

# Learning Discriminative Chamfer Regularization

Pradeep Yarlagadda \*  
pradeep.yarлага@iwr.uni-heidelberg.de

Angela Eigenstetter \*  
aeigenst@iwr.uni-heidelberg.de

Björn Ommer  
ommer@uni-heidelberg.de

Interdisciplinary Center for Scientific  
Computing (IWR)  
University of Heidelberg  
Germany

---

## Abstract

Chamfer matching is an effective and widely used technique for detecting objects or parts thereof by their shape. However, it treats objects as being a mere sum of the distance transformation of all their contour pixels, thus leading to spurious matches. We take account of the fact that boundary pixels are not all equally important by applying a discriminative approach to chamfer distance computation, thereby increasing its robustness. While this is improving the behavior in the foreground, chamfer matching is still prone to accidental responses in spurious background clutter. To estimate the accidentalness of a match, a small dictionary of simple background contours is utilized. These background elements are trained to focus at locations where, relative to the foreground, typically accidental matches occur. Finally, a max-margin classifier is employed to learn the co-placement of all background contours and the foreground template. Our approach is easily integrated into an off-the-shelf directional chamfer matching approach and it shows significant improvements over state-of-the-art chamfer matching on standard benchmark datasets.

## 1 Introduction

Chamfer matching is a well established and widely used technique for registration and detection of whole objects and contour segments. Due to its simplicity and efficiency, numerous application areas have benefited from chamfer matching. However, a serious limitation is its susceptibility to background clutter. Although the inclusion of orientation information [14, 19] has improved the specificity, performance is still seriously affected by clutter. The primary reason for this is that the presence of individual model points in a query image is measured independently. A match with the object model is then represented by the sum of all the individual model point distance transformations. Consequently, i) all object pixels are treated as being independent and equally relevant, and ii) the model contour (the foreground) is prone to accidental matches with background clutter. As demonstrated by Biederman [5], Attneave [2], and various experiments on illusionary contours, object boundary pixels are

not all equally important due to their statistical interdependence. Moreover, in dense background clutter the points on the model have a high likelihood to find good spurious matches [10, 5]. However, any arbitrary model would match to such a cluttered region, which consequently gives rise to matches with high accidentalness. Chamfer matching only matches the template contour and thus fails to discount the matching score by the accidentalness, i.e., the likelihood that this is a spurious match. To improve the robustness of model matching, we learn the co-occurrence of model points (or rather their matches). To reduce the accidentalness of chamfer matching, we learn a flexible co-placement of generic background contours. Both these contributions are combined into a single discriminative learning algorithm. Our approach is built upon the publicly available, state-of-the-art directional chamfer matching approach [14] and we evaluate the proposed method on standard benchmark datasets for chamfer matching.

## 2 Related Work

Approaches for object recognition can be divided into voting based approaches, e.g. [11, 12, 13, 20] and sliding window based approaches. In our work we focus on sliding window methods based on Chamfer matching which has been used in a large number of applications in computer vision. It was first introduced by Barrow et al. [9] to match two sets of contour fragments. Since that time until today chamfer matching is a widely applied and successful technique detecting complete objects or their parts. In [6] hierarchical chamfer matching was suggested where edge points are matched in a coarse-to-fine-manner. Later, chamfer matching was used to build powerful detectors as suggested in [10, 12, 13]. Leibe et al. [12] combine local features with global shape cues obtained from chamfer matching to verify and refine hypotheses. In [11] Gavrilu and Munder have applied, chamfer matching for real-time pedestrian detection and tracking. Lin et al. [13] have proposed a hierarchical part-template tree that matches part templates using shape information based on chamfer matching scores.

In [20] Thayananthan et al. have compared shape context [9] and chamfer matching of templates for object detection in cluttered images. They report that chamfer matching is more robust in clutter than shape context. Nevertheless, false positives in cluttered background were found to be the major downside of chamfer matching. More recent research has made attempts to address this problem. Shotton et al. [19] suggested an improved matching scheme called oriented chamfer matching (OCM) that takes into account the orientation mismatch between pixels. In [14] an alternative approach for incorporating edge orientation has been proposed which solves the matching problem in an augmented space. It was shown that the suggested directional chamfer matching (DCM) achieves a superior performance compared to oriented chamfer matching. Another improvement was suggested in [15] where manually specified tuples of contour fragments have been used as normalizers for oriented chamfer matching.

