A Compositional Exemplar-Based Model for Hair Segmentation

Nan Wang[1], Haizhou Ai[1], and Shihong Lao[2]

[1] Computer Science & Technology Department, Tsinghua University, Beijing, China
ahz@mail.tsinghua.edu.cn
[2] Core Technology Center, Omron Corporation, Kyoto, Japan
lao@ari.ncl.omron.co.jp

Abstract. Hair is a very important part of human appearance. Robust and accurate hair segmentation is difficult because of challenging variation of hair color and shape. In this paper, we propose a novel Compositional Exemplar-based Model (CEM) for hair style segmentation. CEM generates an adaptive hair style (a probabilistic mask) for the input image automatically in the manner of Divide-and-Conquer, which can be divided into decomposition stage and composition stage naturally. For the decomposition stage, we learn a strong ranker based on a group of weak similarity functions emphasizing the *Semantic Layout similarity* (SLS) effectively; in the composition stage, we introduce the *Neighbor Label Consistency* (NLC) Constraint to reduce the ambiguity between data representation and semantic meaning and then recompose the hair style using alpha-expansion algorithm. Final segmentation result is obtained by Dual-Level Conditional Random Fields. Experiment results on face images from Labeled Faces in the Wild data set show its effectiveness.

1 Introduction

In computer graphics, hair acquisition [1] [2] and hair geometry modeling [3] have achieved significant progresses. While in computer vision, hair style analysis or hair segmentation discussed in this paper is still an ongoing research issue. Hair is a very important part of human appearance especially in consumer images. In visual surveillance condition or criminal cases, face details usually cannot be seen or remembered or described clearly. However, hair style is easier to be identified and described in most cases, so it usually becomes one of the most important descriptors for some specific target person. For this application, hair segmentation becomes a necessary intermediate step to hair style identification. Moreover, with the rapid development of internet, online makeup has become more and more popular. When people want to see whether or not some hair style fits them, a good hair style identification or search tool could help a lot, which also makes hair segmentation necessary. Nevertheless, there are challenges for segmenting hair area in consumer images because of the variation of shape and color. Robust hair segmentation is by far an unsolved problem.

R. Kimmel, R. Klette, and A. Sugimoto (Eds.): ACCV 2010, Part III, LNCS 6494, pp. 171–184, 2011.

Fig. 1. It is easy to tell bald from the long hair. But it is extremely hard to tell the long hair from longer ones.

Yacoob and Davis [4] build a hair color model and then adopt a region growing algorithm to modify the hair region. However, this method will only work when the hair color doesn't change significantly, especially for the dark hair. Consumer images do not fit in this constraint.

Lee et al [5] give a more practical algorithm for consumer images. They first cluster hair style and the color of hair and face into several typical patterns manually. And then for each hair style, choose the fittest hair and face color model and modify it according to the input image. A Markov Random Field is built and inferred to maximize the joint probability distribution of each pixel on each label. The one whose labeling result has the minimized distance to its corresponding hair style is chosen as the final hair style. Their work gives a practical idea to solve the problem; nevertheless there are still several issues we need to focus on. Hair style classification is a hard issue. It is difficult to decide how many patterns are appropriate even for just front view, let alone cases with side view. With a predefined cluster label, it is still hard to decide which hair style an input image belongs to. It may be easy to tell bald from long hair, but extremely hard to tell long hair from longer one, as shown in Figure 1. Unfortunately, this classification is vital because unary term plays dominant role in graph model [6].

In Borenstein and Ullman [7], a combined top-down and bottom-up algorithm is proposed to solve the problem of figure-ground segmentation. During top-down procedure, image fragments and the corresponding figure background labels are extracted from training data first and then used to optimally cover an object in a novel image to induce the final segmentation result. Wang and Tang [8] approached the problem of face photo-sketch synthesis and recognition. The input image is normalized and divided into overlapped rectangles. For each rectangle, K candidate patches from the training set are selected. A multi-scale Markov Random Field model is used for the selection of optimal combination of patches. Jolic et al. [9] model the spatial correlations in image class structure by introducing the Stel to make image models invariant to changes in local measurement, while sensitive to changes in image structure.

