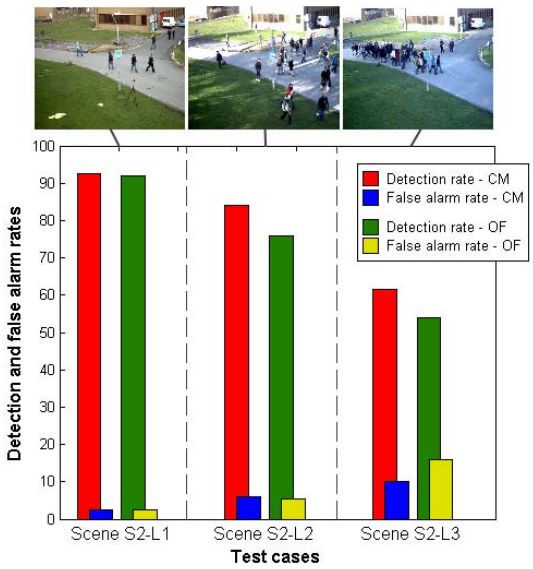

Figure 5. Detection and false alarm rates obtained on three PETS 2009 [1] sequences by comparing the current human detector employing chamfer matching (CM) to previous work [3] which used a set of oriented high-pass filters (OF) for performing shape matching.

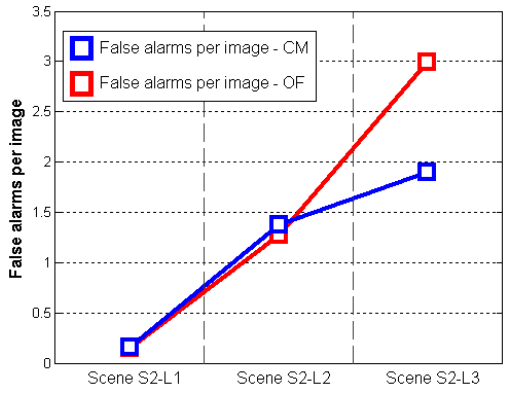

Figure 6. False alarm rates per frame, obtained on three PETS 2009 [1] sequences by comparing the current human detector employing chamfer matching (CM) to [3] which used a set of oriented high-pass filters (OF) for performing shape matching.

equivalent to the measures of true positive and false positive rates when detection is based on discriminative classification.

Results and evaluation: Results obtained for the three sequences are depicted in Figure 5. As can be seen from the plot, the increasing pedestrian density leads to a drop in detection rates and to an increase in false alarm rates. This is due to the increasing partial occlusion rate between the humans in the scene, where the shape matching fails to recover the correct human locations from partially visible contours. The partial visibility of humans was quantified by the ground truth bounding boxes: in the three scenes,

Algorithmic step	Scene S2-L1	Scene S2-L2	Scene S2-L3
Shape matching	23.8 ms	25.6 ms	25.7 ms
Non-maxima suppression	1.5 ms	1.6 ms	1.7 ms
Occlusion analysis	0.1 ms	0.3 ms	0.4 ms
Background model	2.0 ms	2.3 ms	3.3 ms
Motion-based validation	0.1 ms	0.6 ms	0.8 ms
Total	**27.5 ms**	**30.4 ms**	**31.9 ms**

Table 1. Computation time of the individual algorithmic steps, tabulated for our three testing scenarios S2-L1, S2-L2 and S2-L3 of the PETS'09 dataset [1].

3.7%, 17.8% and 41.7% of the body (bounding box) area are occluded, respectively.

When comparing the detection frameworks CM and OF against each other, it becomes apparent that the Chamfer Matching based shape detection can better detect partially occluded persons (red vs. green in Figure 5). Moreover, false alarm rates slightly improve when facing high pedestrian densities (blue vs. yellow in Figure 5). This better sensitivity likely arises from the distance transform's nature, where an oriented segment has a larger spatial spread than a gradient filter response, while well preserving orientation information. Therefore, smaller sets of contour fragments can be matched with a greater precision.

In order to convey a meaningful measure for the amount of false alarms, we compute the amount of *false alarms per image* by dividing the total number of false alarms by the number of frames in the sequence. As can be seen from Figure 6, the per-image false alarm rates are sufficiently low for practical applications (tracking, counting) and comparable to those of competing approaches [7].

In order to assess the computational performance of the proposed CM-based detection scheme, we analyzed the run-times of the individual algorithmic components on the three sequences, as shown in Table 1. The run-times were obtained by averaging over 1000 frames for each sequence. As can be seen from the table, there is only a slight increase in computation time with growing human density. An increasing pedestrian density affects mostly the computation of the distance map due to more gradient structure, while the computational cost of contour integral evaluation remains constant. The other density-sensitive component is the background model computation due to more complex models with increasing density. The overall computation times remain highly competitive, corresponding to approximately 32 *fps* on this dataset.

5. Conclusions

In this paper we presented a GPU-accelerated human detection framework which proposes an efficient computational scheme on CPU-GPU for shape and motion cues, and allows for real-time human detection in challenging scenarios. The detection framework improves a previous framework by performing shape-based matching using the variable scale FDCM algorithm, instead of using oriented high-

pass filters. A comparison between the previous and the current frameworks shows an increase in detection rate as well as a decrease in false alarm rate, especialy in crowded scenarios. Experimental results show that the method is capable of reliably detecting humans in moderately crowded scenarios and it exhibits a slow degradation of detection performance at higher human densities.

