

# Learning Feature Distance Measures for Image Correspondences

Xi Chen<sup>1</sup>      Tat-Jen Cham<sup>1,2</sup>

<sup>1</sup>School of Computer Engineering; Nanyang Technological University; Singapore 639798

<sup>2</sup>Singapore-MIT Alliance

chenxi@pmail.ntu.edu.sg

astjcham@ntu.edu.sg

## Abstract

*Standard but ad hoc measures such as sum-of-squared pixel differences (SSD) are often used when comparing and registering two images that have not been previously observed before. In this paper, we propose a framework to address the problem of learning a parametric feature distance measure to measure the dissimilarity between pairs of images. The method is based on optimizing the parameters of the distance measure in order to minimize correspondence classification errors on training data. Because the learning process involves relative (rather than absolute) visual content between image pairs, the learned distance measure may also be applied to other images with very different visual content. Results on matching classification with a wide variety of image content show that the learned feature distance measure clearly outperforms the standard measures of SSD, chamfer and Bhattacharyya histogram distances.*

## 1. Introduction

Appearance-based object detection methods [16, 8] are increasingly popular, in which classifiers are trained to recognize specific categories of objects. The training data typically comprise of a positive class of images that represent the appearance of objects in the object category, and a negative class of images that represent the appearance of objects that do not belong to the object category.

Such appearance-based classifiers may be learned for image matching or registration problems only if there are apriori known reference exemplars of the images or image features, such that they may be included in a training set. Generally once training is completed, the classifiers may only be used to retrieve images with similar visual content to the training images.

However there is a substantially large class of image matching or registration problems and scenarios that in-

volve the comparison (or deriving the correspondence) of two test images having non-specific visual content that cannot be represented by training data. One example of such a problem is image mosaicing, where the visual content may not only be unknown apriori but may also be wildly varying from one instance to the next; thus traditional appearance-based classifiers cannot be applied. Another related scenario is in the online tracking of objects for which there may not be sufficient (or even any) prior exemplars for the use of appearance-based classifiers. For example, this may occur if a visual tracking system needs to be operated quickly after hardware set up, and for which the camera angles are not significantly constrained. Nevertheless, there will likely be a strong appearance similarity between consecutive video frames that should lend itself to be exploited.

### 1.1. Learning Image Correspondences

In the past, standard measures for evaluating correspondences are the sum-of-squared pixel differences (SSD), standard metrics in feature space (e.g. Euclidean is used in the SIFT framework [11]), the chamfer distance [3], the shuffle distance [15] and histogram measures such as  $\chi^2$  distance, Bhattacharyya distance (as used in the mean-shift framework [6]), Kullback-Leibler divergence and Earth Mover's Distance [12]. These measures, while intuitive, are nevertheless ad hoc and may be sub-optimal in terms of accuracy. The only related work that has non-trivial overlap with this paper is our earlier work [4], where an optimal feature distance was learned from training data. However, this earlier method was based on a framework of a single image exemplar representing a class of images, followed by optimizing the distance measure based on the Fisher discriminant. Our new framework is based on pairs of images, and employs different features and measures that makes it much superior and relevant.

In this paper, we will address the problem of *learning and classifying correspondences* rather than learning appearance classifiers. Our goal is to *learn a feature distance measure* from labeled training data that is able to be ap-

plied to other images containing substantial *visual bias* from those present in the training data (see figure 1). The training data consists of pairs of (spatially transformed) images that are either labeled as corresponding or non-corresponding. The matching classifier takes the form of a novel learned distance measure and threshold. Additionally, this parametric family of distance measures subsumes SSD, chamfer distance and shuffle distance.

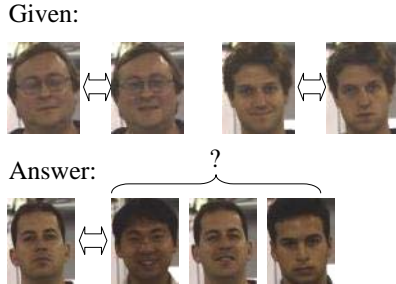


Figure 1. Classifying correspondences with different visual content. From a training set of paired images, the goal is to learn a match classifier that works on images with different visual content.

