

Context-aided Human Recognition - Clustering

Yang Song and Thomas Leung

Fujifilm Software (California), Inc.
1740 Technology Drive, Suite 490, San Jose, CA 95110, USA
{ysong,tleung}@fujifilmsoft.com

Abstract. Context information other than faces, such as clothes, picture-taken-time and some logical constraints, can provide rich cues for recognizing people. This aim of this work is to automatically cluster pictures according to person's identity by exploiting as much context information as possible in addition to faces. Toward that end, a clothes recognition algorithm is first developed, which uses color and texture information and is effective for different types of clothes (smooth or highly textured). Clothes recognition results are integrated with face recognition to provide similarity measurements for clustering. Picture-taken-time is used when combining faces and clothes, and the cases of faces or clothes missing are handled in a principle way. A spectral clustering algorithm which can enforce hard constraints (positive and negative) is presented to incorporate logic-based cues (e.g. two persons in one picture must be different individuals) and user feedback. Experiments on real consumer photos show the effectiveness of the algorithm.

1 Introduction

Being able to identify people is important for automatic organizing and retrieving photo albums and for security applications, where face recognition has been playing a major role. But reliable face recognition is still a challenging problem after many research efforts [5], especially when imaging condition changes. On the other hand, information besides faces (called 'context' relative to face) can provide rich cues for recognizing people.

Generally speaking, there are three types of context information. The first type is appearance-based, such as a person's hair style or the clothes he is wearing; the second type is logic-based, for instance, different faces in one picture belong to different persons or some people are more likely to be pictured together (e.g. husband and wife); the third type is the meta-data for pictures such as the picture-taken-time. This context information is often used by human observers consciously or unconsciously. It is very tempting to investigate how to build algorithms which can utilize this context information effectively to improve human recognition accuracy.

The aim of this work is to automatically organize pictures according to person's identity by using faces and as much context information as possible. Assuming we have a face recognition engine, we want to improve upon it via contexts.

We want to develop a clustering algorithm which can put persons in the pictures into groups (clusters). The ideal results will be that all the images of the same individual are in one cluster and images from different individuals are in different clusters. Towards this end, we need to answer the following three questions: 1) what context information to use? 2) what is the clustering algorithm? 3) how to put context information into the clustering algorithm?

Regarding to the first question, we use the appearance-based and logic-based context explicitly, and the picture taken time implicitly. For the appearance-based context, clothes provide an important cue for recognizing people in the same event (or on the same day) when clothes are not changed. They are complimentary to faces and remain very useful when face pose changes, poor face quality, and facial expression variations occur. Therefore, it is intuitively appealing to use clothes information. However, in practice, due to different types of clothes (solid colored or heavily textured) and changes in clothes imaging condition (occlusions, pose changes, lighting changes, etc), it is not a trivial matter to use clothes information effectively. We strive to develop an effective clothes recognition method in this paper. For the logic-based context, we want to enforce some hard constraints. A constraint is hard when it must be satisfied in order for a clustering result to be correct. For example, the fact that different faces in one picture belonging to different individuals is a hard constraint.

Many clustering algorithms have been developed, from traditional K-means to the recently popular spectral clustering ([10, 14, 8, 15]). One major advantage of spectral clustering methods over K-means ([8]) is that K-means easily fails when clusters do not correspond to convex regions (similar for mixture of models using EM, which often assumes that the density of each cluster is Gaussian). In human clustering, imaging conditions can change from different aspects, hence one cluster doesn't necessarily form a convex region. Therefore a spectral clustering algorithm is favored.

Now we are facing the question of how to put the context information into the clustering algorithm. The base of a spectral clustering algorithm is the similarity measure between nodes (for human recognition, each node represents a person image). It is a natural thought to combine clothes recognition results with face recognition results as the similarity measurements. But due to occlusion or pose changes, either face or clothes information may be missing or when different people wear the same clothes on the same day, the clothes information can become unreliable. We propose a principled way to handle these cases. The next issue is how to enforce the hard constraints? For K-means, hard constraints can be enforced as in [13]. Though spectral clustering methods have the aforementioned advantage over K-means, it is hard to enforce hard constraints. In [15], a solution of imposing positive constraints (two nodes must belong to the same cluster) is addressed, but there is no guarantee that the positive constraints will be respected and the problem of enforcing negative constraints (two nodes cannot belong to the same cluster) remains open. In this paper, by taking advantages of both K-means and spectral clustering methods, we devise a spectral clustering method which can enforce hard constraints.