Inspired by these works, we build a Compositional Exemplar-based Model (CEM, section 2) for hair style generation, which could generate an appropriate hair style for the input image. In our paper, actually four labels are used:

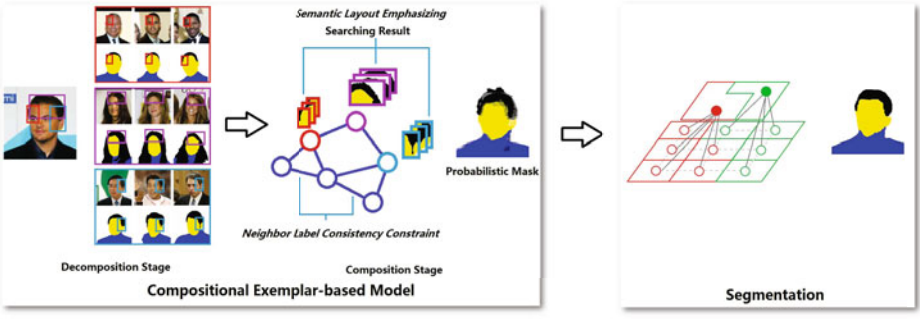

Fig. 2. Work flow of Compositional Exemplar-based Model. Color code for labels are: white - background, yellow - face, black - hair and blue - clothes. (This figure is best viewed in color.)

background, face, hair and clothes. CEM works in the Divide-and-Conquer way as illustrated in Figure 2.

In the decomposition stage, we design a group of *Semantic Layout Similarity* (SLS) features (section 2.1), which are combined together to get a strong and effective similarity function for each location respectively. Based on the similarity function, candidate segmentation results are collected for each local patch from a manually labeled library in this stage.

In the composition stage, we introduce a *Neighbor Label Consistency* (NLC) Constraint and organize local patches as a Markov Network (section 2.2). A well-defined consistency function promises the *regularity* [10], which allows us to optimize the CEM using α-expansion algorithm [10] [11] [12]. CEM finally generate a probabilistic mask as illustrated in Figure 2. With the favor of the mask, we obtain the final segmentation result using a dual-level Conditional Random Fields (section 3).

2 Compositional Exemplar-Based Model

It is hard to model the hair styles integrally, since hair styles have large variation as shown in Figure 1. The basic idea is to decompose a hair style into local patches and model each patch respectively. The reason is that although hair styles can differ from each other dramatically in global, they can still share some common *Semantic Layout* in local. In our paper, *Semantic Layout* means the actual label patterns of patches. There is intuitional evidence in the diagram of the Decomposition Stage in Figure 2. The purple patch of the input image covers forehead and hair root regions. The first three searching hair style are very different from the query one, but just in this local patch, they seem the same. This is why we can model hair style locally.

In the decomposition stage, candidate segmentation results are obtained independently. However, the independence of searching will lead to ambiguity sometimes, because we use the similarity defined in data representation level

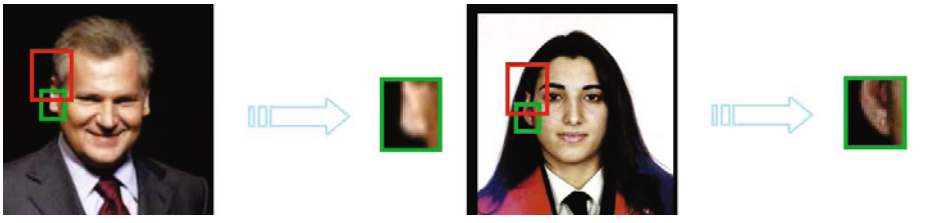

Fig. 3. Ambiguity of patches. Although the two green patches seem very similar to each other in date representation level, the dark color parts of the patches have totally different semantic meanings. When neighbor patch (in red) are considered together, the ambiguity can be avoided.

to approximate the actual one in semantic level. We give an example in Figure 3, where the local patches from two images are almost the same in the data representation level but have totally different meanings for the dark color part in the semantic level. However, if its neighbor patch (in the red rectangle) is considered together, this ambiguity can be avoided most of the time. From this point, we introduce the *Neighbor Label Consistency* (NLC) Constraint to reduce the ambiguity.

There are two key problems in the model. The first one is how to define a similarity to capture the *Semantic Layout* information. And the second one is how to select the best candidates for all patches together when NLC Constraint is introduced. They will be described in the next subsections respectively. Before that, we define some notations.