This paper makes the following contributions:

- A novel framework is described for learning image correspondences, rather than learning appearance classes directly;
- A parametric family of measures, known as the *cross-affinity distance measures*, is proposed that subsume commonly-used measures such as SSD, chamfer distance and shuffle distance as degenerate instances – hence if one of these measures are truly optimal it will be selected in the learning process; and
- the paper reports experimental results that clearly indicate how the optimally learned distance measure outperforms existing measures in test images containing substantial bias in visual content.

## 2. Basic Theoretical Concepts

Suppose we have an image  $A$ , which is to be compared to a second image  $B$ , subject to a spatial transformation  $S_{\theta}(\cdot)$  with parameters  $\theta$ . The exact representation for  $A$  and  $B$  will be described in section 4.1. Let the transformed version of  $B$  be expressed as  $B'$  given by

$$B' = S_{\theta}(B)$$

The *match decision* problem of determining if  $A$  matches  $B'$  may then be expressed as a binary classifier function

$T_{\phi,t}(\cdot)$ :

$$\begin{aligned} T_{\phi,t}(A, B, \theta) &= T'_t(D_{\phi}(A, S_{\theta}(B))) \\ &= \begin{cases} 1, & \text{if } D_{\phi}(A, B') \leq t \\ -1, & \text{otherwise} \end{cases} \quad (1) \end{aligned}$$

where  $D_{\phi}(A, B')$  is a positive real-valued distance function that increases with a conceptual dissimilarity between images  $A$  and  $B'$ , and is parameterized by  $\phi$ ; while  $T'_t(\cdot)$  is the threshold function on the distance measure, with threshold  $t$ . The exact form of  $D_{\phi}(\cdot)$  will be described in section 4.2.

An important point here is that the term “match” does *not* mean that  $A$  and  $B'$  are identical. Matching here indicates that  $A$  and  $B$  are related with sufficient accuracy in the relevant transformation parameters  $\theta$  that the user is interested in.  $A$  and  $B'$  may be considered to be “matched” despite large differences in transformation parameters that the user wants to ignore or be *invariant* to. An example is that an image patch may be considered to be matched with a 90° rotated version if the user wants rotation invariance.

## 3. The Notion of Similarity

The notion of similarity between two images or features is highly subject to *human* opinion, depending on the perception of individuals or the requirements of the domain. For example, in various problems that involve tracking an object, the notion of “similar” can range from similarity in the color histograms of the tracked patches, to similarity of gray levels between exact corresponding pairs of pixels. There are no universally true quantitative distance measures of dissimilarity, except that all such distance measures should be zero when the images or features are absolutely identical.

Any proposed distance measures must therefore be context-specific and dependent on the spatial transformation model complexity. An ideal distance measure should be sensitive to the transformation parameters of interest and to the error tolerance on these estimates, but be fully invariant to those that the user wants to ignore. However, most vision algorithms in matching converge on standard measures such as the sum-of-squared pixel differences (SSD), chamfer distance, shuffle distance and histogram distances, which may not meet the proper contextual requirements for a suitable measure.

On the other hand, it may be argued that having the binary match classifier function in (1) is sufficient for the purpose of matching. However, a real-valued distance measure is still very useful in many circumstances. A distance measure allows the ranking of different match hypotheses. Additionally, such a measure will also enable directed (e.g.

gradient-based) searches that are based on minimizing this distance measure.

### 3.1. Learning from Labeled Correspondences

As stated previously, the goal in this paper is to learn a suitable feature distance measure from labeled training data that capture the contextual requirements for matching. In our context, this means learning the function  $D_\phi(\cdot)$  by discovering the optimal parameters  $\hat{\phi}$ .

Suppose that training data of correspondences is available comprising instances that are tuples  $(A, B, \theta, l)$  where  $l$  is a label. There are two approaches to learning a distance measure from labeled correspondences:

- *Direct approach.* Here we can attempt to directly learn the distance measure  $D_\phi(A, B')$  if the labels  $l$  are *real positive values* denoting the distance between  $A$  and  $B'$ . In this approach,  $\phi$  may be optimized such that

$$\hat{\phi} = \arg \min_{\phi} \left\{ \sum_{\text{training set}} (D_\phi(A, B') - l)^2 \right\} \quad (2)$$

for all instances. However, this approach requires either that ground truth transformation parameters are available per training image to generate the distance labels (not always readily available), or that users manually specify a subjective distance label (noisy and error-prone).

