Viewpoint Invariant Pedestrian Recognition with an Ensemble of Localized Features

Douglas Gray and Hai Tao

University of California, Santa Cruz
{dgray,tao}@soe.ucsc.edu
http://vision.soe.ucsc.edu/

Abstract. Viewpoint invariant pedestrian recognition is an important yet under-addressed problem in computer vision. This is likely due to the difficulty in matching two objects with unknown viewpoint and pose. This paper presents a method of performing viewpoint invariant pedestrian recognition using an efficiently and intelligently designed object representation, the ensemble of localized features (ELF). Instead of designing a specific feature by hand to solve the problem, we define a feature space using our intuition about the problem and let a machine learning algorithm find the best representation. We show how both an object class specific representation and a discriminative recognition model can be learned using the AdaBoost algorithm. This approach allows many different kinds of simple features to be combined into a single similarity function. The method is evaluated using a viewpoint invariant pedestrian recognition dataset and the results are shown to be superior to all previous benchmarks for both recognition and reacquisition of pedestrians.

1 Introduction

Pedestrian tracking is a deceptively hard problem. When the camera is fixed and the number of targets is small, pedestrians can easily be tracked using simple naive methods based on target location and velocity. However, as the number of targets grows, occlusion creates ambiguity. This can be overcome by delaying decisions and considering multiple hypothesis [1] and efficient solutions exist for solving this correspondence problem [2]. However, as the size of the scene itself grows, additional cameras are needed to provide adequate coverage. This creates another problem known as *pedestrian re-identification*. This is a much more challenging problem because of the lack of hard temporal (frame to frame) constraints when matching across non overlapping fields of view in a camera network. However, the ultimate goal of any surveillance system is not to track and reacquire targets, but to understand the scene and provide a more effective interface to the operator. Central to this goal is the ability to search the camera network for a person of interest. This is effectively the same as pedestrian re-identification without any temporal constraints. This problem of *pedestrian recognition* is the main subject of this paper.

Pedestrian recognition presents a number of challenges beyond the tracking problem, most importantly a lack of temporal information. Thus the matching

D. Forsyth, P. Torr, and A. Zisserman (Eds.): ECCV 2008, Part I, LNCS 5302, pp. 262–275, 2008.

decision must be made on the appearance model alone. So what is the best appearance model for pedestrian recognition?

The default option is a simple template, but this representation is viewpoint and pose specific. If the viewpoint angle is known one can compensate using flexible matching and alignment [3] [4] [5], but this will not work well for non-rigid objects. If the problem is limited to a frontal viewpoint then one could fit a triangular graph model [6] or part based model [7] to account for pose change. If multiple overlapping cameras are available it is possible to build a panoramic appearance map [8]. While template methods model the spatial layout of the object, histogram methods model its statistical properties. Histograms have proven useful for an assortment of tasks including tracking [9], texture classification [10], and pedestrian detection [11]. Many attempts have been made to combine the advantages of templates and histograms. Past approaches include recording correlations in correlograms [12], spatial position in spatiograms [13], vertical position in principal axes [14], or scale in multi-resolution histograms [15]. Both template and histogram methods suffer from problems with illumination changes, however it has been shown that this can be compensated for by learning the brightness transfer function between cameras [16].

The appearance model presented in this paper is a hybrid of the template and histogram, however instead of designing the model by hand, machine learning is used to construct a model that provides maximum discriminability for a set of training data. The learned model is an ensemble of localized features, each consisting of a feature channel, location and binning information, and a likelihood ratio test for comparing corresponding features. Once the model has been learned, it provides a similarity function for comparing pairs of pedestrian images. This function can be used for both pedestrian re-identification and recognition. In a practical implementation of the latter case it is expected that

Fig. 1. Some examples from the viewpoint invariant pedestrian recognition (VIPeR) dataset [17]. Each column is one of 632 same-person example pairs.

a human operator would be involved, so we provide both the recognition rate and the expected search time for a human operator.

While there is a great deal of data available for pedestrian detection, training a similarity function for recognition requires multiple images of the same individual. We have chosen to use the VIPeR dataset [17], which contains two views of 632 pedestrians. Some examples of pedestrian image pairs can be found in figure 1. Results for the proposed method are presented in section 4 and shown to far exceed the existing benchmarks.