P_i is the local patch of the image and its corresponding label result is denoted as L_i. The local patches are required overlapping with its adjacent ones. Then the neighbor patch indices of P_i is denoted as $N(i)$. For each patch P_i, there is an exemplar library for it, which is denoted as $\left\{\mathcal{P}_i^k\right\}$. The manually labeled result for the exemplar library is $\left\{\mathcal{L}_i^k\right\}$. The *similarity function* in data representation level between patch P and Q is defined as $H(P,Q)$. The similarity function $C(P,Q)$ for *Semantic Layout* between patch P and Q is defined as follows:

$$C(P,Q) = \frac{1}{\Delta}\sum_m \delta\left(L_p^m = L_Q^m\right) \tag{1}$$

where Δ is the size of region P and Q. $\delta(\cdot)$ is Kronecker delta function.

2.1 Learning Similarity Function by SLS Features

In this subsection, we focus on how to define a similarity function to capture the *Semantic Layout* information. Similarity can be defined on the statistic information, such as histograms, or on the data structure, such as Euclid distance, or on a fusion of them. One thing should be noticed in the problem is that the feature compactness of different labels are not the same. For example, face and hair have some typical pattern of color or texture distributions; while clothes feature distribution is looser and background feature distributions barely share

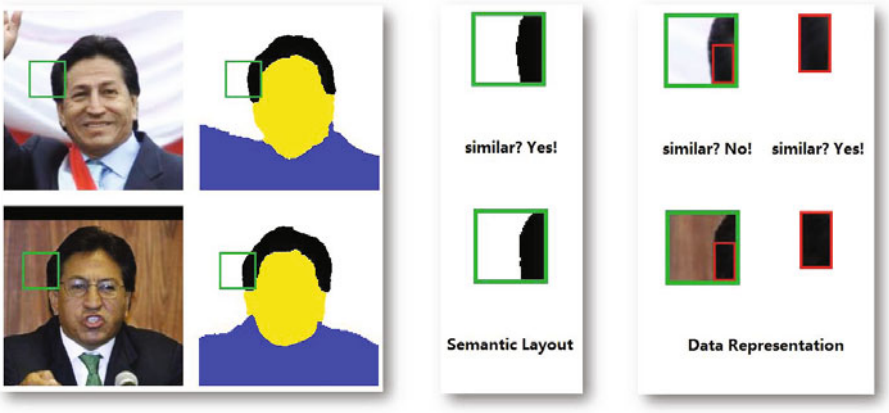

Fig. 4. Green patches of the two images have very similar *Semantic Layout*. However, if the similarity is defined based on the whole feature of the patch, background difference will dominate the similarity between them and cause a loss of the good exemplar candidate. Our SLS feature is calculated in selected sub-patches, such as the red rectangle. In this way, better consistency between data representation and semantic meanings can be achieved.

anything from one image to another. This characteristic will cause loss of good exemplar labels sometimes, as shown in Figure 4. To achieve a better consistency between data representation and semantic meanings, our similarity function is constructed based on the features in local sub-patches. A similar work to capture the *Semantic Layout* information is that of Shotton et al. [6] which presents a discriminative model to fuse shape, appearance and context information to recognize efficiently the object classes. Our algorithm is different from that since we focus on the explicit similarity of *Semantic Layout* of *patches*, while they focus on the classification of *pixel* using *Semantic Layout* as a learning cue.

Formally, denote the SLS feature set $\Phi = \{\phi_0, \phi_1, \cdots, \phi_M\}$. Our algorithm use color and texture as basic features, such as RGB, HSL color space, Gabor wavelet, which are represented as histograms (Gabor wavelet is transformed as LGBP [13](Local Gabor Binary Pattern)). Each SLS feature in Φ is determined by a triple $\langle F_m, R_m, B_m \rangle$. F_m denotes the feature type, which can be histogram of R channel in the RGB color space or LGBP in some specific frequency and orientation. R_m is the rectangle where F_m histogram is calculated. B_m is the bin index of the histogram. Let $\phi_m(\mathcal{P}_i) = \phi_{\{F_m, R_m, B_m\}}(\mathcal{P}_i)$ be the B_m bin value of F_m histogram extracted from $\mathcal{P}_i$ in the rectangle R_m. Then the weak ranker $h_m(\mathcal{P}_i, \mathcal{P}_j)$ is calculated as:

$$h_m(\mathcal{P}_i, \mathcal{P}_j) = -\frac{(\phi_m(\mathcal{P}_i) - \phi_m(\mathcal{P}_j))^2}{\phi_m(\mathcal{P}_i) + \phi_m(\mathcal{P}_j)} \tag{2}$$

which actually is the opposite number of one addend term from the Chi-Square Distance equation. So the final *similarity function* is a generalization of Chi-Square Distance.