In [18], clothes information is used for annotating faces. Our work differs from that in (1) a new clothes recognition algorithm is developed, and the results from face and clothes recognition are integrated in a principled way; (2) a constrained spectral clustering algorithm, which can enforce hard constraints, is proposed, so that other context cues (e.g. persons from one picture should be in different clusters) and user feedback can be imposed.

The rest of the paper is organized as follows. The clothes recognition method is presented in Section 2. Section 3 describes how to combine clothes recognition results with face recognition into one similarity measurement. Section 4 depicts the spectral clustering algorithm and how to put some logic-based context cues (i.e. enforcing hard constraints) into the clustering algorithm. Experimental results are presented in Section 5. Finally, Section 6 gives concluding remarks.

2 Clothes Recognition

Clothes recognition is to judge how similar two pieces of clothes image are and therefore to indicate how likely they are from the same individual. There are three major steps for clothes recognition: clothes detection and segmentation, clothes representation (or feature extraction), and similarity computation based on extracted features.

2.1 Clothes Detection and Segmentation

Clothes detection and segmentation is to obtain the clothes part from an image. For recognition purpose, precise contours of clothes are not necessary, but we need to get the representative part and get rid of clutters.

An initial estimation of the clothes location can be obtained by first running face detection ¹ and taking some parts below the head. However, this is often unsatisfactory due to occlusion by another person or by the person’s self limbs (skin) or presence of other objects in the environment. To improve upon the initial estimations, the following two steps are therefore performed. One is to segment clothes among different people via maximizing the difference of neighboring clothes pieces, which can be computed by the χ^2 distance of color histograms in CIElab space. Assuming that the ‘true’ clothes are not far away from the initial guess, candidate locations can be obtained by shifting and re-sizing the initial estimation. The candidates which can maximize the difference are chosen. Figure 1 shows an example.

The next step is to get rid of clutters not belonging to clothes. Clutters are handled in two ways. For predictable clutters like human skin, a common cause of occlusion, we build a skin detector using techniques similar to what described

¹ Here we obtain a quick initial guess of the clothes location from face detection. Face detection [9, 12, 2] can currently achieve better accuracy than face recognition so results derived from face detection can be complimentary to face recognition results. For example, profile faces can be detected (so are the corresponding clothes), but they present a challenge for state-of-the-art face recognition algorithms.



Fig. 1. (a) initial estimation from face detection (shown by the dashed yellow lines, small red circles show the eye positions); (b) refined segmentation by maximizing the difference between people (shown by the solid green lines).

in next section. More details on skin detection will be given in Section 2.4. For more random clutters not persistent across pictures, we diminish their influence in the feature extraction step (Section 2.2).

2.2 Clothes Representation (or Feature Extraction)

After extracting clothes from an image, the next issue is to represent it quantitatively: clothes representation (or feature extraction). In the literature, there are generally two types of features being extracted: local features and global features. Local features have recently received a lot of research attention (such as [6, 1, 7, 11]) and have been successfully used in some recognition systems. However, most local features are selected based on some kind of local extrema (e.g. with 'maximum entropy' or 'maximum change'), which cannot work if the clothes under consideration is a smooth colored region without textures or patterns (e.g. a single-colored T-shirt). Then how about global features like color histogram and/or orientation histogram? Color histogram suffers when lighting changes. Clothes are often folded and therefore create false edges and self-shadows, which create difficulties for orientation histograms. Thus some more effective features are desired. To take advantage of global representations (which can be more robust to pose changes), the features extracted will be histograms of 'something'. But unlike color histograms or orientation histograms, we want the 'something' to be representative patches for clothes under consideration and to exclude random clutters. In order to achieve that, we devise the following feature extraction method - the representative patches are learned automatically from a set of clothes.

The method uses code-word (representative patches) frequency as feature vectors. The code-words are learned as follows. Overlapped small image patches (e.g. 7x7 pixel patches with two neighboring patches 3 pixels apart) are taken from each normalized clothes piece (according to the size of faces - from face detection module). All the patches from all the clothes pieces in the image set are gathered. If a small patch is of 7x7 pixels, and the total number of small patches is N , we have N 147-dimensional (3 color channels for each pixel) vectors.

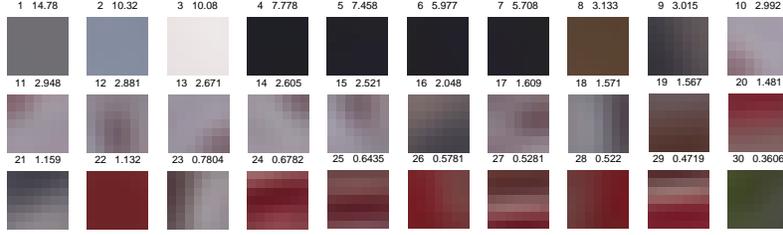


Fig. 2. Examples of code-words obtained. The occurrence frequency of these code-words in a clothes piece is used as the feature vector.