[14, 19] focus on adding orientation information to improve the matching quality of the foreground template. In both approaches an object hypothesis is represented as the sum of its distance transformation costs under additional consideration of pixel orientations. However, an object is more than a mere sum of the deformation of individual pixels, i.e. considering for instance the fact that not all boundary pixels are equally important. We improve the matching performance of directional chamfer matching by learning co-occurrences of boundary pixels and background contours to represent the object and distinguish it from clutter. In [15] a coarse attempt was made to model background by normalizing template matches with manually combined normalizer contours with fixed location at the center of



Figure 1: Pixel weights learned in a discriminative max-margin framework for various shape templates are visualized here. The pixels are weighted relative to the template and therefore are not comparable among different object classes. Red indicates high and blue low weight.

the object. However, to sufficiently model complex background it is important to combine simple contours in a more robust way and with flexible placement going beyond the manual combinations of normalizers. Motivated by that, we measure the accidentalness of a match, by learning the co-placement of background contours dependent on the foreground, which enables our approach to handle complex background patterns. Our final detection system improves the matching performance of the foreground template while suppressing difficult spurious matches in cluttered background by the proposed background regularization.

### 3 Max-Margin Chamfer Regularization

We base our study in this paper on the recently proposed improved fast directional chamfer approach [14]. The method by Liu et al. [14] achieves state-of-the-art performance in chamfer-based matching and it is publicly available, thus enabling our extension to be easily applicable. Let us now briefly review the fast directional chamfer matching [14] and introduce the required notation. Let  $T = \{\mathbf{t}_i\}$  and  $Q = \{\mathbf{q}_j\}$  be the sets of template and query edge map respectively. Let  $\phi(\mathbf{t}_i)$  denote the edge orientation of the edge point  $\mathbf{t}_i$ .

For a given location  $\mathbf{x}$  of the template in the query image, directional chamfer matching aims to find the best  $\mathbf{q}_j \in Q$  for each  $\mathbf{t}_i \in T$  by minimizing the cost  $|\mathbf{t}_i + \mathbf{x} - \mathbf{q}_j| + \lambda |\phi(\mathbf{t}_i + \mathbf{x}) - \phi(\mathbf{q}_j)|$ .  $\lambda$  denotes the weighting factor between location and orientation terms. Thus the directional chamfer distance for placing the template at location  $\mathbf{x}$  is defined as

$$d_{DCM}^{(T,Q)}(\mathbf{x}) = \frac{1}{|T|} \sum_{\mathbf{t}_i \in T} \min_{\mathbf{q}_j \in Q} |\mathbf{t}_i + \mathbf{x} - \mathbf{q}_j| + \lambda |\phi(\mathbf{t}_i + \mathbf{x}) - \phi(\mathbf{q}_j)| \quad (1)$$

where  $\lambda$  denotes the weighting factor between location and orientation terms.

#### 3.1 Learning the Relevance of Model Points

Not all the pixels on the shape template are equally important for detecting objects. Consider for instance the famous Kanizsa triangle. Provided only contour fragments around the corners, the whole triangle can be easily recognized. Similarly, Biederman [5] presents perceptual experiments with degraded contours that demonstrate the varying importance of different points on object contours. Another example is Attneave’s cat [2], where for instance, points of high curvature are proposed as the most useful features for recognition. However, we do want to automatically learn, which parts of the model are important, rather than manually encoding a set of rules that define the importance of contour points.