Table 1. Preference Pairs Generation Algorithm

Input: Exemplar library in specific location $\left\{\mathcal{P}_i^k\right\}$, threshold τ
Output: Preference Pair Set (Training Set) $\{T_j\}$

- Initialization: $\{T_j\} \leftarrow \Phi$
- For each $\mathcal{P}_i^j$
  - Sort the other patches based on $C\left(\mathcal{P}_i^j, \mathcal{P}_i^k\right)$, and get a permutation of the other patches $\pi(m)$, which maps the patch's sorting index m to its original index $\pi(m)$ in $\left\{\mathcal{P}_i^k\right\}$.
  - For each $\mathcal{P}_i^{\pi(m)}$, which satisfies that $C\left(\mathcal{P}_i^j, \mathcal{P}_i^{\pi(m)}\right) < \tau$
    - Add $\left(\mathcal{P}_i^j, \mathcal{P}_i^{\pi(0)}, \mathcal{P}_i^{\pi(m)}\right)$ in $\{T_j\}$
  - End For
- End For

To get a good enough similarity function, we apply the RankBoost [14] learning algorithm to select the best SLS features and evaluate their weights. For the RankBoost algorithm, preference pairs should be defined to serve as the training data. In our problem, the preference is defined by the manually labeled result similarity of the two exemplar patches $C(\mathcal{P}_i, \mathcal{P}_j)$. For each exemplar library $\left\{\mathcal{P}_i^k\right\}$, one of them is used as the query patch, and the others are sorted based on $C(\mathcal{P}_i, \mathcal{P}_j)$. And we *prefer* that the similarity between the query one and the first one is larger than that between the query one and the end ones. Specifically, the preference pairs (training set) generation algorithm is shown in Table 1.

The training objective of our algorithm is to construct a strong ranker function (*similarity function* in our paper) so that:

$$\forall \left(\mathcal{P}_i^j, \mathcal{P}_i^{\pi(0)}, \mathcal{P}_i^{\pi(m)}\right) \in \{T_j\}, H\left(\mathcal{P}_i^j, \mathcal{P}_i^{\pi(0)}\right) > H\left(\mathcal{P}_i^j, \mathcal{P}_i^{\pi(m)}\right) \tag{3}$$

The *similarity function* $H(\cdot)$ is the weighted sum of weak rankers, the same as other boosting algorithm. The details of RankBoost training algorithm can be found in [14].

2.2 Introduce NLC Constraint into CEM

In CEM, NLC Constraint is achieved by enforcing pixel to be assigned the same label no matter which patch it locates in. So the *consistency function* can be defined by $C_A(P, Q)$, which is $C(P, Q)$ restricted on the overlapping area A of P and Q. The CEM can be represented formally as a Markov Network. The Node is the patch set $\{P_i\}$, and the neighborhood system is just defined before. Suppose C best candidate exemplars are reserved. The optimization of CEM can be done by minimizing the following energy function:

$$E(P) = \sum_i \left(\varphi_i(c_i) + \sum_{j \in N(i)} \varphi_{i,j}(c_i, c_j) \right) \tag{4}$$

where c_i denotes the index of exemplar that P_i finally take. The unary function $\varphi_i(c_i)$ is defined as:

$$\varphi_i(c_i) = -\log\left(H\left(P_i, \mathcal{P}_i^{c_i}\right)\right) \tag{5}$$

And the pairwise function $\varphi_{i,j}(c_i, c_j)$ is defined as:

$$\varphi_{i,j}(c_i, c_j) = -\log\left(C_A\left(\mathcal{P}_i^{c_i}, \mathcal{P}_j^{c_j}\right)\right) \tag{6}$$