In order to get rid of noise and make the computation efficient, principle component analysis (PCA) is used to reduce the dimensionality of these vectors. Each small patch is represented by projections under the first k (we use $k = 15$) principle components. Vector quantization (e.g. K-means clustering) is then run on these N k -dimensional vectors to obtain code-words. The Mahalanobis distance, given by $d(x_1, x_2) = \sqrt{(x_1 - x_2)^T \Sigma^{-1} (x_1 - x_2)}$ for any two vectors x_1 and x_2 (where Σ is the covariance matrix), is used for K-means clustering. The number of code-words (i.e. the number of clusters for K-means) can vary according to the complexity of the data. 30 code-words are used in our experiments. Figure 2 shows code-words obtained (i.e. centers of k-means clustering) for the image set including the image in Figure 1.

By vector quantization, each small patch is quantized into one of the code-words, and one clothes piece can be represented by the vector describing the frequency of these code-words. Suppose that the number of code-words is C , then this code-word frequency vector is C -dimensional, $V_{thiscloth} = [v_1, \dots, v_i, \dots, v_C]$, with each component $v_i = \frac{n_i^{thiscloth}}{n^{thiscloth}}$, where $n_i^{thiscloth}$ is the number of occurrence of code-word i in the clothes piece and $n^{thiscloth}$ is the total number of small patches in the clothes piece.

The above feature extraction method has the following advantages for clothes recognition. 1) The clustering process selects consistent features as representative patches (code-words) and is more immune to background clutters which are not consistently present since small image patches from non-persistent background are less likely to form a cluster. 2) It uses color and texture information at the same time, and it can handle both smooth and highly textured regions. 3) Code-word frequency counts all the small patches and does not rely on any particular features. Hence it can handle pose changes to a certain degree. 4) Compared to color histograms, it is more robust to lighting changes. Image patches corresponding to the same clothes part can have different appearance due to lighting changes. For example, a green patch can have different brightness and saturation. Through PCA dimension reduction and using Mahalanobis distance, these patches are more likely to belong to the same cluster than to the same color bin for color histogram.

2.3 Similarity Computation

The similarity between two pieces of clothes is computed in a way similar to [11]. Each component of the code-word frequency vector is multiplied by $\log(\frac{1}{w_i})$, where w_i is the percentage of small patches quantized into code-word i among all the N patches. By putting these weights, higher priorities are given to those code-words (features) occurring less frequently. This is based on the idea that less frequent features can be more distinctive therefore more important.

The similarity score of two pieces of clothes is given by the normalized scalar product (cosine of angle) of their weighted code-word frequency vectors.

2.4 Skin Detection

As described in section 2.1, skin is a common type of clutter. However, general skin detection is not a trivial matter due to lighting changes. Fortunately for a set of images, skin from faces and from limbs usually looks similar. Therefore a skin detector can be learned from faces.

Learning Skin Code-words from Faces. The representative skin patches (code-words for skin detection) are learned from faces. First, small skin patches are obtained from faces (majorly cheek part). Each small skin patch is represented by the mean of each color channel. K-means clustering are then performed on these 3-dimensional vectors. The centers from k-means clustering form the code-words for skin detection.

Detect Skin in Clothes. In order to decide whether a small patch is skin or not, we first get its mean of three color channels, and then compute its Mahalanobis distance to each code-word. If the smallest distance is less than a pre-defined threshold and the patch satisfies certain smoothness criterion, the patch is taken as skin. The smoothness of a patch is measured by the variance of luminance. Only those non-skin patches will be used for further computation.

3 Integrating clothes context with face recognition

The clothes recognition scheme presented in the previous section tells how similar a pair of clothes pieces are. To achieve higher human recognition accuracy, clothes cues are to be integrated with face cues. These combination results provide similarity measurements for clustering (section 4).

For any pair of person images, let x_f be the score from face recognition (e.g. [5]), x_c be the score from clothes recognition. Let random variable Y indicate whether the pair is from the same person or not: $Y = 1$ means from the same person and $Y = 0$ means otherwise. We want to estimate the probability of the pair belonging to the same individual given certain face and clothes scores $P(Y = 1|x_f, x_c)$. In linear logistic regression,

$$P(Y = 1|x_f, x_c) = \frac{1}{1 + \exp(-w_f x_f - w_c x_c - w_0)} \quad (1)$$

where $\bar{w} = [w_f, w_c, w_0]$ are parameters to be learned. The best \bar{w} , which maximizes the log-likelihood of a set of training examples, can be obtained iteratively through Newton-Raphson's method.