References

[1] http://www.cvg.rdg.ac.uk/PETS2009/.

[2] H. G. Barrow, J. M. Tenenbaum, R. C. Bolles, and H. C. Wolf. Parametric correspondence and chamfer matching: two new techniques for image matching. In *Proceedings of the 5th international joint conference on Artificial intelligence - Volume 2*, pages 659–663. Morgan Kaufmann Publishers Inc., 1977.

[3] C. Beleznai and H. Bischof. Fast human detection in crowded scenes by contour integration and local shape estimation. In *CVPR09*, pages 2246–2253, 2009.

[4] C. Beleznai, D. Schreiber, and M. Rauter. Pedestrian detection using gpu-accelerated multiple cue computation. In *Embedded Computer Vision Workshop 2011*, pages 58–65, 2011.

[5] F. Crow. Summed-area tables for texture mapping. In *Proceedings of SIGGRAPH*, volume 18, pages 207–212, 1984.

[6] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, volume I, pages 886–893, 2005.

[7] P. Dollár, R. Appel, and W. Kienzle. Crosstalk cascades for frame-rate pedestrian detection. In *ECCV*, 2012.

[8] P. Dollár, W. C, B. B. Schiele, and P. Perona. Pedestrian detection: A benchmark. In *CVPR*, pages 1–8, 2009.

[9] A. Elgammal, D. Harwood, and L. Davis. Non-parametric model for background subtraction. In *Proc. ECCV*, volume II, pages 751–767, 2000.

[10] M. Enzweiler and D. M. Gavrila. Monocular Pedestrian Detection: Survey and Experiments. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(12):2179–2195, Oct. 2008.

[11] D. Gavrila and V. Philomin. Real-time object detection for "smart" vehicles. In *ICCV*, pages 87–93, 1999.

[12] M. Hussein, F. Porikli, and L. Davis. A comprehensive evaluation framework and a comparative study for human detectors. *Trans. Intell. Transport. Sys.*, 10:417–427, September 2009.

[13] K. Kim, T. H. Chalidabhongse, D. Harwood, and L. Davis. Real-time foreground-background segmentation using codebook model. *Real-Time Imaging*, 11:172–185, June 2005.

[14] B. Leibe, E. Seemann, and B. Schiele. Pedestrian detection in crowded scenes. In *CVPR*, volume 1, pages 878–885, 2005.

[15] Z. Lin, L. S. Davis, and D. Doermann. Hierarchical part-template matching for human detection and segmentation. In *ICCV*, pages 1–8, 2007.

[16] M.-Y. Liu, O. Tuzel, A. Veeraraghavan, and R. Chellappa. Fast directional chamfer matching. In *CVPR*, pages 1696–1703. IEEE Computer Society, 2010.

[17] T. Pham. Non-maximum suppression using fewer than two comparisons per pixel. In *Advanced Concepts for Intelligent Vision Systems*, volume 6474, pages 438–451. Springer Berlin Heidelberg, 2010.

[18] F. Porikli, P. Meer, and O. Tuzel. Human detection via classification on riemannian manifolds. In *CVPR*, pages 1–8, 2007.

[19] M. Rauter and D. Schreiber. A GPU accelerated fast directional chamfer matching algorithm and a detailed comparison with a highly optimized cpu implementation. In *Embedded Computer Vision Workshop 2012*, pages 68–75, June.

[20] M. D. Rodriguez and M. Shah. Detecting and segmenting humans in crowded scenes. In *MULTIMEDIA '07: Proceedings of the 15th international conference on Multimedia*, pages 353–356, 2007.

[21] D. Schreiber and M. Rauter. GPU-based non-parametric background subtraction for a practical surveillance system. In *Embedded Computer Vision Workshop 2009*, pages 870–877, 2009.

[22] P. Simard, L. Bottou, P. Haffner, and Y. L. Cun. *Boxlets: a fast convolution algorithm for signal processing and neural networks*, volume 11, pages 571–577. Advances in Neural Information, Eds. M. Kearns, S. Solla, and D. Cohn, MIT Press, 1999.

[23] C. Stauffer and W. E. L. Grimson. Adaptive background mixture models for real-time tracking. In *CVPR*, volume 2, page 2246, 1999.

[24] A. Thayananthan, B. Stenger, P. H. S. Torr, and R. Cipolla. Shape context and chamfer matching in cluttered scenes. In *CVPR*, CVPR'03, pages 127–133, Washington, DC, USA, 2003. IEEE Computer Society.

[25] O. Tuzel, F. Porikli, and P. Meer. Region covariance: A fast descriptor for detection and classification. In *ECCV*, pages 589–600, 2006.

[26] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *CVPR*, volume 1, pages 511–518, 2001.

[27] X. Wang, G. Doretto, T. B. Sebastian, J. Rittscher, and P. H. Tu. Shape and appearance context modeling. In *ICCV*, pages 1–8, 2007.

[28] B. Wu and R. Nevatia. Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet based part detectors. *Int. J. Comput. Vision*, 75(2):247–266, 2007.

[29] L. Zhao and L. S. Davis. Closely coupled object detection and segmentation. In *ICCV*, pages 454–461, 2005.

[30] T. Zhao and R. Nevatia. Bayesian human segmentation in crowded situations. In *CVPR*, volume 2, pages 459–466, 2003.