- *Indirect Approach.* This approach involves indirectly learning a suitable distance measure through learning the match classifier function. This approach only requires that the labels  $l$  are binary class labels:

$$l_{A,B'} = \begin{cases} 1 & A \text{ and } B' \text{ match with sufficient accuracy} \\ -1 & A \text{ and } B' \text{ do not match} \end{cases} \quad (3)$$

Such labels are much easier to specify, particularly if they have to be labeled manually. Treating  $T_{\phi,t}(\cdot)$  in the classification framework, we optimize  $\phi$  and threshold  $t$  in order to *minimize the probability of classification error*:

$$(\hat{\phi}, \hat{t}) = \arg \min_{\phi, t} \{ p(T_{\phi,t} = 1 | l = -1) + p(T_{\phi,t} = -1 | l = 1) \} \quad (4)$$

This is the approach that is taken in this paper.

By providing a substantially comprehensive training set of example pairs of matched images and mismatched images, the framework will learn the match classifier function

$T_{\phi,t}(A, B, \theta)$ , and thus the correct feature distance measure  $D_\phi(A, B')$  and threshold  $\hat{t}$  to use for matching. This distance measure will not be dependent on the *absolute* visual content of the images, but instead depend only on the *relative* visual content of pairs of images.

## 4. Parametric Family of Distance Measures for Feature Comparison

The form of distance measures is crucial in achieving a good result in the classification framework, but should also have smooth properties that make the distance measure usable in more general situations beyond matching. The form that is selected in this paper has a further advantage that it subsumes other commonly used distance measures as degenerate special cases, as discussed later in section 4.3.

The form of the distance measure is however tied to the image representation used, and since we are using a non-traditional representation, this is described below.

### 4.1. Image Representation

Most vision algorithms treat images as a vector of pixels. In contrast, we use an image representation that is a set of pixel (or feature) vectors. This representation is similar to those used in [4, 17, 10, 9].

Consider an image in the continuous domain defined by  $F(x, y)$ . Unlike most existing work that consider  $F(x, y)$  to be sampled with the same sampling function such that the resulting images may be treated as vectors in the same observation space, we instead propose a framework whereby the sampling functions are different. This is useful in a number of different situations, e.g.

- when images are at different resolutions, or
- if the images are preprocessed by a segmentation algorithm giving rise to regions with different number of pixels, or
- if geometric features are used such as corners and edges (as we do later), or
- in a real-time algorithm where not all pixels can be processed and need to be randomly sampled.

Suppose that  $K$  samples are obtained from  $F(x, y)$ . The image is then expressed in a discrete form as a set of samples given by:

$$\mathcal{F} = \{ \mathbf{f}_k \mid k = 1, \dots, K \} \quad (5)$$

where

$$\mathbf{f}_k = [ x_{fk} \quad y_{fk} \quad F(x_{fk}, y_{fk}) ]^T \quad (6)$$

Here  $(x_{fk}, y_{fk})$  are sampled locations in  $F$  according to the spatial sampling function. As hinted above,  $F(x_{fk}, y_{fk})$  may be replaced by feature values obtained through the use of feature detectors. See figure 2 for an illustration.

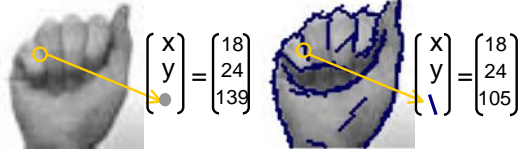


Figure 2. Different image representations.

While this image representation is similar to those used in previous work cited above, the application of this representation is vastly different (except in our previous work [4]).

## 4.2. Distance Measure based on Feature Cross-Affinity

Given the image  $\mathcal{F}$  and another image  $\mathcal{G} = \{g_j \mid j = 1, \dots, J\}$  expressed in the form of sets of feature vectors, we now define a parametric distance measure to evaluate the similarity between them. The proposed distance measure is a generalization of the *Average Hausdorff Distance* [7, 2]:

$$D_\phi(\mathcal{F}, \mathcal{G}) = \frac{1}{|\mathcal{F}|} \sum_{\mathbf{f} \in \mathcal{F}} \min_{\mathbf{g} \in \mathcal{G}} h(\mathbf{f}, \mathbf{g}) + \frac{1}{|\mathcal{G}|} \sum_{\mathbf{g} \in \mathcal{G}} \min_{\mathbf{f} \in \mathcal{F}} h(\mathbf{g}, \mathbf{f}) \quad (7)$$