In testing, for any pair of face recognition and clothes recognition scores, we plug them into equation (1), and get $P(Y = 1|x_f, x_c)$, i.e., the probability of being from the same person. Other cue combination algorithms, such as using Fisher linear discriminant analysis and mixture of experts ([4]), were also experimented. They gave close results for our application though the mixture of experts method is potentially more powerful. Linear logistic regression is adopted here because it is simple and works well. It also provides a good way for handling the cases of face or clothes information missing.

3.1 Recognition when face or clothes are missing

While one advantage of using clothes context is to help improve human recognition accuracy, another is that it makes human recognition possible when face recognition results are unavailable (e.g. faces are occluded or profile to back view of faces). Clothes information can also be missing due to occlusion or become unreliable for images taken on different days (events) or when different people in the same picture wearing the same clothes. Hence we need to handle the case of face or clothes information missing. The similarity measurements under all the situations (with face recognition only, clothes recognition only, and face and clothes combined) need to be compatible so that they can be compared directly and fairly.

Using the same notations as in the previous section, when face or clothes scores are missing, $P(Y = 1|x_c)$ or $P(Y = 1|x_f)$ needs to be computed. The compatibility requirement is satisfied if $P(Y = 1|x_f)$ and $P(Y = 1|x_c)$ are the marginal probabilities of $P(Y = 1|x_f, x_c)$. By Bayesian rule and equation (1),

$$P(Y = 1|x_c) = \int_{x_f} \frac{1}{1 + \exp(-w_f x_f - w_c x_c - w_0)} P(x_f|x_c) dx_f$$

If we assume that $x_f = C \cdot x_c + C_0$ for some constant C and C_0 , i.e., $P(x_f|x_c) = \delta(x_f - Cx_c - C_0)$, then

$$\begin{aligned} P(Y = 1|x_c) &= \frac{1}{1 + \exp(-w_f \cdot C \cdot x_c - w_f \cdot C_0 - w_c x_c - w_0)} \\ &= \frac{1}{1 + \exp(-w'_c x_c - w'_0)} \end{aligned} \quad (2)$$

Therefore, $P(Y = 1|x_c)$ is also in the form of a logistic function, so does $P(Y = 1|x_f)$. The parameters of these logistic functions such as w'_c , and w'_0 can be estimated in a similar fashion to those of equation (1).

Note that equation (2) is derived assuming that face scores are a linear function of clothes scores so that only clothes information determines the similarity between a pair of person images. This could be a reasonable assumption when face information missing. We tested the compatibility of computed $P(Y = 1|x_f, x_c)$, $P(Y = 1|x_f)$ and $P(Y = 1|x_c)$ in experiments.

3.2 Handling the case of people wearing the same clothes

People wearing the same (or similar) clothes poses difficulties for incorporating clothes information. Two persons in one picture usually are not the same individual. Thus if in one picture, two persons wear the same (or similar) clothes, we need to discard the clothes information. The clothes information also becomes possibly misleading when the pair-wise similarity between other clothes pieces and either of those two is high. The clothes information is therefore treated as missing for these cases, and similarities are computed as in section 3.1.

4 Human Clustering with Hard Constraints

The previous sections depict a clothes recognition algorithm as well as how to integrate clothes context with faces into one similarity measure. These pair-wise similarity measurements provide grounds for clustering. This section focuses on the clustering algorithm and how to put logic-based contexts (such as some hard constraints) into clustering.

4.1 Spectral Clustering

Spectral clustering methods cluster points by eigenvalues and eigenvectors of a matrix derived from the pair-wise similarities between points. Spectral clustering is often looked as a graph partitioning problem: each point is a node in the graph and similarity between points gives weight of the edge. In human clustering, each point is a person's image, and similarity measurements are from face and/or clothes recognition.

One effective spectral clustering method used in computer vision is normalized cuts [10], with generalization in [15]. The normalized cuts criterion is to maximize links (similarities) within each cluster and to minimize links between clusters. Suppose that we have a set of points $S = \{s_1, \dots, s_N\}$, and we want to cluster them into K clusters. Let W be the $N \times N$ weight matrix with each term W_{ij} being the similarity between points s_i and s_j , and let D denote the diagonal matrix with the i -th diagonal element being the sum of W 's i th row (i.e. the degree for the i th node). The clustering results can be represented by a $N \times K$ partition matrix X , with $X_{ik} = 1$ if and only if point s_i belongs to the k th cluster and 0 otherwise. Let X_l denote the l th column vector of X , $1 \leq l \leq K$. X_l is the membership indicator vector for the l th cluster. Using this notations, the normalized cut criterion is to find the best partition matrix X^* which can maximize $\varepsilon(X) = \frac{1}{K} \sum_{l=1}^K \frac{X_l^T W X_l}{X_l^T D X_l}$.

