FINDING ROWS OF PEOPLE IN GROUP IMAGES

Andrew C. Gallagher

Carnegie Mellon University
Department of ECE
agallagh@cmu.edu

Tsuhan Chen

Cornell University
School of ECE
tsuhan@ece.cornell.edu

ABSTRACT

People are among the most popular subjects in photography, and in many social settings, images of groups of people are captured. People often arrange themselves in a very structured manner in these group images. For example, taller people might stand in a row behind smaller people. This structure is often exploited in captions that sequentially label the individuals in each row.

We present an algorithm for automatically finding rows of people in group images. A graph is defined for the image, where each face is a vertex. Energy terms are learned from a training set of images. A minimum cut on this graph defines the rows of people in the image. On our test set, the algorithm achieves perfect results with 67.5% of the images. Detecting rows of people is useful for a number of applications.

Index Terms— group shots, people, object recognition, graph cuts

1. INTRODUCTION

It is common in social gatherings to capture an image of a group of people. In these situations, people are often arranged in rows to ensure that the camera can view each face. Depending on the situation, the rows can be either highly structured, or more informal as shown in Fig. 1. The definition of what constitutes a row of people is not obvious. Our definition of a row of people is as follows: within a row of people, each person is at roughly constant distance from the camera, roughly in the same physical posture (e.g. sitting, standing, or kneeling), and roughly supported by the same surface (e.g. all people in a row stand on the same step in a flight of stairs). In this paper, we present an algorithm for detecting rows of people using graph cuts with learned energy terms.

The algorithm itself relies on the fact that there is order in the manner in which people arrange themselves in social situations. In the social sciences, the study of personal space dates to the mid-twentieth century [1]. Even without conscious effort, the relative positions of people in social situations is affected by, among other factors, age, gender, social status, the local culture, and even lighting. Our broader goal is to use the discoveries from the social sciences as *social context* for interpreting images of people. Social context is context that describes people, their culture, and the social aspects of their interactions at the time and place the image was captured.

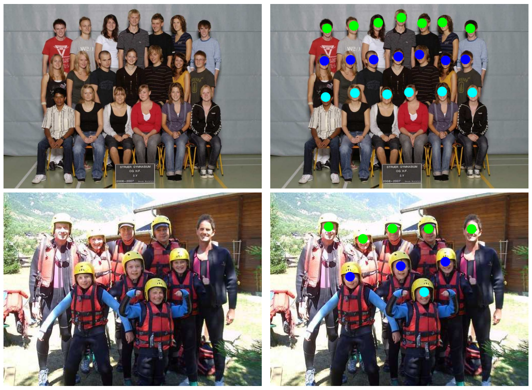


Fig. 1. In many group shots, people are arranged in rows that have physical meaning in the scene. Sometimes these rows are highly structured (top) and other times less so (bottom). Our algorithm discovers rows of faces in the images. In the images on the right, each row's faces are marked with a dot of the same color.

Recovering the rows of people in a group image has applications in searching, organizing, and annotating images.

2. RELATED WORK

There is, of course, a large body of work on facial features and recognition, e.g. [2, 3]. However, the majority of this work considers each face as an independent problem. Exceptions include efforts to characterize the frequency of an individual appearing in a personal image collection and modeling the likelihood that specific combinations of people will appear together in an image or event [4, 5, 6]. None of this work considers the position of a face within the image.

Despite the prevalence of group shots, there is surprisingly little work devoted to their analysis and understanding. In [7], the authors attempt to match people in repeated shots

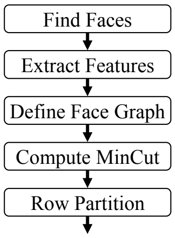


Fig. 2. The flow diagram of our algorithm for finding rows of faces in an image of a group of people.

of the same scene. Clothing, face, and hair are considered to establish correspondences. Facial arrangement was considered in [8] as a way to measure similarity between pairs of images, but was not explored as a means for understanding a single image. In [9], a rule-based system is proposed for tagging faces based on directional cues in annotations.