where  $h(\mathbf{f}, \mathbf{g})$  is a Mahalanobis distance function comparing  $\mathbf{f}$  to  $\mathbf{g}$  given by

$$h(\mathbf{f}, \mathbf{g}) = \sqrt{(\mathbf{f} - \mathbf{g})^T \mathbf{C}^{-1} (\mathbf{f} - \mathbf{g})} \quad (8)$$

where  $\mathbf{C}$  is a symmetric positive definite matrix (equivalent to the covariance matrix for a Mahalanobis distance). The matrix  $\mathbf{C}$  is called the cross-affinity matrix and encapsulates the six degrees of freedom (i.e.  $\phi$ ) available in optimizing the distance measure  $D$  for image matching.

Intuitively, this family of distance measures computes a ‘best’ correspondence between two pixels in  $\mathcal{F}$  and  $\mathcal{G}$ , where the measured feature differences are not limited to intensity differences of pixels with the same relative spatial position as in the case of straightforward SSD. Instead, some trade-off is allowed in pixel intensity differences and pixel position perturbation, when measuring the feature differences. The amounts of difference and trade-off are controlled by  $\mathbf{C}$ , and learned from data.

## 4.3. Comparison to Standard Distance Measures

The cross-affinity family of distance measures subsume commonly-used metrics as degenerate instances of  $\mathbf{C}$  as

shown below:

$$\mathbf{C}_{\text{SSD}} = \begin{bmatrix} 0^+ & 0 & 0 \\ 0 & 0^+ & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{C}_{\text{Chamfer}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0^+ \end{bmatrix}$$

$$\mathbf{C}_{\text{Shuffle}} = \begin{bmatrix} \infty & 0 & 0 \\ 0 & \infty & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (9)$$

This may be considered intuitively. SSD enforces a strict correspondence and only pixels with the same spatial coordinates to be compared. Since  $\mathbf{C}$  is the covariance matrix under a Mahalanobis interpretation, the positive zeros in the first two diagonal entries indicate that spatial coordinates are not allowed to vary (zero variance) when scouring for the best pixel ‘correspondences’ in computing the distance. Similarly, chamfer matching requires that edge pixels match to only edge pixels, hence the diagonal entry corresponding to feature value variance is zero; however it allows for spatial variation and thus the non-zero variances for spatial coordinates. The same analysis may be applied to the shuffle distance.

Having the family of cross-affinity distance measures subsume these commonly-used metrics is a strong benefit. If one of these standard measures is truly the optimal measure, learning the best cross-affinity distance measure will result in learning this same optimal standard measure. Hence the cross-affinity distance measure is instantly guaranteed to outperform these standard measures when applied to the training set. This is also shown to be empirically true for test sets in section 6.

## 5. Discriminative Analysis by Histogram Intersection

The misclassification rate here is the sum of probabilities of false positives and false negatives. Given a fixed  $\phi$  and  $t$ , this probability is

$$p(\text{error}) = p(T_{\phi, t} = 1 \mid l = -1) + p(T_{\phi, t} = -1 \mid l = 1) \quad (10)$$

Assuming that an optimal threshold  $t$  is used, this error may also be expressed as the *intersection* of the distributions of matches and mismatches in the one-dimensional cross-affinity distance space (figure 3):

$$p(\text{error}) = \int_{-\infty}^{+\infty} \min \{p(l = 1, D_\phi), p(l = -1, D_\phi)\} dD_\phi \quad (11)$$

The distributions of matches and mismatches in feature distance space can be approximated through the use of smoothed histograms. Given the set of matched training

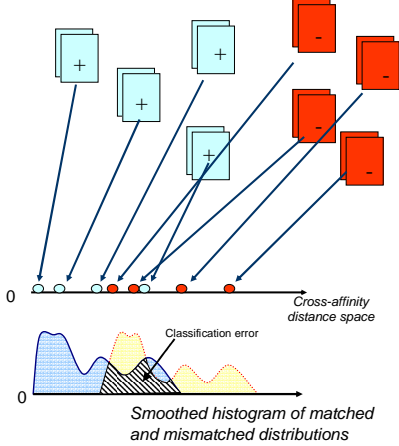


Figure 3. Smoothed histogram computation for matched and mismatched classes, and classification error in the form of histogram intersection.