Relaxing the binary partition matrix constraint on X and using Rayleigh-Ritz theorem, it can be shown that the optimal solution in the continuous domain are derived through the K largest eigenvectors of $D^{-1/2} W D^{-1/2}$. Let v_i be the i th largest eigenvector of $D^{-1/2} W D^{-1/2}$, and $V^K = [v_1, v_2, \dots, v_K]$. Then the continuous optimum of $\varepsilon(X)$ can be achieved by X_{conti}^* , the row normalized version of V^K (each row of X_{conti}^* has unit length). In fact, the optimal solution is not

unique - the optima are a set of matrices up to an orthonormal transformation: $\{X_{conti}^* O : O^T O = I_K\}$, where I_K is the $K \times K$ identity matrix.

In [15], a repulsion matrix is introduced to model the dissimilarities between points. The clustering goal becomes to maximize within-cluster similarities and between-cluster dissimilarities, but to minimize their compliments. Let A be the matrix quantifying similarities (affinity matrix), R be the matrix representing dissimilarities (repulsion matrix), and D_A and D_R be the diagonal matrices corresponding to the row sum of A and R respectively. Define $\hat{W} = A - R + D_R$ and $\hat{D} = D_A + D_R$, then the goal is to find the partition matrix X which can maximize $\frac{1}{K} \sum_{l=1}^K \frac{X_l^T \hat{W} X_l}{X_l^T \hat{D} X_l}$. The continuous optima can be found through the K largest eigenvectors of $\hat{D}^{-1/2} \hat{W} \hat{D}^{-1/2}$ in a similar fashion to the case of without a repulsion matrix.

Since a continuous solution can be found by solving eigensystems, the above methods are fast and can achieve global optimum in the continuous domain. However, for clustering, a continuous solution needs to be discretized. In [15], discretization is done iteratively to find the binary partition matrix $X_{discrete}^*$ which can minimize $\|X_{discrete} - X_{conti}^* O\|^2$, where $\|M\|$ is the Frobenius norm of matrix M : $\|M\| = \sqrt{tr(MM^T)}$, O is any orthonormal matrix, and $X_{conti}^* O$ is a continuous optimum.

4.2 Incorporating more context cues: enforcing hard constraints

Some logic-based contexts can be expressed as hard constraints, e.g., one useful negative hard constraint is that different persons in one picture should be different individuals. It is desirable to be able to enforce these constraints in human clustering. However, incorporating priors (such as hard constraints) poses a challenge for spectral clustering algorithms. In [16, 15], a method to impose positive constraints (two points must belong to the same cluster) was proposed, but the constraints may be violated in the discretization step. To the best of our knowledge, there is no previous work which can enforce negative hard constraints (two points cannot be in the same cluster) in spectral clustering methods. This section explores how to enforce hard constraints, negative as well as positive.

Using the same notations as in section 4.1, if s_i and s_j are in the same picture, we want to make sure s_i and s_j are in different clusters. To achieve that, the corresponding term in the affinity matrix A_{ij} is set to be zero. A repulsion matrix R is also used to enhance the constraints: R_{ij} is set to be 1 if s_i and s_j cannot be in the same cluster. However, this is not enough: there is no guarantee that the hard constraints are satisfied. We resort to the discretization step.

A constrained K-means algorithm is presented in [13] to integrate hard constraints into K-means clustering. We want to take advantage of that: we propose to use constrained K-means in the discretization step to enforce hard constraints. Our work was inspired by [8], where K-means was used in the discretization step. But in [8], a repulsion matrix was not used, the use of K-means with a repulsion matrix was not justified, regular K-means instead of constrained K-means was used, and therefore no constraints were imposed.

In the following, we will first justify the use of K-means (with or without a repulsion matrix), and therefore the use of constrained K-means. We take each row of X_{conti}^* as a point, and perform K-means clustering². If the i^{th} row of X_{conti}^* belongs to the k^{th} cluster, then assign the original point s_i to the k^{th} cluster. We argue that this K-means clustering can achieve as good results as the best partition matrix $X_{discrete}^*$ minimizing $\|X_{discrete} - X_{conti}^*O\|^2$.