Regarding the method we use to solve the problem, graph cuts are used to solve many problems in computer vision [10, 11]. In pairwise models, an energy function is composed of unary and pairwise energy terms in a manner that a graph cut provides the optimal solution. However, the unary and pairwise terms are usually defined by hand (e.g., based on pixel intensity difference as in [12, 13]) to achieve good results. Recently, there have been efforts to learn distance metrics [14] from labeled training samples to better express the similarity between samples. Our problem is essentially to produce a clustering of the faces into k rows, where k is unknown. In this paper, we take the latter approach by training a classifier to distinguish between pairs of faces that are in the same row, and pairs that are in different rows. This classifier is used to establish the energy terms in our graph model.

Our contributions are the following: We present an algorithm for detecting rows of people in group images. Our approach uses graph cuts on a graph whose edge weights are learned with a classifier from training data. Our model represents the social context of personal space for solving a practical image understanding problem.

3. FEATURES AND THE FACE GRAPH

Fig. 2 shows the algorithm flow. First, faces are detected with a face detector and an Active Shape Model [15] is used to locate the left and right eye positions that serve as features. Next, an undirected graph is defined where each human face is represented as a vertex. Edge weights are learned as a function of the features of the pair of faces connected by each edge. The graph is constructed so that under certain assumptions, a minimum cut produces the most likely binary split separating the group of faces into sets of rows of faces. By recursively applying this binary split until a stopping criteria is met, the row partition of the image is found.

3.1. Face Features

The position $\mathbf{p} = \begin{bmatrix} x_i & y_i \end{bmatrix}^T$ of a face f is the two dimensional centroid of the left and right eye center positions $\mathbf{l} = \begin{bmatrix} x_l & y_l \end{bmatrix}^T$ and $\mathbf{r} = \begin{bmatrix} x_r & y_r \end{bmatrix}^T$. The distance between the two eye center positions for the face is the size $e = \langle \mathbf{l} - \mathbf{r} \rangle_2$ of the face. To capture the structure of the people image, and allow the structure of the group to represent context for each face, we compute the following features and represent each face $\mathbf{f}_n$ as a 3-dimensional contextual feature vector $\mathbf{f}_n = \begin{bmatrix} x_n & y_n & e_n \end{bmatrix}^T$.

3.2. The Face Graph

Next, the face graph $G = (V, E)$ is constructed where each face n is represented by a vertex $v_n \in V$, and each edge $(v_n, v_m) \in E$ connects vertices v_n and v_m. This graph defines a Conditional Random Field (CRF) that represents $P(\mathbf{v}|\mathbf{F})$, the probability of a labeling given the features associated with all faces in the image. We seek the most probable binary labeling $\mathbf{v}^*$ of the faces.

$$P(\mathbf{v}|\mathbf{F}) \propto \prod_n \Psi(v_n) \prod_{(v_m,v_n)\in E} \Phi(v_m, v_n) \tag{1}$$

The most probable labeling $\mathbf{v}^*$ is found as:

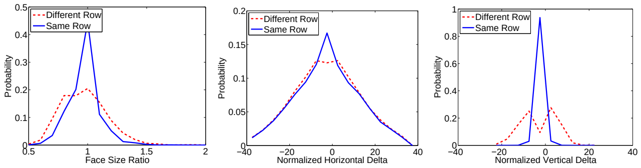
$$\mathbf{v}^* = \underset{\mathbf{v}}{\operatorname{argmax}}\, P(\mathbf{v}|\mathbf{F}) \tag{2}$$

$$= \underset{\mathbf{v}}{\operatorname{argmin}} - \sum_n \log \Psi(v_n) - \sum_{(v_m,v_n)\in E} \log \Phi(v_m, v_n) \tag{3}$$

where n and m are indices over particular faces in the image. Possible row labels for each face are $v_n \in \{0, 1\}$ where 0 and 1 represent different rows. The unary term $\Psi(v_n)$ is constant because nothing in our model distinguishes between the facial features and which row that face is likely to be in. The pairwise term $-\log \Phi(v_m, v_n)$ represents the cost of assigning either the same or different labels (row indices) to a pair of faces.