In chamfer matching, matching costs for a template are obtained by summing over all the template pixels in the distance transform of the query image as in (1). Thus, all the pixels are implicitly considered to be equally important when computing the matching costs. To take

into account the fact that not all pixels are equally important, we learn discriminative weights for the co-occurrence of individual template points, i.e., of their matching costs  $p_i^{(T,Q)}(\mathbf{x})$ ,

$$p_i^{(T,Q)}(\mathbf{x}) = \min_{\mathbf{q}_j \in Q} |(\mathbf{t}_i + \mathbf{x}) - \mathbf{q}_j| + \lambda |\phi(\mathbf{t}_i + \mathbf{x}) - \phi(\mathbf{q}_j)| \quad (2)$$

Adjacent template pixels are statistically dependent and, thus, we do average (2) over the direct neighbors of pixel  $i$ . The resulting  $\bar{p}_i$  are then used to learn the importance of contour pixels. The discriminative learning algorithm that discovers the weights for the co-occurrences of pixels is described in Sect. 3.3. For visualization purpose, we learned the importance of each pixel using a linear SVM and display the resulting weights for various shape templates in Fig. 1.

## 3.2 Using Background Contours to Model Accidentalness

Chamfer matching is notoriously prone to spurious matches in background clutter. Although adding orientation information [24, 19] and learning the relevance of foreground pixels increase the specificity of the approach, they fail to eliminate false positives in intense clutter (for an example see Fig. 5). Consequently we need to measure the accidentalness of a match. We use a codebook of simple, generic contour segments, which obviously feature a low specificity and high accidentalness. To obtain the set of simple contour segments we collect differently oriented straight and curved lines (see Fig. 2 a)). These simple contours will be called background contours  $T_{bg}$  in the following. As a negative side effect these background contours will, however, also respond to the foreground object. To make up for the lack of specificity of individual contours we learn discriminative co-occurrence patterns of all of these background contours. These co-occurrence patterns identify matches to clutter and distinguish them from actual foreground matches. In contrast to [15], who manually combine tuples of normalizers consisting of one or two contours to form hand designed complex background templates, we propose to automatically learn flexible arrangements of all the background contours to improve detection accuracy.

False positives occurring in background clutter are caused by the edges in the query image at the locations where the foreground contour is placed. Consider a U-shaped template being matched to a query image. Clutter from the query image that is situated within the U does not interfere with the template. Only clutter that is close to the contour of the U will have an impact. Therefore, we need to check for spurious background contours in the neighborhood of model contours, but not elsewhere. In contrast to this, [15] place background contours at a fixed single location, i.e., at the center of the model contour, thereby not measuring the susceptibility of the model contour to clutter. Rather than measuring the amount of clutter on the template contour where it actually matters, they check for clutter simply at the center of the object. To measure where clutter typically interferes with the model contour we compute the directional chamfer matching score  $d_{DCM}^{(T_{bg},T)}$  between each background contour and the object template. We consider placements of the background contour with better (lower) chamfer matching score to be more important since they occur on or close to the model contour. In order to weight these matching locations higher we create a mask

$$M^{(T_{bg},T)}(\mathbf{x}) = 1 - d_{DCM}^{(T_{bg},T)}(\mathbf{x}) \quad (3)$$

from the directional chamfer matching scores. Each combination of a foreground template and a background contour results in a different mask. Fig. 2 shows examples of these masks for different background contours. One can see that high weight is assigned where the background contour matches well to the foreground contour and low weight otherwise. Therefore

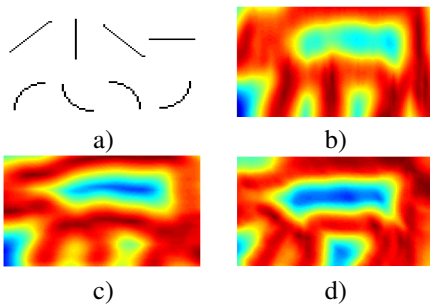


Figure 2: A set of simple background contours  $T_{bg}$  is shown in a). These background contours were placed relative to the cow shape mask shown in Fig. 1 to create masks described in (3). b)-d) show the resultant masks. b) shows the mask for the vertical line, c) shows the mask for the horizontal line and d) shows the mask for arc 3 in the second row of panel a). Red indicates high weight and blue indicates low weight.