2 Learning the Similarity Function

Learning domain specific distance or similarity functions is an emerging topic in computer vision [18] [19] [20] [21]. Some have attempted to learn fast approximations [20] to other more computationally expensive functions such as EMD [22]. While others have focused on the features that are found in the process [19]. These approaches can be summarized as follows: AdaBoost is used to sequentially learn computationally inexpensive features to solve a classification problem. Our approach is quite similar in these respects, however our similarity function is domain specific (*i.e.* only applicable to comparing pedestrian images).

The proposed similarity function is a weighted ensemble of likelihood ratio tests, constructed with the AdaBoost algorithm, a brief review of which can be found in Algorithm 1. At each iteration of the algorithm, the feature space is searched for

Algorithm 1: AdaBoost

Given:

- N labeled example training examples (x_i, y_i) where x_i is a pair of pedestrian images and $y_i \in \{-1, 1\}$ denotes if the two image are of the same person.
- A distribution over all training examples: $D_1(i) = 1/N$ for $i = 1 \ldots N$.

For $t = 1, \ldots, T$:

- Find the best localized feature λ_t and model m_t for the current distribution D_t.
- Calculate the edge γ_t

$$\gamma_t = \sum_{i=1}^{N} D_t(i) h(x_i) y_i$$

- If $\gamma_t < 0$ break
- Set $\alpha_t = \frac{1}{2} \ln \frac{1+\gamma_t}{1-\gamma_t}$
- Set $D_{t+1}(i) = \frac{1}{Z_t} D_t(i) \exp\left(-\alpha_t h(x_i) y_{(i)}\right)$, where Z_t is a normalizing factor
- Add α_t, λ_t and m_t to the ensemble

Output the ensemble of weights as A, features as Λ, and models as M.

Fig. 2. The AdaBoost algorithm for learning the similarity function

the best classifier w.r.t. the current distribution and added to the ensemble. In order to keep the problem tractable, we have selected a feature space that can be searched in a reasonable amount of time and is appropriate for the class of input data. While the main objective is to learn an effective similarity function, the size of the classifier is also important. For this reason the input to each classifier is always selected to be a scalar.

2.1 Formulation

The proposed task is to simultaneously learn a set of discriminative features and an ensemble of classifiers. We begin with the following set of definitions. The set of features learned is defined as $\Lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_T\}$. Each feature consists of three elements: a feature channel, a region, and a histogram bin. Denoted as:

$$\lambda =< \text{channel}, (x, y, w, h), (\min, \max) > \tag{1}$$

A specific instance of a pedestrian is defined as $\mathbf{V} = [v_1, v_2, \ldots, v_T]^{\mathrm{T}}$ where each $v_i = p(\lambda_i|I)$, is the probability of a pixel from the specified channel and region being in the specified range. The set of models used to discriminate between two specific instances is defined as $M = \{m_1, m_2, \ldots, m_T\}$, where each m_i denotes the parameters of a likelihood ratio test.

2.2 Feature Channels

A feature channel is any single channel transformation of the original image. Two varieties of feature channel are explored in this paper, color and texture. Eight color channels corresponding to the three separate channels of the RGB YCbCr and HSV[1] colorspace are considered, as well as nineteen texture channels. Two families of texture filters are used, Schmid [23] and Gabor [24]. Each texture channel is the result of convolution with a filter and the luminance channel.

Schmid filters are defined here as:

$$F(r, \sigma, \tau) = \frac{1}{Z} \cos\left(\frac{2\pi\tau r}{\sigma}\right) e^{-\frac{r^2}{2\sigma^2}} \tag{2}$$

Where r denotes the radius, Z is a normalizing constant, and the parameters τ and σ are set to (2,1), (4,1), (4,2), (6,1), (6,2), (6,3), (8,1), (8,2), (8,3), (10,1), (10,2), (10,3), and (10,4) respectively. These filters were originally designed to model rotation invariant texture, but their use here is motivated by the desire to be invariant to viewpoint and pose. Additionally, six Gabor filters are used with parameters set to γ, θ, λ and σ^2 set to (0.3,0,4,2), (0.3,0,8,2), (0.4,0,4,1), (0.4,0,4,1), (0.3,$\frac{\pi}{2}$,4,2), (0.3,$\frac{\pi}{2}$,8,2), (0.4,$\frac{\pi}{2}$,4,1) and (0.4,$\frac{\pi}{2}$,4,1) respectively. All 19 texture filters used here can be seen in figure 3.