instances  $\{(A, B, \theta, l) \mid l = 1\}$ , we can compute a set of positive class distances  $\{d_{P_i} = D_{\phi}(A_i, B'_i) \mid i = 1, \dots, n\}$ . The smoothed histogram is defined as

$$\hat{f}_P(x) = \frac{1}{nh} \sum_{i=1}^n K\left(\frac{x - d_{P_i}}{h}\right) \quad (12)$$

where  $x$  is the cross-affinity distance space ordinate,  $K(\cdot)$  is the Gaussian kernel and  $h$  is a smoothing parameter. In this paper, an optimal  $h$  based on [13] is used.

Similarly, for the negative distances  $\{d_{N_i}, i = 1, \dots, m\}$ , a smoothed histogram may also be obtained:

$$\hat{f}_N(x) = \frac{1}{mh} \sum_{i=1}^m K\left(\frac{x - d_{N_i}}{h}\right) \quad (13)$$

The smoothed histogram intersection distance is defined by

$$d_{\cap}(\hat{f}_P, \hat{f}_N) = 1 - \int_{-\infty}^{+\infty} \min(\hat{f}_P(x), \hat{f}_N(x)) dx \quad (14)$$

The optimal matrix  $\mathbf{C}$  corresponds to the largest  $d_{\cap}$  by this definition.

### 5.1. Minimization of the Histogram Intersection

The minimization of the histogram intersection distance is carried by manipulating the cross-affinity matrix  $\mathbf{C}$  via the BFGS Quasi-Newton method [5]. Since this is a local gradient-based search, a number of different random initialization points were used in order to better find the global minimum.

There is a minor complication to updating the  $\mathbf{C}$  matrix since the elements are not independent. In order to preserve

the symmetric positive definiteness of  $\mathbf{C}$ , the Cholesky decomposition form is used instead

$$\mathbf{C} = \mathbf{U}^T \mathbf{U} \quad (15)$$

where  $\mathbf{U}$  is an upper triangular matrix. Optimization is carried out with respect to the six free parameters in the  $\mathbf{U}$  matrix.

## 6. Experiments

We conducted classification and tracking experiments on three sets of data: the first is the Triesch hand posture database, which is used as the PETS2002 [1] posture data set. The second is the CMU pose, illumination and expression (PIE) database of human faces [14]. The third is a car sequence, which contains 50 frames of size  $320 \times 240$ , showing images from a camera in a trailing car, observing the car in front negotiating a slight bend.

### 6.1. Experiments on the Triesch Database

An edge-based classification task was performed on the Triesch database, with the intention of testing the ability of our method to distinguish between different postures performed by random people. The Triesch hand posture database consists of 10 hand signs performed by 24 persons. The images are recorded in 8-bit grayscale and are  $128 \times 128$  pixels in size. We selected pairs of images from the Triesch database to form a training set and a testing set. The training set includes images of the first 5 hand postures performed by all the 24 persons in front of uniform light background, and the test set includes images of the last 5 postures performed by all the 24 persons in front of uniform light background. All images in the training and testing sets are cropped hand images, so they are of different sizes, ranges from  $45 \times 47$  to  $95 \times 117$ .

For either of the training and testing sets, the images are organized into 50 positive pairs and 100 negative pairs. Each positive pair contains two images of the same hand posture but from different persons, and each negative pair contains two images of different hand postures from random persons. Our aim is to find the optimal Cross-Affinity Distance to best distinguish between the positive and the negative pairs. Some sample image pairs from the training set are shown in figure 4. The 3 pairs in the top row are positive pairs, and the 3 pairs in the bottom are negative pairs. All the 5 hand postures in the training set can be found in figure 4.