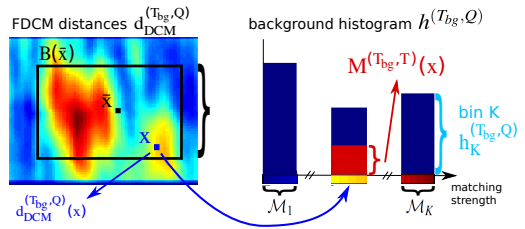


Figure 3: Construction of weighted background histograms from fast directional chamfer matching score maps, see (4). In the score map on the left, red indicates high matching score and blue indicates low matching score of the background template in the query image. For a bounding box region  $B(\bar{\mathbf{x}})$  centered at  $\bar{\mathbf{x}}$ , each directional chamfer matching score  $d_{DCM}^{(T_{bg}, Q)}(\mathbf{x})$  is assigned to its corresponding histogram bin range  $\mathcal{M}_k$  and casts a vote with weight  $M^{(T_{bg}, T)}(\mathbf{x})$  (see (3) and Fig. 2) to this bin.

matches of background contours inside the object are less important than those on the object boundary.

To describe the background matching costs for a hypothesis in a robust way we are building weighted histograms over chamfer matching scores  $d_{DCM}^{(T_{bg}, Q)}$  obtained from matching a background contour  $T_{bg}$  with the query image  $Q$ . Let  $B(\bar{\mathbf{x}})$  be the bounding box region with center  $\bar{\mathbf{x}}$  for a specific placement of the foreground template  $T$  in the query image  $Q$  (see Fig. 3). For each foreground hypothesis we build weighted histograms  $h^{(T_{bg}, Q)}$  over the directional chamfer matching scores  $d_{DCM}^{(T_{bg}, Q)}$  in the corresponding bounding box region. The weights introduced in (3) are used to weight the histogram votes. Therefore chamfer matching scores  $d_{DCM}^{(T_{bg}, Q)}$  are weighted according to their position relative to the foreground template. Each histogram consists of  $K$  bins where  $\mathcal{M}_k$  is the range of the  $k$ th bin and  $k = 1, \dots, K$ . We define a histogram bin  $h_k^{(T_{bg}, Q)}$  as

$$h_k^{(T_{bg}, Q)} = \sum_{\substack{\mathbf{x} \in B(\bar{\mathbf{x}}) \\ d_{DCM}^{(T_{bg}, Q)}(\mathbf{x}) \in \mathcal{M}_k}} M^{(T_{bg}, T)}(\mathbf{x}), \quad (4)$$

for each background contour  $T_{bg}$  on a certain position of the foreground template  $T$  in the query image  $Q$  (see Fig. 3).

### 3.3 Learning Chamfer Regularization

From above we know that we need to model the co-occurrence of all template points. Moreover, a codebook of simple generic contours needs to be matched close to the template contour where accidental matches typically occur. We combine these challenges in one discriminative approach.

The aim is to regularize directional chamfer matching by learning the characteristic co-occurrence of template pixels and the joint placement of background contours. As training

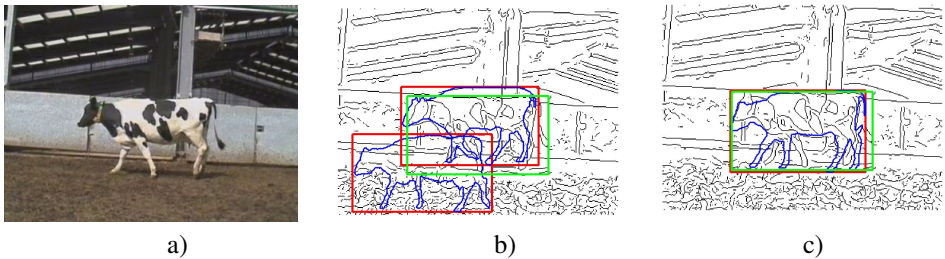


Figure 4: Learning discriminative weights for the co-occurrences of  $p_i^{(T,Q)}(\mathbf{x})$  improves the matching score of shape template as shown in the example here. The original image, the result obtained from directional chamfer matching and the result obtained from foreground reweighing are shown in panels a, b and c respectively. The groundtruth bounding box is shown in green and the top scoring object hypotheses are shown in red.