From (3), the most likely row labeling $\mathbf{v}^*$ corresponds with the minimum cut of the graph G, when the edge weights are $-\log \Phi(v_n, v_m)$. Learning these parameters $\Phi(v_n, v_m)$ in an undirected graphical model is notoriously difficult. Intuitively we would like to be rewarded for cutting between faces that are probably in different rows, and penalized for cutting between faces that are likely in the same row. Under the naïve Bayes assumption that each pair of faces is independent of all others, then $\Phi(v_n, v_m)$ is related to the probability $P(s_{mn}|\mathbf{f}_m, \mathbf{f}_n)$, where s_{mn} is the event that $v_m = v_n$ (faces m and n belong to the same row) as follows:

$$\Phi(v_m, v_n) = \begin{bmatrix} 1 & \frac{1-P(s_{mn})}{P(s_{mn})} \\ \frac{1-P(s_{mn})}{P(s_{mn})} & 1 \end{bmatrix} \tag{4}$$



(a) Relative Size (b) Horizontal Delta (c) Vertical Delta

Fig. 3. The distributions of the features $\mathbf{f}_{mn}$ associated with a pair of faces in an image. Faces in the same row tend to be of similar size (a), and be close horizontally (b), and vertically (c).

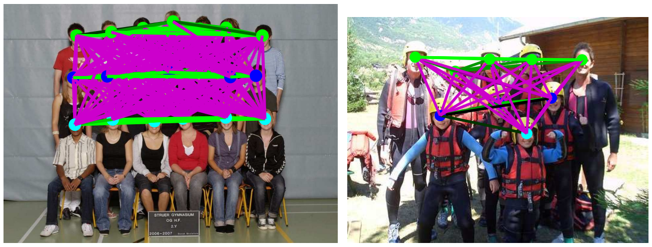


Fig. 4. A visualization of the graph G for the two images from Fig. 1. Every face is a node in the graph, and edges weights are the cost to cut the edge. Green edges indicate a cost to cut (the pair is likely in the same row) and magenta edges indicate a reward for cutting (the pair is likely in different rows). Black edges neither reward nor penalize a cut.

Using a set of training images where each face row has been identified, the term $P(s_{mn}|\mathbf{f}_m, \mathbf{f}_n)$ can be learned with any classifier. In our work, we learn $P(s_{mn}|\mathbf{f}_m, \mathbf{f}_n)$ with Gaussian Maximum Likelihood (GML), using a multivariate Gaussian to represent each of the two classes (either $v_n = v_m$ or $v_n \neq v_m$) from the training data, where an equal prior is assumed. The feature vector $\mathbf{f}_{mn}$ is produced from $\mathbf{f}_m$ and $\mathbf{f}_n$) and represents the relative position and scales of the two faces in the image as follows:

$$\mathbf{f}_{mn} = \begin{bmatrix} \frac{e_m}{e_n} & \frac{x_m - x_n}{e_m} & \frac{y_m - y_n}{e_m} \end{bmatrix}^T \tag{5}$$

Fig. 3 shows one-dimensional projections of the distributions of the values of $\mathbf{f}_{mn}$ for each of the two class values of s_{nm}.

3.3. Recursive Minimum Cuts

The graph $G = (V, E)$ is formed with vertices at each face in the image and edge weights assigned according to (3) and (4). A visualization of the graph G is shown in Fig. 4. The minimum cut of the graph is found, partitioning the graph vertices into two sets V_1 and V_2. Note that the value of the cut need never exceed 0, since when V_1 is the empty set no edges are cut. The subgraph associated with each set is then recursively cut until the value of the cut is zero. In this way, the original graph G is partitioned into k components, each representing a row of people. Unlike many unsupervised clustering algorithms, the number of components does not need to be supplied by a human because the whole process is guided by the labeled training data. After the process converges, the rows are renumbered starting from the image top.

