

# MULTIPLE CAMERA-BASED CHAMFER MATCHING FOR PEDESTRIAN DETECTION

*Itai Katz, Hamid Aghajan*

Stanford University, Stanford, CA  
{itai, aghajan}@stanford.edu

## ABSTRACT

This paper presents a vision system for detecting pedestrians using chamfer matching. We verify the effectiveness of chamfer matching for single cameras and propose a novel method for combining results from multiple views. A key insight is that making independent decisions in each camera and combining them in a higher level is prone to error. By communicating during the template matching stage, camera nodes can avoid making hard decisions. Incorporating observations from multiple cameras should in theory reduce detection error. Additionally, we provide a conceptually straightforward algorithm for building a database that maximizes the space of poses in a minimum number of templates which results in real-time performance.

*Index Terms*— pedestrian detection, chamfer matching, camera networks

## 1. INTRODUCTION

The ability to reliably localize pedestrians in a scene is a long-standing problem in computer vision. The existence of such an algorithm would lead to a number of commercially interesting applications. Automated pedestrian detection naturally extends systems that currently require a human operator (e.g., surveillance and security) and enables applications that have no manual analogue (e.g., smart environments). A working implementation has remained elusive because the human body has high intra-class variance: shape, color and size can vary widely between individuals. In contrast to general object detectors [12], pedestrians share few commonalities.

In this paper we verify the performance of chamfer matching-based pedestrian detection, and describe an extension to a multiple camera implementation. The paper is structured as follows. Section 2 outlines existing work. Section 3 describes chamfer matching for single cameras. Section 4 presents our proposed extension to multiple cameras, and Section 5 concludes with experimental results.

## 2. PREVIOUS WORK

The literature on pedestrian detection is vast. While a thorough examination of related work is beyond the scope of this paper, we describe a few notable approaches below. Interested readers are directed to surveys in [6, 8].

Viola, et. al. [13] presents a feature-based detector that uses a cascade of simple, haar wavelet-like classifiers. In contrast to other sliding window techniques, the cascaded approach quickly eliminates unlikely candidate regions. By combining appearance and motion descriptors a computationally efficient system is achieved with a very low false positive rate. Due to the large number of potential descriptors, however, a very large training set (12,000 frames) is required consisting of manually labeled pedestrians.

Higher-level approaches detect humans by identifying body parts. [9] learns co-occurrences of local features for each body part. The output is improved by assigning a likelihood score to the relative positions of the detected parts. This accounts for the gross structure of the body without enforcing rigid pose constraints. An earlier method by Mohan, et. al, takes a similar approach, but requires that all body parts (left arm, right arm, legs, and face) be visible [10].

Full-body detection has also been attempted using shape-based techniques such as chamfer matching [5, 7]. A query image is first converted into an edge map before being matched against a database of human silhouettes. Chamfer matching is appealing for pedestrian detection since it relied on edges which are invariant to color or lighting variation. The human body is highly non-rigid but in practice only a few silhouettes are necessary to cover the majority of common poses (section 3). Although the initial concept is not recent [1,2], its application to pedestrian detection has only been explored minimally.

Our primary contribution is to describe a method for minimizing the database size in chamfer-based applications. Previous implementations rely on databases consisting of thousands of templates; in constrained domains we show that far fewer templates are necessary, ensuring real-time operating even when multiple views are processed simultaneously. We also extend the existing work to use

multiple cameras in a collaborative manner in order to improve detection accuracy.

### 3. SINGLE CAMERA CHAMFER MATCHING

Chamfer matching is a special case of template matching. The templates in this case are binary arrays which take on the value 1 at a silhouette edge and 0 elsewhere. The template  $t$  is shifted over an edge mapped-image  $E$  and for each position the distance metric  $D_{chamfer}$  is computed, which is defined at location  $l$  as:

$$D_{chamfer}(t, \ell) = \frac{1}{|t|} \sum_{k \in t} DT(k + \ell) \quad (1)$$

where distance transform  $DT$  is an image that, for each pixel, that contains the distance to the nearest edge pixel in  $E$ . Since  $DT$  takes the value 0 on edges, the lowest theoretical value for  $D_{chamfer}$  is 0 and occurs when a template perfectly matches a patch in the  $E$ . The optimum template values for  $t$  and  $l$  are given by iterating over all pairs:

$$(t_0, \ell_0) = \arg \min_{t \in T, \ell \in L} D_{chamfer}(t, \ell) \quad (2)$$

for template database  $T$  and the set of all pixel positions  $L$ .