Other filters could be added as well, but proved less effective. It has been observed that adding additional features has few drawbacks other than increasing computational and storage requirements. The methodology used to select these specific channels was somewhat haphazard, so it is likely that better feature channels may still be found.

[1] Only one of the luminance (Y and V) channels is used.

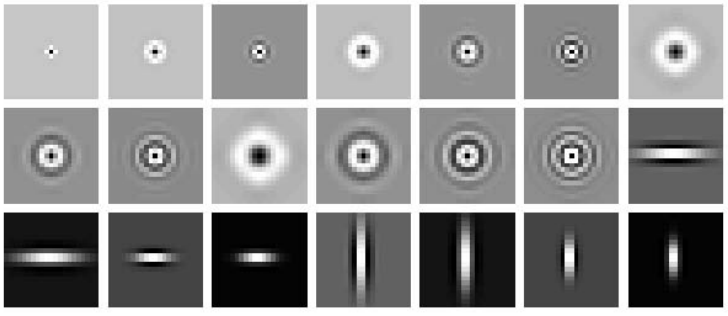

Fig. 3. The filters used in the model to describe texture. (a) Rotationally symmetric Schmid filters. (b) Horizontal and vertical Gabor filters.

2.3 Feature Regions

A feature region could be any collection of pixels in the image, but for reasons of computational sanity they will be restricted to a more tractable subset. Some popular subsets of regions include the simple rectangle, a collection of rectangles [19], or a rectangularly shaped region [7]. The motivation for this has been the computational savings of computing sums over rectangular regions using an integral image. However we can use our intuition about the problem to significantly reduce the number of regions to be considered. Since we know the data consists of pedestrians seen from an arbitrary horizontal viewpoints, we can disregard the horizontal dimension as it is not likely to be relevant, which leaves us with a set of "strips", or rectangles which span the entire horizontal dimension.

2.4 Feature Binning

A feature bin is simply a range over which pixel values are counted. In a traditional histogram, an orthogonal collection of bins is selected to uniformly cover the range of possible values. While this is justified in the general case where the image domain and task are unknown, there is little justification for this approach here, as both computation time and storage can be saved by selecting only the regions of bin space that are relevant to discriminating between pedestrians.

2.5 Feature Modeling

The basis for our similarity function is a collection of likelihood ratio tests performed on the features $\mathbf{V}$. Each test is performed on the value δ, which is defined as the absolute difference between two instances of the same feature:

$$\delta = |v^{(a)} - v^{(b)}| \tag{3}$$

The training data for the proposed approach consists of a collection of pedestrian images. Each individual is seen from two different camera angles, denoted (a) and

(b). δ is computed for every pair of training images between the two cameras. If there are N individuals, then there are N positive training examples denoted Δ_{p}, and $N(N-1)$ negative training examples denoted Δ_{n}. Three possible probability distributions are considered here, Exponential, Gamma, and Gaussian. For each model, the parameters are estimated and a likelihood ratio is computed. This gives three possible weak classifiers for the ensemble.

If δ is distributed as exponential:

$$h(v^{(a)}, v^{(b)}) = \begin{cases} 1 & \text{If } a\delta + b > 0 \\ -1 & \text{Otherwise} \end{cases} \tag{4}$$

Where the coefficients a and b can be expressed in terms of the estimated parameters of the positive and negative distributions as:

$$a = \widehat{\lambda_n} - \widehat{\lambda_p} \qquad b = \ln(\widehat{\lambda_p}) - \ln(\widehat{\lambda_n}) \tag{5}$$

The parameters of an exponential distribution can be estimated as $\widehat{\lambda} = \frac{1}{\hat{\mu}}$.