It is important to note several approximations in our approach. First, the general problem of finding a minimum cut on a graph with negative weights is NP-hard, although in special cases efficient algorithms exist. We use a spectral relaxation [13] to find an approximate solution. Let $\mathbf{A}$ represent the adjacency matrix, where each element $a_{mn} = -\log \frac{1-P(s_{mn}|\mathbf{f}_{mn})}{P(s_{mn}|\mathbf{f}_{mn})}$ is the cost associated with cutting an edge. Then the eigenvector of the graph Laplacian $\mathbf{L} = \mathbf{D} - \mathbf{A}$ associated with the smallest eigenvalue is binarized to approximate the minimum cut solution. $\mathbf{D}$ is a diagonal matrix with each element equal to the corresponding row sum of $\mathbf{A}$.

Second, is must be noted that even if each recursive minimum cut is exact, there is no guarentee that the final partition will be equal to that achieved by performing an optimal exact k-way minimum cut of the graph. In image segmentation, this problem is addressed with an application of k-means after a dimensionality reduction [16]. We leave this topic as future work to explore.

Despite these approximations, the model provides useful solutions to the row segmentation problem, as shown in the next Section.

4. EXPERIMENTS

To test our ideas, we collected a images from Flickr using the search string:
"`group shot`" or "`group photo`" or "`group portrait`"

The rows of people were manually labeled in 234 images. In total, these images contain 2222 faces and 465 rows of people (approximately 2 rows per image and 4.8 people per row. The number of people in each image ranges from 4 to 28, and the number of rows per image ranges from 1 to 5.

We test on one image at a time, leaving the rest of the images for training the GML classifier. We use a complete graph G over the face vertices to find the rows of people. The row clustering quality is compared to the manually labeled rows using the Rand Index [17]. Each partition is viewed as a collection of $n * (n-1)/2$ decisions, one decision per pair of data points. In a given partition of the data, two data points are either in the same or in different cluster. The Rand Index quantifies the proportion of the decisions for which the algorithm's decision and the ground truth decision match. A perfect score in this metric is 1.0, or 100%.

Table 1 reports our results. The algorithm's Rand Index is 92.6%. The algorithm achieved perfect row segmentation on

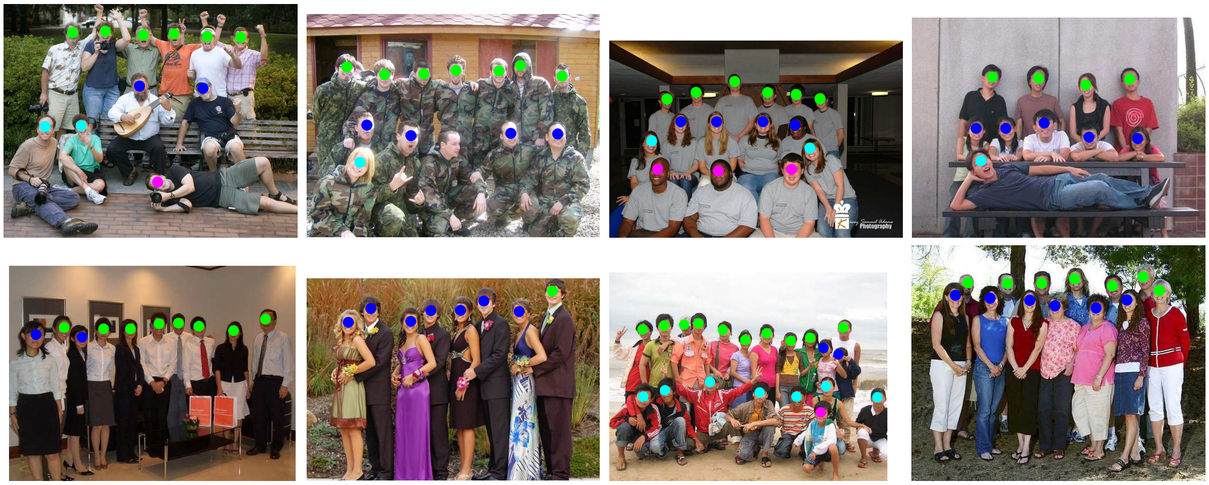


Fig. 5. **Top:** Examples where our algorithm perfectly recovered the rows of people. Notice the variety of postures and arrangements of the people, from standing, sitting, and even laying. **Bottom:** Imperfect results. Sometimes people smaller (left) or taller (second image) than the rest of the row cause mistakes. Each of these images actually has only a single row of people. In the third and fourth images, mistakes occur near the right side where the algorithm is confused by a junction of multiple rows. Best viewed in color.