The distance transform step has a number of benefits. Matching templates against a distance transformed image instead of an edge image makes  $D_{chamfer}$  a smooth function when evaluated at each position  $l$ . This was originally intended to enable gradient descent optimization to speed up the process, although modern hardware has mitigated this need. The smoothness property is also useful for identifying shapes that are close, but not identical to the template being matched. This aspect makes chamfer matching robust to partial occlusions, and reduces sensitivity to the parameters used in edge mapping the input image.

Besides the aforementioned lighting and color invariance, chamfer matching has a number of qualities that make it well suited for our application. Using edge maps instead of pixel intensity obviates the need for an error-prone background subtraction step and its associated parameter tweaking. Since the templates are binary arrays, matching them against the distance transformed query images can be performed with a cheap AND operation. Lastly, having a silhouette outlining the detected pedestrian (as opposed to say, a bounding box) can potentially enable 3D reconstruction using visual hull construction [4].

Regions with high edge densities can result in spurious matches. Since chamfer matching seeks out similarly-shaped silhouettes, any highly textured region with high edge density could, as a subset of its edges, contain the silhouette in question (see figure 1). In the example given, the spurious detection is due to a background object. Background

subtraction could effectively eliminate these false detections, but would be ineffective against textured moving objects. An alternative approach is to consider *oriented* edges [11]. Oriented edges extend the traditional template by considering each edge pixel as vector  $[p_i \ p_j \ p_\theta]^T$ , where  $i$  and  $j$  are pixel coordinates and  $\theta$  is orientation. The resulting pixels are put into a histogram where each bin represents a range of orientations. An example of the resulting set of templates is shown in figure 2.

The edge mapped query image is binned in a similar manner. Matching proceeds as in (1), but each template bin is matched against the corresponding image bin and the resulting scores for all bins are summed.



Figure 1. Highly textured image regions can result in spurious detection.

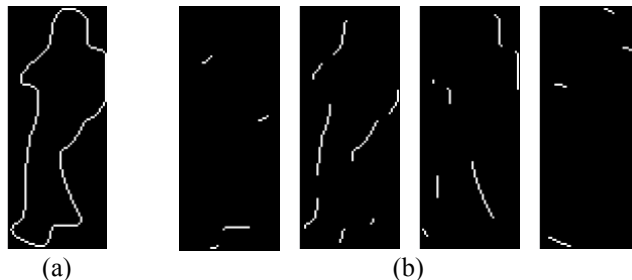


Figure 2. (a) original template (b) edge oriented templates (from l to r: 0-45 degrees, 46-90, 91-135, 136-180)

Little attention has been given to the selection of templates to be included in the database. The common approach calls for including a large number of templates ( $\approx 10^3$  examples) to comprehensively describe the range of possible human poses. Since matching thousands of templates against an image is impractical, the authors of [5] cluster similar templates and arrange them in a tree, with the most generic silhouette at the root and the most detailed at the leaves. Empirically, we found that for constrained domains (e.g., indoor settings), only a small set of templates is needed to describe the most likely poses (Section 5). Gaps in detection caused by an incomplete database can be compensated for by tracking with an appropriate motion model [3].

We propose a greedy algorithm for constructing a template database from a pool of candidate templates. First,  $n$  candidates are collected by manually segmenting

pedestrian silhouettes from a sequence of frames  $F$ . Each candidate is then matched against all frames in  $F$  and the best performing candidate across all frames (measured by spatial overlap (see Section 5) is added to the database  $T$ . This process repeats, matching each frame against the current  $T$  and each of the remaining candidate templates, adding one template to  $T$  at each iteration until an accuracy threshold is reached. Details are given below.

---

**Algorithm 1** Database construction for single camera

---

1: **Initialization**

$T = \{ \}$   
 $T_c = \{t_i, i \in 1..n \}$   
 $p = 0$ ;

2: **Reduction**

Repeat until  $p > p_{\text{thresh}}$   
 $t_0 = \arg \max_{t \in T_c} \text{MeanSpatialOverlap}(\{T, t\}, F)$   
 $T \leftarrow T + \{t_0\}$   
 $T_c \leftarrow T_c - \{t_0\}$   
 $p = \text{MeanSpatialOverlap}(T, F)$

3: return  $T$

---

**4. MULTIPLE CAMERA CHAMFER MATCHING**

Chamfer matching using a single camera has shown promising performance, but limitations persist. As with any monocular system, tracking is lost when the object of interest is occluded. Although tracking could be reestablished by incorporating learned appearance and motion models, the problem of occlusion becomes increasingly severe as the number of observed objects increases. This is particularly true of tracking pedestrians in enclosed environments.