data this learning algorithm utilizes the object hypotheses obtained from running the directional chamfer matching code [14] on the training images. A hypothesis  $j$  with an overlap greater than 80% with the groundtruth is labeled as positive  $y_j = 1$ . This ensures that only good hypotheses which are matching to the actual object contours are selected as positive examples. For negative examples, we want to have the hypotheses where most of the object template matches in the background. Therefore, hypotheses with an overlap smaller than 40% with the groundtruth are labeled as negative  $y_j = -1$ . The learning algorithm is found to be robust to small variations in the cutoff values of 80% and 40% overlap criterion with the groundtruth. For each object hypothesis we build a feature vector  $f_j = [\bar{p}_1 \dots \bar{p}_L h_1 \dots h_G]$  consisting of the average pixel cost  $\bar{p}_i$  and the corresponding background histograms  $h_i$ , where  $L$  is the number of template edge pixels and  $G$  is the number of background contours.

Let  $\mathcal{K}(f_i, f_j)$  be a kernel that represents the similarity between feature vectors  $f_i, f_j$ . Subsequently, we use the radial basis kernel  $\mathcal{K}(f_i, f_j) = \exp(-\gamma \|f_i - f_j\|^2)$ . It is common practice in the field of kernel machines, to interpret the kernel  $\mathcal{K}(f_i, f_j)$  as a dot product of transformed features  $\psi(f_i), \psi(f_j)$ . Here  $\psi$  represents the mapping of the feature vector into a higher dimensional space. Due to the seminal ‘kernel trick’ [14] it is sufficient to define the kernel  $\mathcal{K}$  without explicitly representing the mapping  $\psi$ . We then seek weights  $w$  to be applied on  $\psi(f_i)$  so that the margin between positive and negative hypotheses in the transformed space is maximized. To model the joint co-occurrences of foreground and background contours we need to utilize a non-linear kernel that captures the relationship between foreground and background pairs, triples, quadruples and so on. From the polynomial kernel  $\mathcal{K}(f_i, f_j) = \langle f_i, f_j \rangle^2$  of degree 2 one can easily determine, that the mapping function  $\psi$  comprises all possible second order terms. It is straightforward, that a polynomial kernel of degree  $d$  comprises all possible combinations between feature dimensions up to degree  $d$ . Since the Taylor expansion of the RBF kernel is a infinite set of features corresponding to polynomial terms it comprises an infinite amount of feature combinations. We need to optimize the following max-margin classification problem to learn the weights  $w$ .

$$\min_{w, b, \xi} \frac{1}{2} \|w\|_2^2 + C \sum_{j=1}^N \xi_j \quad (5)$$

$$\text{subject to: } y_j (w^T \psi(f_j) + b) \geq 1 - \xi_j \quad \wedge \quad \xi_j \geq 0, \quad \forall j$$

where  $N$  is the number of training samples,  $b$  is the offset,  $C$  is the penalty and  $\xi_j$  are slack variables allowing for margin violations. Commonly (5) is converted into its dual form and

	Pedestrians	Cows	Giraffes	Mugs
DCM	3.0	88.1	27.0	10.1
Foreground Regularization	6.8	89.2	36.3	27.3
Regularized Chamfer Matching	11.2	91.9	43.0	27.3

Table 1: Comparison of **average precision** (in %) for three datasets namely, TUD Pedestrians, Cows and the ETHZ giraffes and mugs. We compare the basis of our approach (DCM) with the extension from Sec. 3.1 and our final learning of regularized chamfer matching.

solved for the dual SVM parameters, the support vectors  $S_i$ , their coefficients  $\alpha_i$  and the offset  $b$ . After training the combined model of foreground relevance and background accidentalness from (5) let us now utilize this model to improve upon the directional chamfer matching cost function (1). This improved, regularized chamfer distance  $d_{RDCM}^{(T,Q)}(\mathbf{x})$  again measures the distortion cost of object hypotheses  $f_j$ .  $f_j$  denotes the feature vector of  $j$ -th object hypothesis obtained by the placement of object template  $T$  at location  $\mathbf{x}$  in the query image  $Q$ . Since a non-linear radial basis kernel is employed, the regularized chamfer distance is obtained using the dual SVM parameters, obtained by solving the SVM optimization problem from (5) in its dual form,

$$d_{RDCM}^{(T,Q)}(\mathbf{x}) = 1 - \left( \sum_i \alpha_i \mathcal{K}(f_j, S_i) + b \right). \quad (6)$$

As in standard chamfer matching, candidate hypotheses are obtained by applying non-maximum suppression onto the regularized distances  $d_{RDCM}$ .