If δ is distributed as gamma:

$$h(v^{(a)}, v^{(b)}) = \begin{cases} 1 & \text{If } a\delta + b\ln\delta + c > 0 \\ -1 & \text{Otherwise} \end{cases} \tag{6}$$

Where the coefficients a, b and c can be expressed in terms of the estimated parameters of the positive and negative distributions as:

$$\begin{gathered} a = \hat{\beta}_n - \hat{\beta}_p \qquad b = \hat{\alpha_p} - \hat{\alpha_n} \\ c = \hat{\alpha_p}\ln\hat{\beta}_p - \hat{\alpha_n}\ln\hat{\beta}_n + \ln\Gamma(\hat{\alpha_n}) - \ln(\Gamma\hat{\alpha_p}) \end{gathered} \tag{7}$$

The parameters of a gamma distribution can be estimated as $\hat{\alpha} = \frac{\hat{\mu}^2}{\hat{\sigma}^2}$ and $\hat{\beta} = \frac{\hat{\mu}}{\hat{\sigma}^2}$.

If δ is distributed as Gaussian:

$$h(v^{(a)}, v^{(b)}) = \begin{cases} 1 & \text{If } a\delta^2 + b\delta + c > 0 \\ -1 & \text{Otherwise} \end{cases} \tag{8}$$

Where the coefficients a, b and c can be expressed in terms of the estimated parameters of the positive and negative distributions as:

$$\begin{gathered} a = \frac{1}{2\hat{\sigma_n^2}} - \frac{1}{2\hat{\sigma_p^2}} \qquad b = \frac{\hat{\mu_p}}{\hat{\sigma_p^2}} - \frac{\hat{\mu_n}}{\hat{\sigma_n^2}} \\ c = \frac{\hat{\mu_n^2}}{2\hat{\sigma_n^2}} - \frac{\hat{\mu_p^2}}{2\hat{\sigma_p^2}} + \ln(\hat{\sigma_n^2}) - \ln(\hat{\sigma_p^2}) - \ln(2\pi) \end{gathered} \tag{9}$$

2.6 Search Strategy

At each iteration we must find the best feature for the current distribution. The traditional approach is to define the feature space to be small enough that an exauhstive search is possible at each iteration. The size of the feature space proposed here is the product of the number of possible channels, regions, bins, and models. While the number of possible channels and models is relatively

small (21 and 3 respectively), the number of possible contiguous regions and bins grows quadratically with the number of quantization levels. Thus without any quantization, the total search space would be $|\mathcal{F}| \approx 10^{10}$.

We have found the following steps greatly improved training time. First, we precompute an intermediate feature representation for every image before training. This feature is a quantized two dimensional map of each channel of each image. The two dimensions of this map are the quantized y coordinate and pixel value. This map is then transformed into an integral image, allowing for any histogram bin to be calculated over any set of vertical strips in constant time using the integral image trick. Second, a coarse to fine search strategy is used to explore the parts of the feature space that we believe to be smooth (*e.g.* the region and binning space). As a result of these search strategies the search time has been reduced from hours to minutes.

2.7 What Is the Model Being Learned?

The usual approach to solving a problem such as this is for the researcher to hand craft a feature representation that appears appropriate for the class of data and then select the distance function or classifier that provides the best results. For example Park *et al.* noticed that people often wear different color shirt and pants, and thus defined their feature representation to be three histograms taken over the head, shirt and pants regions of a person [25]. Hu *et al.* noticed that the principal axis is often different among different pedestrians and choose a model accordingly [14]. Gheissari *et al.* have taken the extreme approach of designing a 2d mesh model of a pedestrian in order to obtain an exact correspondence between frontal pedestrian images [6].

As researchers we never really know what the *correct* model to use in any particular problem is. However we have a great deal of intuition about how we as humans would solve the problem. What we have done here is use our intuition to define a broad (*i.e.* intentionally vague) feature space that we believe contains a good feature representation, and then allowed the AdaBoost algorithm to build the best model for the training data available. In this case, the model is a collection of simple color and texture features and some spatial and intensity information.

3 Modeling Pedestrian Recognition Performance

3.1 Evaluation Data

The experimental setups used to evaluate pedestrian recognition have varied widely. Gandhi and Trivedi provide results on 10 individuals seen simultaneously from 4 opposing viewpoints which are used to create a panoramic image map [8]. This setup is ideal for tracking, but very costly to implement. Gheissari et al. collected 3 frontal views of 44 unique individuals [6] and Wang et al. later added 99 additional individuals to that dataset [7]. While their data contains a diverse set of people and poses, the lack of viewpoint variation is insufficient to

model the real world recognition problem. Gray et al. have collected two views of 632 individuals seen from widely differing viewpoints [17]. In contrast to the aforementioned data, *most* of their examples contain a viewpoint change of 90 degrees or more, making recognition very challenging. The method presented in this paper is evaluated using their public dataset. Some examples from this dataset can be found in figure 1.