Proposition 1. For any orthonormal matrix O , row vectors of X_{conti}^*O and X_{conti}^* have the same K-means clustering results under the following condition: if c_l is the l^{th} initial center for X_{conti}^* , then c_lO is the l^{th} initial center for X_{conti}^*O .

Proposition 2. Suppose $X_{discrete}^*$ and O^* are the discrete partition matrix and rotation matrix minimizing $\|X_{discrete} - X_{conti}^*O\|^2$. If rows of $K \times K$ identity matrix I_K are taken as cluster centers, then one iteration of K-means clustering on row vectors of $X_{conti}^*O^*$ achieves the same clustering results as what represented by partition matrix $X_{discrete}^*$. Further, if $\|X_{discrete}^* - X_{conti}^*O^*\|^2$ is small, then the cluster centers will not go far away from rows of I_K , and therefore the K-means clustering on rows of $X_{conti}^*O^*$ will converge to the same clustering results as $X_{discrete}^*$.

From propositions 1 and 2, if $\|X_{discrete}^* - X_{conti}^*O^*\|^2$ is small, and rows of $(O^*)^{-1}$ are taken as initial cluster centers, then K-means clustering on X_{conti}^* achieves the same results as $X_{discrete}^*$. Small $\|X_{discrete}^* - X_{conti}^*O^*\|^2$ means that the points actually form good clusters, otherwise no clustering algorithm can work well. A good approximation of $(O^*)^{-1}$ can be found by finding orthogonal vectors among rows of X_{conti}^* .

K-means clustering on rows of X_{conti}^* with proper initializations (or through multiple initializations) can achieve as good results³ as minimizing $\|X_{discrete} - X_{conti}^*O\|^2$. On the other hand, hard constraints can be enforced by constrained K-means. So to incorporate hard constraints, K-means is a better discretization method.

Using constrained K-means in discretization step is to take row vectors of X_{conti}^* as points and run constrained K-means on them. In each iteration of the constrained K-means algorithm [13], when a point is assigned to a cluster, two criteria are used: (1) distance to the center of the cluster; and (2) whether the hard constraint is satisfied. A point is assigned to the closest cluster not violating hard constraints. Therefore, the constrained K-means guarantees that the hard constraints are satisfied.

² One might wonder what the difference is between performing K-means clustering on the original points and here at the discretization step. K-means clustering can work here because previous steps in spectral clustering have possibly transformed non-convex clusters into convex clusters (See more examples in [8]).

³ In [17], similar observation is presented through simulation, for the case of regular K-means and without a repulsion matrix.

Table 1. Summary of image data. Time span of each collection is shown in the second column. The third column gives the total number of days when the pictures were taken.

	time span	number of days	number of pictures with person	number of faces of faces labeled	number of persons (clusters)	number of faces for each person
family 1	Apr-Aug 2002	13	182	342	8	126,68,45,35,26,16,15,11
family 2	May-Nov 2003	14	149	224	16	42,16,16,16,16,14,13,12,11,11,11,11,10,10,9,9,8
family 3	May-Dec 2002	22	165	203	3	85,69,49

5 Experiments

Experiments are performed on real consumer photos. Collections from three families are used (Table 1). Face detection ([2]) is first run on these photos, and persons’ identities are manually labeled to provide ground truth for evaluation (only those individuals with 8 or more pictures are labeled). The data include a variety of scenes such as vacations in theme parks, a group of friends mountain climbing, having parties, fun activities at home, and children’s sports event.

5.1 Proposed Clothes Features vs. Color Histogram

The proposed clothes features (sections 2.2 and 2.3) are compared with color histograms (using χ^2 distance in CIElab space). To make the comparison fair, the same clothes detection and segmentation method (section 2.1) is used. Figure 3(a) shows the results by receiver operating characteristics (ROC) curves on five days’ images (from families 1 and 2), with around 100 pictures. Any pair of clothes pieces from the same person the same day are considered as a positive example, and any pair of clothes pieces from different people are considered as a negative example. These results show that the proposed method outperforms color histograms. More detailed studies reveal that the advantages of the new feature representation are more dominant when lighting condition changes.

5.2 Integrating clothes and hard constraints with face recognition

Clothes recognition results are to be combined with face recognition to provide pair-wise similarity measurements for clustering. Raw face scores are obtained from a face recognition module ([3, 5]). Logistic regression is used to combine face and clothes recognition results (section 3). The parameters of those logistic functions are learned using data from another family with around 200 faces and clothes pieces.