	Accuracy
Rand Index	92.6%
Correct Images	67.5%
Correct No. Rows	73.5%

Table 1. Quantitative results from applying our algorithm to 234 images containing 2222 people.

greater than two-thirds of the images (67.5%). The discovered number of rows k is correct 73.5% of the time. Fig. 5 provides discussion of the algorithm results on eight image examples.

5. CONCLUSIONS

In this paper we introduce a graph-based algorithm for finding rows of people in group images. In our approach, a graph is constructed whose minimum cut corresponds to a separation between rows of people. Our approach is shown to be effective by testing on a large number of images of people. We demonstrate that image understanding benefits by considering the social context provided by the structure of multiple people in an image. We feel this is a rich area for researchers, and we provide our image collection to other interested researchers [18].

6. REFERENCES

[1] E. Hall, “A system for the notation of proxemic behavior,” in *American Anthropologist*, 1963.

[2] W. Zhao, R. Chellappa, P. Phillips, and A. Rosenfeld, “Face recognition: A literature survey,” *ACM Comput. Surv.*, 2003.

[3] P.J. Phillips, W. Scruggs, A. O’Toole, P. Flynn, K. Bowyer, K. Schott, and M. Sharpe, “FRVT 2006 and ICE 2006 large-scale results,” 2007.

[4] A. Gallagher and T. Chen, “Using group prior to identify people in consumer images,” in *Proc. CVPR SLAM workshop*, 2007.

[5] M. Naaman, R. Yeh, H. Garcia-Molina, and A. Paepcke, “Leveraging context to resolve identity in photo albums,” in *Proc. JCDL*, 2005.

[6] Z. Stone, T. Zickler, and T. Darrell, “Autotagging facebook: Social network context improves photo annotation,” in *Proc. CVPR, Internet Vision Workshop*, 2008.

[7] J. Sivic, C.L. Zitnick, and R. Szeliski, “Finding people in repeated shots of the same scene,” in *Proc. BMVC*, 2006.

[8] M. Abdel-Mottaleb and L. Chen, “Content-based photo album management using faces’ arrangement,” in *Proc. ICME*, 2004.

[9] R. Chopra and R. Srihari, “Control structures for incorporating picture-specific context,” in *in Image Interpretation, Proc. IJCAI ’95*, 1995.

[10] Y. Boykov, O. Veksler, and R. Zabih, “Efficient approximate energy minimization via graph cuts,” *PAMI*, 2001.

[11] C. Rother, V. Kolomogorov, and A. Blake, “Grabcut- interactive foreground extraction using iterated graph cuts,” in *Proc. ACM Siggraph*, 2004.

[12] P. Felzenszwalb and D. Huttenlocher, “Efficient graph-based image segmentation,” *Int. J. Comput. Vision*, vol. 59, no. 2, 2004.

[13] J. Shi and J. Malik, “Normalized cuts and image segmentation,” *PAMI*, 2000.

[14] L. Yang and R. Jin, “Distance metric learning: A comprehensive survey,” http://www.cse.msu.edu/˜yangliu1/frame_survey_v2.pdf, 2006.

[15] T. Cootes, C. Taylor, D. Cooper, and J. Graham, “Active shape models-their training and application,” *CVIU*, 1995.

[16] A. Ng, Y. Weiss, and M. Jordan, “On spectral clustering: Analysis and an algorithm,” in *Proc. NIPS*, 2002.

[17] *Objective Criterial for the Evaluation of Clustering Methods*, 1971.

[18] A. Gallagher and T. Chen, “The images of groups dataset,” http://amp.ece.cmu.edu/people/andy/imagesOfGroups.html, 2009.