Oriented edge features are used in this experiment. These edge features are extracted by a set of 12 templates covering the directions from  $0^\circ$  to  $180^\circ$  with  $15^\circ$  step. The size of the templates is  $7 \times 7$ . The first seven templates are

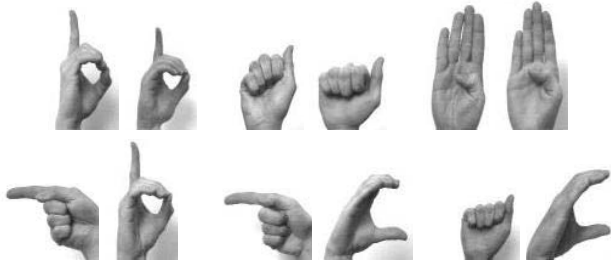


Figure 4. Positive and negative image pairs from cropped Triesch training set.

show in the top row of figure 5. The edge extraction algorithm is simple normalized correlation with thresholding, and non-maximum suppression is used to reduce the number of extracted edges. Five extraction results are show in the bottom row of figure 5, short lines representing extracted oriented edges are superposed on the original images. These five images also show the 5 hand postures in the testing set. Then in the following experiment, each hand posture is represented by a set of oriented edge features with 3 properties: the  $x$ ,  $y$  positions and the edge direction.



Figure 5. Edge templates and extracted edge features.

We also ran experiments on traditional Chamfer matching method to compare the results with our method. The resulting ROC curves for our method and Chamfer matching is show in figure 6.

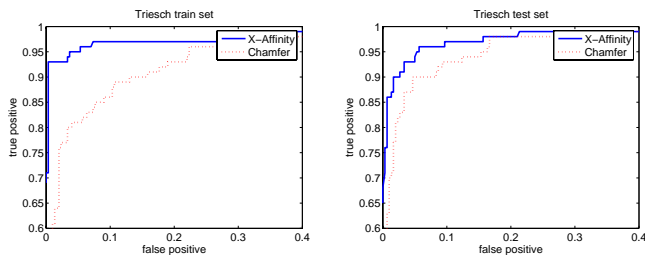


Figure 6. ROC curves for training(left)and test(right) sets of Triesch hand dataset.

Some classification results of the test set are shown in figure 7. All the image pairs are misclassified by Chamfer method, but only the two indicated by “ $\times$ ” are misclassified by Cross-Affinity. The classification threshold is based on an EER (equal error rate) threshold.

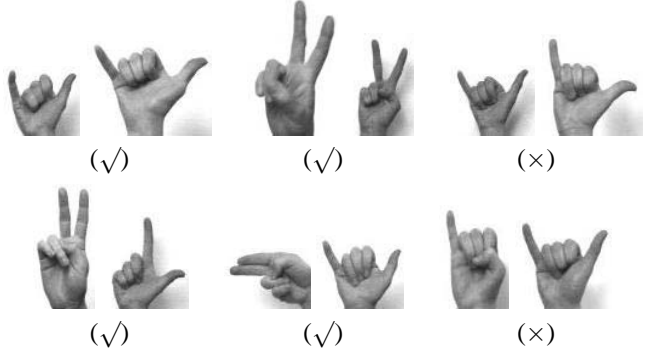


Figure 7. Positive and negative image pairs from cropped Triesch test set. The top row are positive pairs, and the bottom row are negative pairs. All these image pairs are misclassified by Chamfer distance, while only two are misclassified by Cross-Affinity distance.

## 6.2. Experiments on the PIE Face Database

An intensity-based classification task was performed on the PIE database, in order to test the ability of our method to distinguish between different human faces in random expressions.

The CMU Pose, Illumination and Expression (PIE) database [14] contains 41368 face images of 68 people. The images are of different poses, illuminations and expressions. We chose images from the first 20 people to form the training set, and images from the next 20 people to form the test set. All the selected images are in the front view with the same illumination, but contain three different expressions – neutral, smile and blink. The images are cropped to just cover the face area, of size  $53 \times 67$ .

For both training and test sets, we constructed 50 positive image pairs as the positive set and 100 negative images pairs as the negative set. Each positive pair contains 2 images from the same person but of different expressions, each negative pair contains 2 images from different persons of random expressions. Some sample image pairs for the training set is shown in figure 8; the three pairs in the top are positive pairs, and the three pairs in the bottom are negative pairs.



Figure 8. Positive and negative image pairs from cropped PIE database.

We also ran experiments on SSD and the histogram-based Bhattacharyya measure for comparison. The resulting ROC curves for our method, SSD and histogram measure are shown in figure 9. The left graph is for the training set, and the right graph is for the test set. The curve for X-Affinity does not appear in the left graph because it is superposed by the axes.