Figure 4 shows an illustrative example using images from a children’s party. Figure 4(b) is from face recognition only. Figure 4(c) gives results using additional contexts (clothes recognition and enforcing the constraint that different persons in one image must belong to different clusters). Five clusters are used,

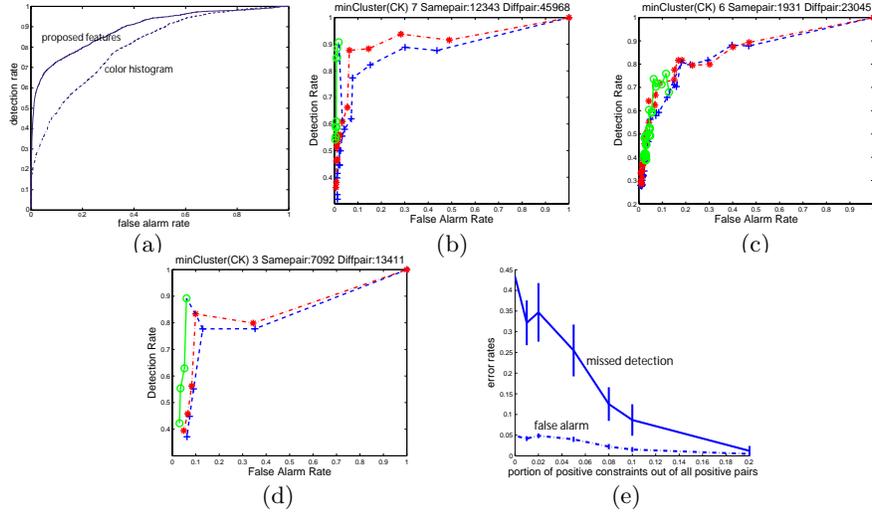


Fig. 3. (a): ROC curves: the proposed clothes features (EER: 20.1%) vs. color histograms (EER:28.3%). (b), (c), and (d): clustering results on family collections 1, 2, and 3 (Table 1), respectively. Blue dashed (with '+'): face recognition only; red dashdot (with '*'): clothes combined with faces, but without constraints; green solid curves (with 'o'): clothes and faces combined, and with constraints enforced. The most upper-right points of blue dashed (with '+') and red dashdot (with '*') curves correspond to the number of clusters being one, and from right to left with the increase of number of clusters. The minimum number of clusters for all the samples to satisfy hard negative constraints is displayed on the title 'minCluster(CK)'. The first point (from top right) of each green solid curve ('o') gives the results for that minimum number of clusters. The dashed curve in each graph connects results under that minimum number of clusters. 'Samepair' and 'Diffpair' on the title mean the total number of positive and negative pairs, respectively. (e): results of adding positive constraints. The vertical bars on the curves give standard deviation (from 30 runs for each fixed proportion).

which is the minimum number of clusters in order to satisfy the hard constraint. Figure 4 illustrates the benefits of using contexts. For instance, in the top row of Figure 4(b), there are faces from persons 'M' and 'R', and two faces from one image are in the same cluster ('R I4' and 'M I4'). This is corrected by using contexts as shown in Figure 4(c).

For **images collected on multiple days**, the affinity matrix is constructed as follows. For any pair of person images, if they are from pictures taken on the same day, both face and clothes information are used; otherwise, only face information is used. Clothes information is treated as missing if clothes are occluded or different people wear similar clothes. To enforce the negative hard constraint that two persons in one picture must be different individuals, repulsion matrix and constraint K-means are applied.

We use Rand index ([13]) to characterize clustering performance. Suppose we have N pieces of person images, any clustering results can be viewed as a

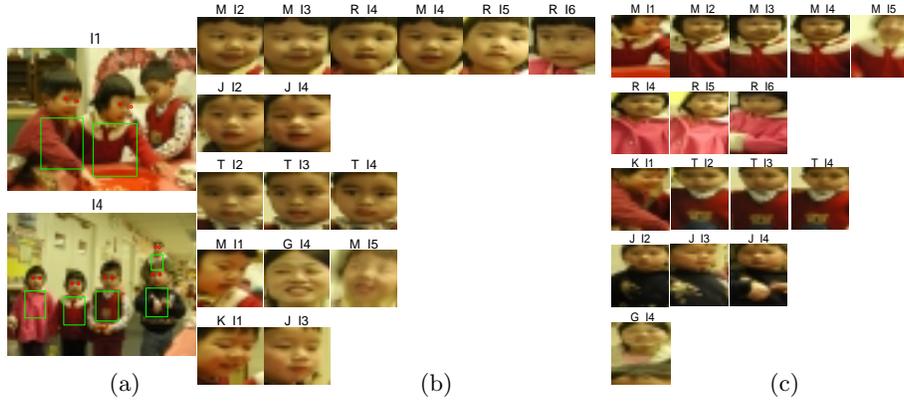


Fig. 4. An illustrative example. **(a)**: two sample images ('I1' and 'I4') with face detection (in small red circles) and clothes detection (in green lines). **(b)**: clustering results from faces only. Each row denotes one cluster. The first letter on top of each face gives the ground truth identity of the face, and the last two letters show which image it comes from. **(c)**: results from faces plus contexts (clothes recognition and the hard constraint that two faces in one image belonging to different clusters).

collection of $N * (N - 1)/2$ pairwise decisions. A false alarm happens when a pair actually from different individuals, but the algorithm claims they are the same individual. A true positive (detection) is when a pair actually from the same individual and the algorithm also claims so.