3.2 Evaluation Methodology

Several approaches have been used for evaluating recognition and re-identification performance. Shan *et al.* has treated the re-identification problem as a same-different detection problem and provided results using a receiver operating characteristic (ROC) curve [4]. Wang *et al.* treat the problem as recognition and provide results using a cumulative matching characteristic (CMC) curve. Gray *et al.* provide results using a CMC curve, but also present a method of converting their results into a re-identification rate. This paper presents results in the form of a recognition rate (CMC), re-identification rate, and expected search time by a human operator.

The evaluation methodology used here is as follows. The training data is split evenly into a training and test set. Each image pair is split and randomly assigned to camera a and camera b. The CMC curve for the test set is calculated by selecting a probe (image from camera a) and matched with a gallery (every image in camera b). This provides an ranking for every image in the gallery w.r.t the probe. This procedure is repeated for every image in camera a and averaged. Camera a and camera b are then swapped and the process is repeated. The CMC curve is then the expectation of finding the correct match in the top n matches.

The expected search time is defined as the expected amount of time required for a human operator to find the correct match from a collection of images. If we assume the human operator makes no mistakes, then we can decomposed the expected search time into three components:

$$\text{E[Search Time]} = \frac{\text{E[Sort Position]}}{\text{Dataset Size}} \times \text{E[Time per Image]} \times \text{Dataset Size}$$

If the system has no prior knowledge about the probable association between the probe and gallery images, then the images will be presented in random order and the first term will be 0.5. For the sake of simplicity, we will assume the second term is one second, making the expected search time and sort position the same value.

Thus there are three ways to reduce the expected search time. The human operator could take less time per image at the expense of missing the correct match. The size of the dataset could be reduced using other information (eg. spatial or temporal information about camera position). Or the operator could be presented with the data sorted by some similarity function. This reduction in search time may be small when the dataset size is small (ie. in laboratory testing), however in a real world scenario this number could be quite large and the time savings could be very significant.

4 Results

4.1 Benchmarks

We compare our results to 4 different benchmark methods. A simple template (SSD matching), a histogram, a hand localized histogram with the configuration proposed by Park *et al.* [25], and a principal axis histogram, which is similar in spirit to the work of Hu *et al.* [14]. Multiple configurations were tried for each method, but only the best results for each are shown here. We found that 16 quantization levels, YCbCr colorspace and the Bhattacharyya distance performed best for all three histogram methods. The key differences between the three approaches are the regions over which histograms are computed. In the hand localized histogram, three regions are chosen to correspond to the head (top $\frac{1}{5}$), shirt (middle $\frac{2}{5}$) and pants (bottom $\frac{2}{5}$)). In the principal axis histogram 32 regions are chosen in 4 pixel increments to cover each horizontal stripe of the image. The ELF model shown here contains 200 features and will be analyzed in greater detail in section 4.4.

Comparisons with the methods proposed in [6] and [7] are desirable, but not practical given the complexity of these methods. Additionally, these two methods were designed for frontal viewpoints, and would likely give poor results on the data used here because of the wide viewpoint changes.

4.2 Recognition

As was mentioned in the beginning of this paper, pedestrian recognition is a hard problem. Figure 4 shows an example of 16 probe images, the top 28 matches using our similarity function, and the correct match. Given a single image, finding its counterpart from a gallery of 316 images is quite challenging for a human operator. We challenge the reader to find the correct matches in this sorted gallery without looking at the key in the caption or the corresponding image on the right.

Figure 5 shows recognition performance as a CMC curve. The rank 1 matching rate of our approach is around 12%, while the correct match can be found in the top 10% (rank 31) around 70% of the time. The utility of these recognition rates can be summarized by looking at the expected search times in figure 6. Without any similarity measure, a user would have to look at half the data on average before finding the correct match. This could take quite some time for a large number of images. At this task, the ELF similarity function yields an 81.7% search time reduction over a pure human operator, and a 58.2% reduction over the next best result.