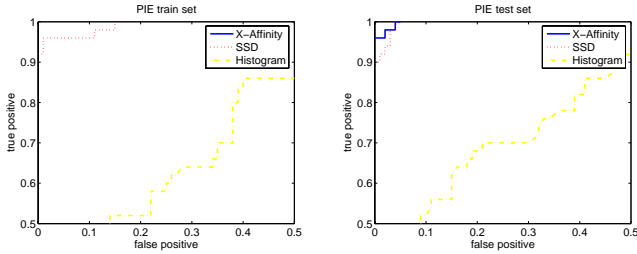


Figure 9. ROC curves for the face classification training (left) and test (right) sets using different measures.

### 6.3. Experiments on a Car Sequence

The car sequence contains 50 frames of size  $320 \times 240$ , showing images from a camera in a trailing car, observing the car in front negotiating a slight bend. In the experiment, we investigated the classification performance of different metrics when applied to a dataset of smaller images that were manually cropped from the sequences mentioned above. The cropped images are of size  $25 \times 21$ .

The cropped dataset was divided into a training set and a test set. The training set contains the first 25 frames and the test set contains the last 25 frames. Either set has 25 images of the car, i.e. the target regions, in the positive class, and 50 images in the negative class which are randomly sampled from the original frames away from the target regions. Example images (in increasing frame order) from the positive class of the car dataset are shown in the top row of figure 10, while background images are shown in the second row. Notice that in the dataset, the target appearance changes systematically across the sequences, e.g. the car in the sequence becomes increasingly smaller.

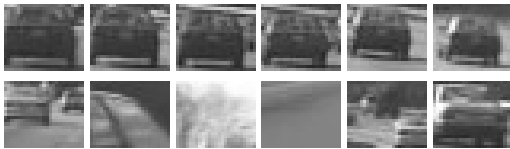


Figure 10. Positive and negative cropped samples from the car dataset.

Then in both cases for the training and test sets, there is a set of 625 ( $25 \times 25$ ) positive image pairs, which contains all the possible pairs between two positive images; and there is

a negative image pairs set of size 1250 ( $25 \times 50$ ), containing all the possible pairs between a positive image and a negative image. We did both pixel-based and edge-based experiments for this data set, the settings are the same with the previous two experiments respectively. For pixel-based experiments, the resulting ROC curves is shown in figure 11. For edge-based experiments, the resulting ROC curves is shown in figure 12.

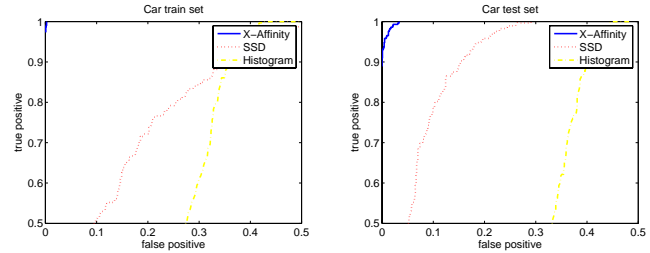


Figure 11. ROC curves for the car training (left) and test (right) sets using pixel intensities.

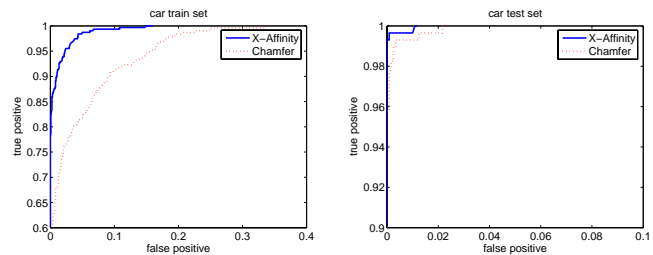


Figure 12. ROC curves for the car training (left) and test (right) sets using edge features.