## 4 Experimental Evaluations

We now evaluate the discriminative chamfer regularization on several datasets which are commonly used for evaluation of chamfer matching. In particular, we compare with the directional chamfer matching (DCM) [14], which our model is built upon and with normalized oriented chamfer matching (NOCM) [15], which is a state-of-the-art extension to chamfer matching.

To obtain the edge maps used in the following we are utilizing the probabilistic boundary detector suggested in [16]. Furthermore we are using the support vector machine implementation of [8]. To perform directional chamfer matching, we are using the publicly available code of [14]. We use the same parameters from the downloaded version of the code for all the datasets. We used the same set of background shapes, as shown in Fig. 2, for all the datasets. The sizes of the background contours were adjusted relative to the size of foreground templates for each dataset. To measure the performance of our detection system we are using standard PASCAL overlap criterion.

In the first part of our experimental evaluation we are analyzing the individual contributions of the suggested foreground and background regularization and compare their performance to that of DCM on which we build our approach. In the second part we compare the performance of our combined object detector to state-of-the-art chamfer matching approaches NOCM and HDT.

### 4.1 Evaluating Foreground and Background Regularization

Subsequently, we evaluate the gain achieved by the proposed foreground and background regularization on context of category-level object detection in three standard datasets and we compare our results with the DCM baseline on which we build our approach.

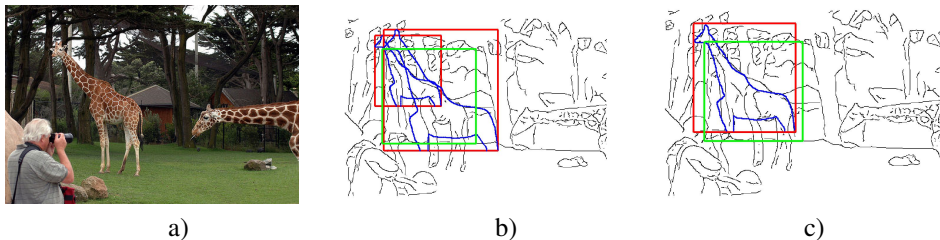


Figure 5: Learning co-occurrences of foreground shape template alone is not enough as shown in the example here. The original image, the result obtained from foreground reweighting and the result obtained from the combined foreground and background regularization are shown in panels a, b and c respectively. The spurious hypothesis resulting in panel b is suppressed by means of the combined regularization learned in (5)

The first dataset we are using is the TUD pedestrian dataset. As suggested in [15] we are using the larger training set, consisting of 400 side-view pedestrians, to build our detector and test our approach on the provided testset consisting of 250 test images. We use five masks of the training images as shape templates. The second dataset is the Cow dataset from the PASCAL Object Recognition Database Collection [16] which consists of 111 images in which cows appear with quite different articulation. We are following the protocol used in [15] to divide the dataset into training and testing sets. We use five masks of the training images as our shape templates. Finally, we evaluate on two challenging categories from the ETHZ shape dataset [9], giraffes and mugs. One hand-drawn template for each category is provided along with the dataset.

Our approach is efficient, as looking up the background contour costs from the integral image has negligible running time compared to computing the distance transformation of directional chamfer matching [14]. In Tab. 1 we are presenting our results for the DCM baseline, the performance of our foreground regularization method and of our combined detector. These experiments show that foreground regularization alone is already improving the average precision on all of these object categories. Additionally applying the background regularization is suppressing even more false positives in cluttered background.

For the TUD Pedestrian dataset the images in the testing set are given at a very high resolution which yields very low average precision for the directional chamfer matching which is around 3%. The low baseline can be attributed to the high resolution of the test images, since it is known that chamfer matching is sensitive to all the fine details in the edge map. Our suggested foreground regularization more than doubled the average precision to the baseline. Adding the background regularization brought a further gain of 4.5%.