4.3 Re-identification

The difficulty in pedestrian re-identification varies with the number of possible targets to match. Figure 7 shows how the re-identification performance of the different approaches performs as the number of possible targets increases.

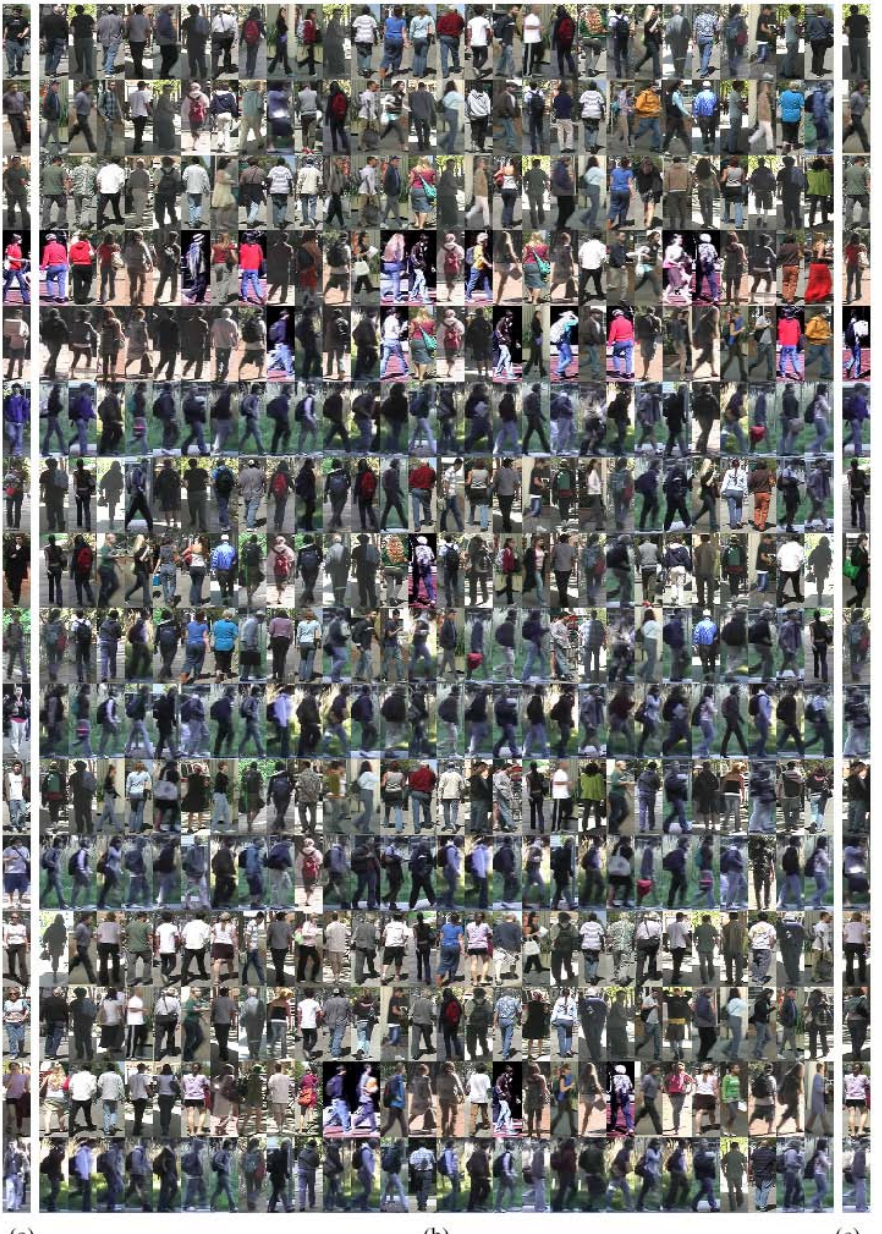

Fig. 4. Example queries to a recognition database. (a) Probe image. (b) Top n results (sorted left to right). (c) Correct match. Note the visual similarity of the returned results and ambiguity created by pose, viewpoint and lighting variations. The correct match for these examples was ranked 2, 2, 1, 3, 39, 2, 2, 196, 1, 33, 16, 55, 3, 45, 6 and 18 respectively from a gallery of 316 people (top to bottom).