However, the straightforward definition is not *regular* [10]. According to the theorem of [10], the *regularity* of pair wise term is a necessary and sufficient condition for graph-representability. So this energy function cannot be minimized by graph-cut based algorithm. The problem can be solved by expanding the node label set from $\mathcal{L} = \{0, 1, \cdots, C-1\}$ to $\mathcal{L} = \{0, 1, \cdots, nC-1\}$, where n is the vertices number. All possible candidate exemplars of all patches are grouped together. Since each patch i can only take label ranged between iC and $(i+1)C-1$ actually, the other assignment should be set as a maximum value. The mapping function between label index is $f(c_i) = c_i - iL$. The unary term is computed as:

$$\tilde{\varphi}_i(c_i) = \begin{cases} \varphi_i(f(c_i)) & iC \le c_i < (i+1)C \\ max & others \end{cases} \tag{7}$$

To satisfy the regular condition in [10], the pair wise term is modified as follows:

$$\tilde{\varphi}_{i,j}(c_i, c_j) = \begin{cases} \beta\varphi_{i,j}(f(c_i), f(c_j)) & iC \le c_i < (i+1)C, jC \le c_j < (j+1)C \\ 0 & c_i = c_j \\ max & others \end{cases} \tag{8}$$

The proof of the *regularity* of $\tilde{\varphi}_{i,j}(c_i, c_j)$ is given in supplementary files. With the constraint of unary term, c_i and c_j will always satisfy the first condition in pair wise term, when the assignment is optimal. So the labeling result of graph model with the expanding label set is equivalent to the former one. Although the expanding label set will increase computation load, in practice the inference is still fast enough, because the number of super pixels is very small in general. Denote the optimal solution of CEM as $\{L_i\}$. The probabilistic hair style mask is constructed to retain all the information of overlapping patches. The mask M is calculated as:

$$M_{i,l} = \frac{\sum_{i,j \in P_i} \delta\left(L_i^{\tilde{j}} = l\right) + \epsilon}{\sum_{i,j \in P_i} 1 + \epsilon} \tag{9}$$

where $M_{i,l}$ denotes the probability of assigning pixel i with label l. $\mathcal{L}_i^{\tilde{j}}$ is the manually labeled result of optimal exemplar in index $\tilde{j}$ which is the corresponding index of pixel j in patch P_i.

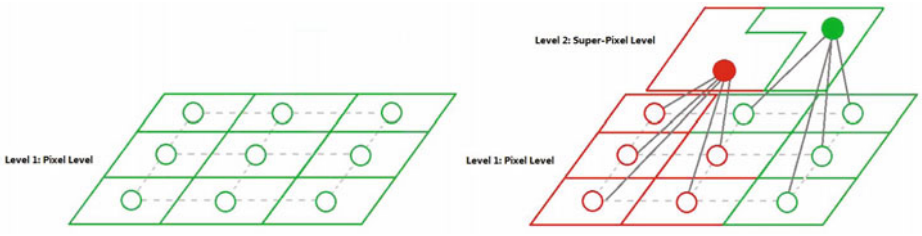

Fig. 5. Diagram of Dual-Level Conditional Random Fields

3 Segmentation with Dual-Level Conditional Random Field

Conditional Random Filed with higher order constrained has been used in segmentation problem and gets significant achievement recent years [15] [16] [17]. In this paper, we use a dual-level CRF to incorporate higher order constraint from super pixels obtained by JSEG [18].

There are two level vertices in the graph model of dual-level CRF. Vertices in level 1 are pixels in images and vertices in level 2 are super pixels produced by JSEG [18]. The structure of dual-level CRFs is illustrated in Figure 5. The edges only exist between vertices in level 1 and vertices between the two levels, while there are no edges between vertices in level 2, because superpixels are used as soft constraint in our model and final labeling results are obtained from level 1. The energy function is defined as follows:

$$E(x) = \sum_{i} \phi_i(x) + \sum_{i,j \in N(i)} \phi_{i,j}(x_i, x_j) + \sum_{i} \tilde{\phi}_i(x_i) + \sum_{i,i \in R_j} \tilde{\phi}_{i,j}(x_i, x_j) \quad (10)$$

where x_i is the label assigned to corresponding pixel or super pixel. ϕ_i and $\phi_{i,j}$ is the energy term defined on pixel level. $\tilde{\phi}_i$ is the super pixel unary term. $\tilde{\phi}_{i,j}$ is the pair wise term between pixel and its corresponding super pixel, which represent the higher order constraint by super pixel.