Clustering performance varies with the number of clusters. We experiment with different number of clusters: from one cluster to two times of the ground truth number of clusters (see Table 1). In applications, the desired number of clusters may be input by the user. Figure 3(b), (c), and (d) show the results on family collections 1, 2, and 3, respectively. From these curves, we can see that (1) clustering performance generally improves with the use of clothes; (2) the compatibility of logistic functions in section 3 is verified to a certain degree since similarities from face and clothes and similarities from face only are used in one affinity matrix, which outperforms the affinity matrix from face only; (3) hard constraints can help improve the results. Note that the performance improvements due to hard constraints are more dominant in Figure 3(b) and (d) than in (c). One possible reason is that the set of labeled faces from family 2 belong to 16 individuals. So for any random pair, the probability of belonging to different individuals is high, and hard negative constraints provide less information.

Positive constraints (meaning that a pair of person images must belong to the same individual) can also be applied. In practice, positive constraints are available through user feedback. Here we randomly choose a certain number of positive pairs to simulate the situation. Figure 3(e) shows experimental results on images from family 2. The ground truth number of clusters, 16, is used. Figure 3(e) indicates that positive constraints can improve clustering performance, especially for the detection rates.

6 Conclusions and Future Work

In this paper, we have developed a clothes recognition method which can work well for different types of clothes (smooth or highly textured), and under imaging condition changes. A principled way is provided to integrate clothes recognition results with face recognition results, and the cases when face or clothes information is missing are handled naturally. A constrained spectral clustering algorithm, which can utilize face, clothes and other context information (e.g. persons from one picture should be in different clusters), has been presented. Hard constraints are enforced in the spectral clustering algorithm so that logic-based context cues and user feedbacks can be used effectively. Picture-taken-time is used when face and clothes recognition results are combined. Experiments on real consumer photos show significant performance improvements. Future work includes exploring how to select the number of clusters automatically, although in human clustering applications, it can possibly be input by users.

References

1. R. Fergus, P. Perona, and A. Zisserman, "Object class recognition by unsupervised scale-invariant learning", In *CVPR*, 2003.
2. S. Ioffe, "Red eye detection with machine learning", In *Proc. ICIP*, 2003.
3. S. Ioffe, "Probabilistic linear discriminant analysis", In *Proc. ECCV*, 2006.
4. M.I. Jordan and R.A. Jacobs, "Hierarchical mixtures of experts and the em algorithm", *Neural Computation*, 6:181–214, 1994.
5. T. Leung, "Texton correlation for recognition", In *ECCV*, 2004.
6. D. Lowe, "Object recognition from local scale-invariant features", In *ICCV*, 1999.
7. K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors", In *CVPR*, 2003.
8. A.Y. Ng, M.I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm", In *NIPS 14*, 2002.
9. H. Schneiderman and T. Kanade, "A statistical method for 3d object detection applied to faces and cars", In *Proc. CVPR*, 2000.
10. J. Shi and J. Malik, "Normalized cuts and image segmentation", In *Proc. CVPR*, pages 731–7, June 1997.
11. J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos", In *Proc. ICCV*, 2003.
12. P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features", In *Proc. CVPR*, 2001.
13. K. Wagstaff, C. Cardie, S. Rogers, and S. Schroedl, "Constrained k-means clustering with background knowledge", In *Proc. ICML*, 2001.
14. Y. Weiss, "Segmentation using eigenvectors", In *Proc. ICCV*, 1999.
15. Stella X. Yu, *Computational Models of Perceptual Organization*, Ph.d. thesis, Carnegie Mellon University, 2003.
16. S.X. Yu and J. Shi, "Grouping with bias", In *NIPS*, 2001.
17. S.X. Yu and J. Shi, "Multiclass spectral clustering", In *Proc. ICCV*, 2003.
18. L. Zhang, L. Chen, M. Li, and H. Zhang, "Automated annotation of human faces in family albums", In *MM'03*, 2003.