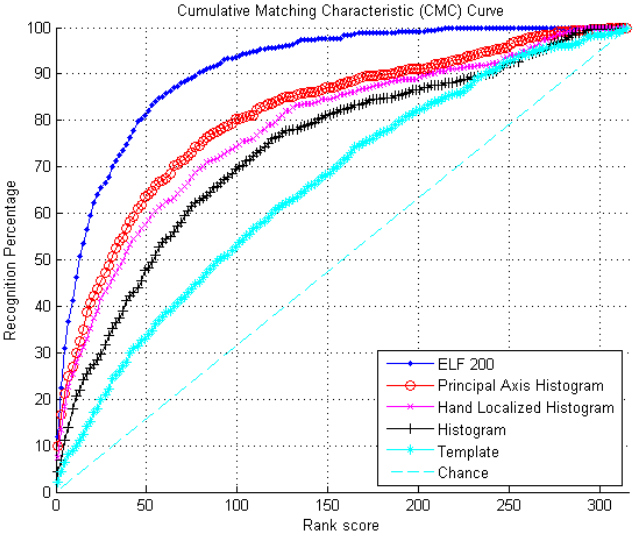

Fig. 5. Cumulative matching characteristic (CMC) curve for ELF model and benchmarks

Method	Expected Search Time (s)
Chance	158.0
Template	109.0
Histogram	82.9
Hand Localized Histogram	69.2
Principal Axis Histogram	59.8
ELF	28.9

Fig. 6. Expected search time for ELF model and benchmarks. This assumes a human operator can review 1 image per second at 100% accuracy.

When the number of possible targets is small, performance is very high. The re-identification task is rarely performed with appearance models alone. In most indoor or otherwise restricted camera networks, spatial information can be used to restrict the number of matches to be quite small, making these results very promising considering that spatial and temporal information can be combined with appearance information to great effect as these cues are independent.

4.4 Model Analysis

One of the strengths of this approach is the ability to combine many different kinds of features into one similarity function. Figure 8 shows the percentage of weight accorded to each feature channel or family of features. It is not surprising given the illumination changes between the two cameras, that the two most informative channels are hue and saturation. Roughly three quarters of the weight

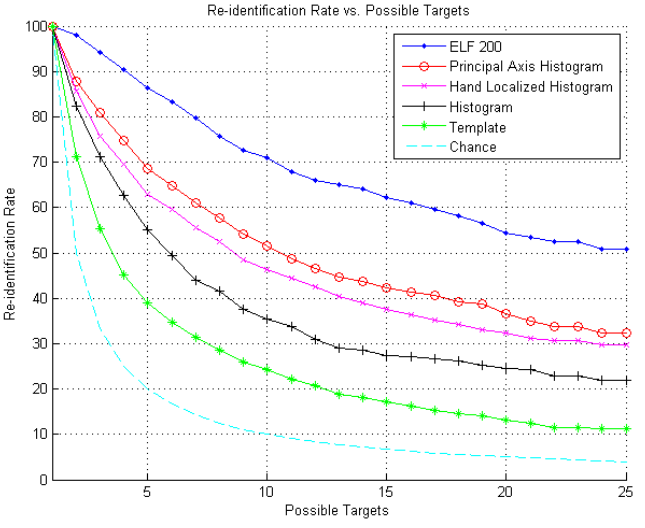

Fig. 7. Re-identification rate *vs.* number of targets for ELF model and benchmarks

Feature Channel	Percent of classifier weight
R	11.0 %
G	9.4 %
B	12.4 %
Y	6.4 %
Cb	6.1 %
Cr	4.5 %
H	14.2 %
S	12.5 %
Schmid	12.0 %
Gabor	11.7 %

Fig. 8. A table showing the percent of features from each channel, model

of the classifier is devoted to color features, which seems to suggest that past approaches which relied on color histograms alone were justified.

5 Conclusions

We have presented a novel approach to viewpoint invariant pedestrian recognition that learns a similarity function from a set of training data. It has been shown that this ensemble of localized features is effective at discriminating between pedestrians regardless of the viewpoint change between the two views. While the automatic pedestrian recognition problem remains unsolved, it has been shown that the proposed approach can be used to assist a human operator in this task by significantly reducing the search time required to match

pedestrians from a large gallery. While the ultimate goal of automated surveillance research is to remove the human entirely, this work represents a significant improvement over past approaches in reducing the time required to complete simple surveillance tasks such as recognition and re-identification.