For the Cow dataset directional chamfer matching yields very good performance around 88% average precision. Nevertheless, our combined detector could still improve performance about 4% by exploiting the advantages of foreground and background regularization. In Fig. 4 one can see how foreground reweighting is improving the alignment with the groundtruth and that it also suppresses false positives.

The background normalization becomes particularly useful in cases of challenging objects appearing in images with a lot of clutter like the ETHZ giraffes. Performance improves by 16% in terms of average precision using our combined detector. 7% out of this gain could be attributed to background regularization. The example in Fig. 5 shows that foreground regularization is not always able to suppress false positives in cluttered background and how background regularization can handle such cases.

For rather simple objects like ETHZ mugs we observed that explaining the foreground more



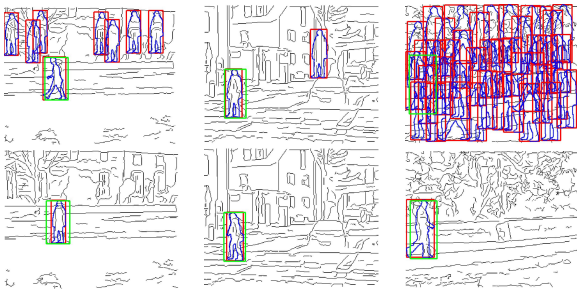


Figure 6: The top row shows detection results using the directional chamfer matching method. Bottom row shows the improved detections applying our regularized chamfer matching. The groundtruth bounding box is shown in green and the top scoring object hypotheses are shown in red.

	Cows	Peds
Chamfer Matching	73.9	4.4
NOCM [15]	91.0	70.0
HDT [22]	88.2	-
Regularized Chamfer Matching	98.3	80.0

Table 2: Comparison in terms of **detection rate** (in %) at 10% precision on the Cow dataset and the TUD Pedestrian dataset with standard chamfer matching, NOCM and HDT.

accurately is more important than suppressing false detections in cluttered background. We observed 17.3% improvement in average precision by learning the co-occurrence of template pixels while our combined detector is giving results in the same range.

All in all our combined detector using foreground and background regularization is achieving significant gain on all of the four categories compared to directional chamfer matching. Additional detection results comparing the regularized chamfer matching to directional chamfer matching are provided in Fig. 6.

## 4.2 Comparison with Chamfer Matching Methods

Furthermore we are comparing our method with two other state-of-the-art approaches on three datasets. The first method we are comparing our approach to is the normalized oriented chamfer matching by Ma et al. [15] (NOCD) since they also incorporate background into chamfer matching. We also compare our approach with the work of Zhu et al. [22] who utilize a novel probabilistic model called hierarchical deformable template model (HDT). [22] use one example learning in their evaluation whereas we utilize 5 templates for the TUD Cows and TUD Pedestrians.

[15] have reported results on two datasets: the TUD Pedestrian dataset [11] and the Cow dataset [10]. [22] have evaluated their method on the Cow dataset. Both approaches are reporting their results in terms of detection rate at 10% precision. In the previous section we are reporting in terms of average precision, since it is taking into account the area under the precision recall curve instead of just one point and therefore is a much more robust measure. However, to compare ourselves to [15, 22], we are reporting results in terms of detection rate at 10% precision.

Tab. 2 shows the results for the Cow dataset and the TUD Pedestrian dataset. We observed that to make the DCM baseline comparable to the OCM baseline the edge maps in the test images need to be downscaled. Hence, we report our final detection performance on the downscaled version of the test images. The results indicate that chamfer regularization is significantly improving performance on the Cow dataset compared to HDT and NOCM. For TUD Pedestrians we gain 10% in detection rate compared to NOCM, when running the directional chamfer matching on downscaled test images. All in all our results confirm that the regularized chamfer matching method is significantly improving over state-of-the-art chamfer matching techniques.