A key insight in our application is that the observations made by each camera are not independent. Since the cameras are simultaneously viewing the same pose (albeit from different angles), their observations are correlated. This notion suggests a mechanism to guide template matching for multiple cameras. Rather than treat each template database as a separate entity and return the optimum template for each camera, we can enforce a template ordering such that the  $i$ 'th template in each database corresponds to the same pose. Matching then proceeds as before, but the optimum template is given by the entry that gives the lowest error *summed across all databases*. If we define a pose  $p$  as an array of templates, where each template  $t_c^p$  corresponds to camera  $c$ 's view of  $p$ , the error for  $p$  is given by

$$Error(p) = \sum_c \min_{\ell \in L} [D_{\text{chamfer}}(t_c^p, \ell)] \quad (3)$$

The optimum template is a vector given by the pose that minimizes the error:

$$t^p \leftarrow \arg \min_p [Error(p)] \quad (4)$$

The difference between matching methods is depicted schematically in figure 3. This scheme has two benefits over a locally optimum search:

- 1) By providing a robust method for combining different cues, we avoid making a hard decision in each camera, where a failure in one can result in significant localization error.
- 2) Each camera maintains a different set of templates in its database. This accounts for different geometries without requiring a large, comprehensive set of templates from every potential view.

The real-time communication requirement is minimal: since, for each template, each camera only transmits a vector of error values (one for each template) and not image data, this method remains practical as the system scales up.

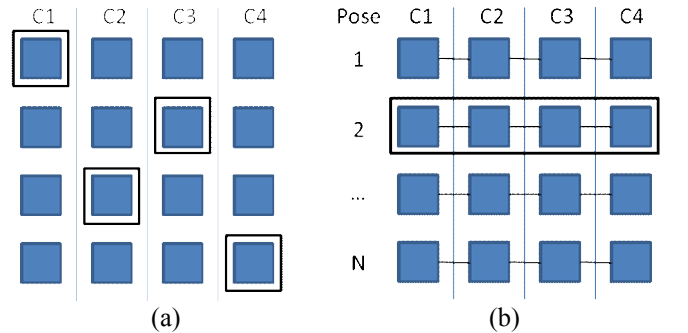


Figure 3. Chamfer matching for multiple cameras. (a) depicts a simple extension from the single camera implementation. Each camera returns the locally optimal template. The proposed method in (b) orders the database entries by pose and returns a set of globally optimal templates.

**5. EXPERIMENTAL RESULTS**

In this section we demonstrate the performance benefits Algorithm 1. Section 5.1 describes the error metrics used and Section 5.2 presents results of the database construction technique and single camera-based chamfer matching. The data were collected from 2 off-the-shelf IP cameras simultaneously viewing a scene of a pedestrian in an indoor setting. Sequence 1 depicts a side view of a pedestrian entering and exiting a classroom. Sequence 2 shows the

simultaneous scene from a  $\frac{3}{4}$  profile. Templates for both sequences were taken from a set of 170 frames and were tested on a set of 340 frames. Ground truth was collected by manually labeling key frames and smoothly interpolating the intermediate frames.

### 5.1 Evaluation criteria

Two criteria are used to evaluate detector performance: spatial overlap (SO) and false positive rate (FPR). Following [14], SO takes into account the bounding box of the ground truth and the detected region,  $B_g$  and  $B_d$ , respectively. It is defined as the ratio between the areas of the intersection and union of  $B_g$  and  $B_d$ , or:

$$SO = \frac{\sum_i B_g \cap B_d}{\sum_i B_g \cup B_d} \quad (3)$$

This metric has the convenient property of ranging continuously between 1 (with perfect overlap) and 0 (with no overlap, regardless of the distance between bounding boxes). The mean spatial overlap is given by the average spatial overlap for all frames in the sequence. The false positive rate (FPR) is used to measure the frequency that detection fails. FPR is defined for a sequence of frames  $F$  as:

$$FPR = \frac{1}{|F|} \sum_{f \in F} (SO(f) < 0.5) \quad (4)$$

### 5.2 Single camera results

Figure 4 presents the quantitative results for pedestrian tracking from a single camera. Twenty silhouettes were manually extracted from each training sequence. These databases were then reordered with Algorithm 1 in order of importance. The pedestrian detector was then evaluated on the testing sequences using databases of size  $n$  templates (where  $n$  ranged from 1 to 20). The results show that detector performance quickly levels off with a database of only a few templates. It should be noted that performance *decreases* slightly when the least relevant templates are included. This is the result of poor or erroneous templates matching background elements.