References

1. Reid, D.: An algorithm for tracking multiple targets. Automatic Control, IEEE Transactions on 24(6), 843–854 (1979)
2. Cox, I., Hingorani, S., et al.: An efficient implementation of Reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence 18(2), 138–150 (1996)
3. Guo, Y., Hsu, S., Shan, Y., Sawhney, H.: Vehicle fingerprinting for reacquisition & tracking in videos. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2 (2005)
4. Shan, Y., Sawhney, H., Kumar, R.: Vehicle Identification between Non-Overlapping Cameras without Direct Feature Matching. In: IEEE International Conference on Computer Vision, vol. 1 (2005)
5. Guo, Y., Shan, Y., Sawhney, H., Kumar, R.: PEET: Prototype Embedding and Embedding Transition for Matching Vehicles over Disparate Viewpoints. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2007)
6. Gheissari, N., Sebastian, T., Hartley, R.: Person Reidentification Using Spatiotemporal Appearance. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 1528–1535 (2006)
7. Wang, X., Doretto, G., Sebastian, T., Rittscher, J., Tu, P.: Shape and appearance context modeling. In: IEEE International Conference on Computer Vision, pp. 1–8 (2007)
8. Gandhi, T., Trivedi, M.: Person tracking and reidentification: Introducing Panoramic Appearance Map (PAM) for feature representation. Machine Vision and Applications 18(3), 207–220 (2007)
9. Comaniciu, D., Ramesh, V., Meer, P.: Real-time tracking of non-rigid objects using mean shift. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2 (2000)
10. Varma, M., Zisserman, A.: A Statistical Approach to Texture Classification from Single Images. International Journal of Computer Vision 62(1), 61–81 (2005)
11. Dalai, N., Triggs, B., Rhone-Alps, I., Montbonnot, F.: Histograms of oriented gradients for human detection. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1 (2005)
12. Huang, J., Ravi Kumar, S., Mitra, M., Zhu, W., Zabih, R.: Spatial Color Indexing and Applications. International Journal of Computer Vision 35(3), 245–268 (1999)
13. Birchfield, S., Rangarajan, S.: Spatiograms versus Histograms for Region-Based Tracking. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2 (2005)
14. Hu, W., Hu, M., Zhou, X., Lou, J.: Principal Axis-Based Correspondence between Multiple Cameras for People Tracking. In: IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28(4) (2006)
15. Hadjidemetriou, E., Grossberg, M., Nayar, S.: Spatial information in multiresolution histograms. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 702–709 (2001)

16. Javed, O., Shafique, K., Shah, M.: Appearance Modeling for Tracking in Multiple Non-overlapping Cameras. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 26–33 (2005)
17. Gray, D., Brennan, S., Tao, H.: Evaluating Appearance Models for Recognition, Reacquisition, and Tracking. In: IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS) (2007)
18. Hertz, T., Bar-Hillel, A., Weinshall, D.: Learning distance functions for image retrieval. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2 (2004)
19. Dollar, P., Tu, Z., Tao, H., Belongie, S.: Feature Mining for Image Classification. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2007)
20. Athitsos, V., Alon, J., Sclaroff, S., Kollios, G.: Boostmap: An embedding method for efficient nearest neighbor retrieval. IEEE Transactions on Pattern Analysis and Machine Intelligence 30(1), 89–104 (2008)
21. Yu, J., Amores, J., Sebe, N., Radeva, P., Tian, Q.: Distance learning for similarity estimation. IEEE Transactions on Pattern Analysis and Machine Intelligence 30(3), 451–462 (2008)
22. Rubner, Y., Tomasi, C., Guibas, L.: The Earth Mover's Distance as a Metric for Image Retrieval. International Journal of Computer Vision 40(2), 99–121 (2000)
23. Schmid, C.: Constructing models for content-based image retrieval. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2 (2001)
24. Fogel, I., Sagi, D.: Gabor filters as texture discriminator. Biological Cybernetics 61(2), 103–113 (1989)
25. Park, U., Jain, A., Kitahara, I., Kogure, K., Hagita, N.: ViSE: Visual Search Engine Using Multiple Networked Cameras. In: IEEE International Conference on Pattern Recognition, 1204–1207 (2006)