## 5 Conclusion

In this paper we have addressed two issues that limit the performance of the established and widely used chamfer matching technique, its susceptibility to clutter due to accidental matches and the fact that all model points are treated as being independent and equally relevant. By learning the co-occurrence of model points we have modeled the varying relevance of different foreground pixels and increased the specificity of the model. By allowing a codebook of simple, generic contours to be flexibly placed along the model contour where spurious matches are most likely, accidental matches can be discovered. Learning the joint placement of all of these generic background contours does then suppress accidental matches to clutter. Both extensions are integrated in a single discriminative learning approach and the method is based upon a publicly available, state-of-the-art chamfer method thus demonstrating its simple and wide applicability. The approach has been shown to successfully improve current chamfer matching approaches on standard datasets. <sup>1</sup>

## References

- [1] M. Andriluka, S. Roth, and B. Schiele. People-tracking-by-detection and people-detection-by-tracking. *CVPR*, 2008.
- [2] F. Attneave. Some informational aspects of visual perception. *Psychological review*, 61(3):183–193, 1954.
- [3] H. G. Barrow, J. M. Tenenbaum, R. C. Bolles, and H. C. Wolf. Parametric correspondence and chamfer matching: Two new techniques for image matching. *Int. Joint Conf. Artificial Intelligence*, pages 659–663, 1977.
- [4] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *PAMI*, 24(4):509–522, 2002.
- [5] I. Biederman. Recognition-by-components: A theory of human image understanding. *Psychological review*, 4(2):115–147, 1987.
- [6] G. Borgefors. Hierarchical chamfer matching: A parametric edge matching algorithm. *PAMI*, 10(6):849–865, 1988.
- [7] B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. *5th Annual ACM Workshop on COLT*, pages 144–152, 1992.
- [8] C. Chang and C. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3):27:1–27:27, 2011. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- [9] V. Ferarri, T. Tuytelaars, and L. V. Gool. Object detection by contour segment networks. *ECCV*, 2006.
- [10] D. M. Gavrila and S. Munder. Multi-cue pedestrian detection and tracking from a moving vehicle. *International Journal of Computer Vision*, 73(1):41–49, 2007.

---

<sup>1</sup>This work was supported by the Excellence Initiative of the German Federal Government and the Frontier fund, DFG project number ZUK 49/1.

- [11] B. Leibe, A. Leonardis, and B. Schiele. Combined object categorization and segmentation with an implicit shape model. *ECCV'04 Workshop on Statistical Learning in Computer Vision.*, 2004.
- [12] B. Leibe, E. Seemann, and B. Schiele. Pedestrian detection in crowded scenes. *CVPR*, 2005.
- [13] Z. Lin, L. S. Davis, D. Doermann, and D. DeMenthon. Hierarchical part template matching for human detection and segmentation. *ICCV*, 2007.
- [14] M. Liu, O. Tuzel, A. Veeraraghavan, and R. Chellappa. Fast directional chamfer matching. *CVPR*, 2010.
- [15] T. Ma, X. Yang, and L. Latecki. Boosting chamfer matching by learning chamfer distance normalization. *ECCV*, 2010.
- [16] D. Martin, C. Fowlkes, and C. Malik. Learning to detect natural image boundaries using local brightness, color and texture cues. *PAMI*, 26(5):530–549, 2004.
- [17] B. Ommer and J. M. Buhmann. Learning the compositional nature of visual object categories for recognition. *PAMI*, 32(3):501–516, 2010.
- [18] B. Ommer and J. Malik. Multi-scale object detection by clustering lines. *ICCV*, 2009.
- [19] J. Shotton, A. Blake, and R. Cipolla. Multi-scale categorical object recognition using contour fragments. *PAMI*, 30(7):1270–1281, 2008.
- [20] A. Thayananthan, B. Stenger, P. Torr, and R. Cipolla. Shape context and chamfer matching in cluttered scenes. *CVPR*, 2003.
- [21] P. Yarlagadda, A. Monroy, and B. Ommer. Voting by grouping dependent parts. *ECCV*, 2010.
- [22] L. Zhu, Y. Chen, and A. Yuille. Learning a hierarchical deformable template for rapid deformable object parsing. *PAMI*, 32(6):1029–1043, 2010.