From the results it can be seen that Algorithm 1 gives an upper limit on the number of templates needed for a particular domain. Since extracting silhouettes is a labor-intensive process, having an upper limit could significantly ease setup time. Since the tracker speed is linear in the number of templates, a smaller database is proportionally faster. By arranging the templates in order of importance the erroneous templates (those at the end of the ordered database) can be discarded. When run on the 5 best

templates, Sequence 1 gave a mean spatial overlap of 61.5% and a false positive rate of 3.50%. Sequence 2 gave a mean spatial overlap 63.3% and a false positive rate of 4.39%. Examples of the typical tracking results are shown in Figure 5.

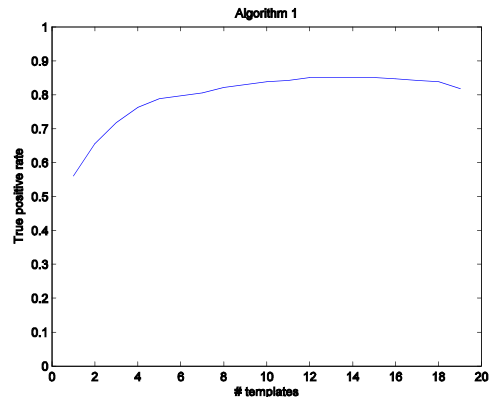


Figure 4. As the number of templates in the database increases, performance increases up to a point. Sequence 2 was used to generate this graph.



Figure 5. Examples of typical tracking performance from sequence 1 (a,b) and sequence 2 (c,d).

## 7. CONCLUSIONS

In this paper we reviewed the existing approach to pedestrian detection using chamfer matching and discussed practical considerations. The two primary contributions are the extension of this work to multiple cameras, and an algorithm for building a template database.

We tested the latter on a pair sequences and verified the performance of chamfer matching for pedestrian detection. It was demonstrated that for domains where only limited behaviors are expected, relatively few templates are necessary to adequately describe most poses. We proposed that incorporating a network of cameras could substantially improve detection accuracy.

Future work will incorporate the multi-camera framework with templates pre-segmented with body part positions. This will be applied towards higher-level behavior identification for use in smart environments.

## 8. REFERENCES

- [1] H. G. Barrow, et. al., "Parametric correspondence and chamfer matching: two new techniques for image matching," Technical note, *SRI International*, 1977.
- [2] G. Borgefors, "Hierarchical chamfer matching: a parametric edge matching algorithm," *Trans. PAMI*, vol. 10, pp. 849-865, November 1988.
- [3]. D. Comaniciu, V. Rameesh, and P. Meer, "Kernel-based object tracking," *Trans. PAMI*, vol 25, May 2003.
- [4] K.M.G. Cheung, S. Baker, and T. Kanade, "Shape-from-silhouette of articulated objects and its use for human body kinematics estimation and motion capture," In *CVPR'03*, 2003, pp. 77-84.
- [5] D.M. Gavrila, "Pedestrian detection from a moving vehicle," In *ECCV'00*, 2000, pp. 37-49.
- [6] D.M. Gavrila, "Sensor-based pedestrian protection," *IEEE Intelligent Systems*, vol. 16, pp. 77-81, November 2001.
- [7] D.M. Gavrila and J. Giebel, "Shape-based pedestrian detection and tracking," In *IEEE Intelligent Vehicle Symposium*, 2002, pp. 8-14.
- [8] D. Geronimo, A Lopez, and A. Sappa. "Computer vision approaches to pedestrian detection: visible spectrum survey," *Pattern Recognition and Image Analysis*, pp. 547-554, 2007.
- [9] C. Mikolajczyk, C. Schmid, and A. Zisserman. "Human detection based on a probabilistic assembly of robust part detectors," In *ECCV'04*, 2004, pp. 69-82.
- [10] A. Mohan, C. Papageorgiou, and T. Poggio, "Example-based object detection in images by components," *Trans. PAMI*, vol. 23, pp. 349-361, April 2001.
- [11] C.F. Olson, D.P. Huttenlocher, "Automatic target recognition by matching oriented edge pixels," *Trans. Image Processing*, vol. 6, January 1997.
- [12] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *CVPR'01*, 2001, pp. 511-518.
- [13] P. Viola, M. Jones, and D. Snow, "Detecting pedestrians using patterns of motion and appearance," in *ICCV'03*, pp. 734-741.
- [14] F. Yin, D. Makris, and S.A. Velastin, "Performance evaluation of object tracking algorithms," in *Performance Evaluation of Tracking and Surveillance*, October 2007.