We did further experiments on a tracking task. In this experiment, we investigated how well the optimized cross-affinity distance measure performs in a tracking task. The tracking procedure is as follows: the reference image is a small image patch located on the target in the first frame of the sequence. For each successive frame, the search window for the new target position is centered on the target position in the previous frame, using a window 2.5 times the size of the reference image. The search window is scanned, and the position with the smallest matching distance to the reference image is selected as the new target position. The frames in which tracking fails catastrophically (as opposed to minor registration errors) are noted. The reference image does not change throughout the tracking task. The optimal measure computed from the previous section is used in the tracking task. No threshold is required as the position with the minimum distance is always selected as the target position. Additionally, no background modeling is done to aid the tracking. Some sample tracked images are shown in figure 13. The dark rectangles represent the target positions, while the light rectangles represent the search windows.



Figure 13. Sample tracked images.

As before, the tracking task was performed using pixel-based SSD and histogram-based Bhattacharyya distance, and edge based Chamfer distance for comparison. Table 1 shows the number of successive frames tracked successfully using the different distance measures. It is clear that the optimal cross-affinity measure leads to more robust tracking.

| SSD | Histogram | X-A | X-A(edge) | Chamfer(edge) |
|-----|-----------|-----|-----------|---------------|
| 13  | 13        | 40  | 20        | 16            |

Table 1. Number of consecutive frames tracked.

## 7. Summary and Conclusions

In this paper, we proposed a framework for learning the optimal feature distance measure to match pairs of images. It involves a parametric family of distance measures, called the cross-affinity distance measure, which is optimized to minimize image matching classification errors. Results on matching classification with a wide variety of image content show that the learned feature distance measure clearly outperforms the standard measures of SSD, chamfer and Bhattacharyya histogram distances.

## Acknowledgements

This work is carried out in the Centre for Multimedia and Network Technology (CeMNet). The authors would also like to thank Chris Dance for valuable comments.

## References

- [1] *Third IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS)*, Copenhagen, Denmark, 2002.
- [2] A. Barla, F. Odone, and A. Verri. Hausdorff kernel for 3D object acquisition and detection. In *Proc. Euro. Conf. on Computer Vision*, pages 20–33, Copenhagen, Denmark, 2002.
- [3] H. Barrow, J. Tenenbaum, R. Bolles, and H. Wolf. Parametric correspondence and chamfer matching: Two new techniques for image matching. In *Proc. Int. Joint Conf. on Artificial Intelligence*, pages 659–663, Cambridge, MA, 1977.
- [4] X. Chen and T.J. Cham. Discriminative distance measures for image matching. In *Proc. Int. Conf. on Pattern Recognition*, pages 691–695, Cambridge, England, 2004.
- [5] E.K.P. Chong and S.H. Zak. *An Introduction to Optimization*. John Wiley, 2nd edition, 2001.
- [6] D. Comaniciu, V. Ramesh, and P. Meer. Real-time tracking of non-rigid objects using mean shift. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 142–149, Hilton Head, SC, 2000.
- [7] M.P. Dubuisson and A.K. Jain. A modified hausdorff distance for object matching. In *Proc. Int. Conf. on Pattern Recognition*, pages 566–568, Jerusalem, Israel, 1994.
- [8] B. Heisele, P. Ho, and T. Poggio. Face recognition with support vector machines: Global versus component-based approach. In *Proc. Int. Conf. on Computer Vision*, pages 688–694, Vancouver, Canada, 2001.
- [9] T. Jebara. Images as bags of pixels. In *Proc. Int. Conf. on Computer Vision*, pages 265–272, Nice, France, 2003.
- [10] R. Kondor and T. Jebara. A kernel between sets of vectors. In *Proc. Int. Conf. on Machine Learning*, pages 361–368, Washington, DC, 2003.
- [11] D.G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. Journal of Computer Vision*, 60(2):91–110, 2004.
- [12] Y. Rubner, C. Tomasi, and L.J. Guibas. The earth mover’s distance as a metric for image retrieval. *Int. Journal of Computer Vision*, 40(2):99–121, 2000.
- [13] B.W. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman and Hall, 1986.
- [14] T. Sim, S. Baker, and M. Bsat. The CMU Pose, Illumination, and Expression database. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 25(12):1615–1618, 2003.
- [15] K. Toyama and A. Blake. Probabilistic tracking in a metric space. In *Proc. Int. Conf. on Computer Vision*, pages 50–57, Vancouver, Canada, 2001.
- [16] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 511–518, Kauai, HI, 2001.
- [17] L. Wolf and A. Shashua. Kernel principal angles for classification machines with applications to image sequence interpretation. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 635–640, Madison, WI, 2003.