$\phi_i(x_i)$ and $\tilde{\phi}_i(x_i)$ have similar definition.

$$\phi_i(x_i) = \omega_{mask}\phi_i^{mask}(x_i) + \omega_{color}\phi_i^{color}(x_i) \quad (11)$$

where $\phi_i^{mask}(x_i) = -\log(M_{i,x_i})$. $\phi_i^{color}(x_i)$ is defined as the minus log of probability that current pixel's color in color distribution for the x_i label. In our experiment, the color distribution is represented as histograms. In $\tilde{\phi}_i(x_i)$, the mask probability is the average of probabilities of the pixels in its corresponding superpixel, and the color probability is defined as the similarity between color histogram of superpixel's and corresponding label's.

$$\phi_{i,j}(x_i, x_j) = \gamma \exp\left(-\beta \|I_i - I_j\|^2\right) \delta(x_i \neq x_j) \quad (12)$$

where β is set as $\left(2\left\langle \|I_i - I_j\|^2 \right\rangle\right)^{-1}$. γ is the model parameters.

Assuming i in level 1, j in level 2 and pixel i belongs to super pixel R_j , $\tilde{\phi}_{i,j}(x_i, x_j)$ is defined as Potts Model:

$$\tilde{\phi}_{i,j}(x_i, x_j) = \begin{cases} 0 & x_i = x_j \\ \tilde{\gamma} \exp\left(-\tilde{\beta} |R_j|\right) & other \end{cases} \tag{13}$$

where $|R_j|$ is the cardinality of super pixel R_j. $\tilde{\beta}$ is the inverse of the average over all super pixel sizes. $\tilde{\gamma}$ is model parameters just like γ. We use α-expansion algorithm to get the labeling result of the dual-level CRF.

4 Experimental Result

First of all, we label the training data, also called exemplar library, manually. Training data comes from Labeled Faces in the Wild database [19]. The reason for choosing this database is that images in LFW are general consumer images that are much less constraint than those used in face recognition researches, which is very good for validating the proposed algorithm. We manually labeled 1026 images. For each image, each pixel is assigned a label from the label set: background, hair, face or clothes. These images are divided into two halves randomly. One of them is used for learning similarity function and the parameters of CEM. The other is used for testing. The training and testing procedure is shown in Table 2 and Table 3 respectively. These dividing, training and testing procedure are repeated 10 times to get the experiment data.

The parameters of CEM are determined empirically. In consideration of speed, images are normalized as 72×72 and divided into 16×16 patches with step of 8 in both x and y directions. R_m in the training data are rectangles with sizes of 4×4, 8×8 and 16×16. The threshold is set as $\tau = 0.5$. The candidate number is set as $C = 10$. β in formula 8 is 8. For the CRF model parameters, $\gamma = \tilde{\gamma} = 8$, $\omega_{mask} = 1.6$ and $\omega_{color} = 0.4$. Both the max in formula 7 and 8 are set as 1000 to prevent an invalid inference result.

In Figure 6, we show some segmentation result by our algorithm. Hair style changes from bald to long and in different colors, it can be seen that our algorithm works robustly in the condition of large variation of hair shape and color and clutter background.

It takes us about 95 hours to train the SLS-based rankers. The training algorithm is applied independently for each local patch. So it can be extended on a distributed system easily to shorten the training time. To show the effectiveness of our similarity function, we used Normalized Discounted Cumulative Gain (NDCG) [22] to estimate the ranking quality. For a list of images sorted in descending order of the scores output by a learned ranking model, the NDCG score at the m-th image is computed as:

$$N_m = C_m \sum_{j=1}^{m} \frac{2^{r(j)} - 1}{\log(j+2)} \tag{14}$$

Table 2. Training algorithm

- Detect Face [20] and Eye locations [21] for each image and normalize it in the same size
- For each location, extracted the exemplar patch set $\{\mathcal{P}_i^k\}$
  - Generate training data as Table 1.
  - Generate strong ranker using RankBoost algorithm [14].
- End For

Table 3. Testing algorithm

- Detect Face [20] and Eye locations [21] for each image and **normalize it in the training size**
- For each location, extracted the exemplar patch set $\{\mathcal{P}_i^k\}$
  - Using strong ranker $H(\cdot)$ for current location to sort $\{\mathcal{P}_i^k\}$
  - Keep C exemplars as candidates
- End For
- Optimize CEM by alpha-expansion algorithm and get mask M
- **Inverse transform M to original image**
- Build dual-level CRF with as stated in section 3. Final segmentation is obtained by α-expansion [10] [11] [12].

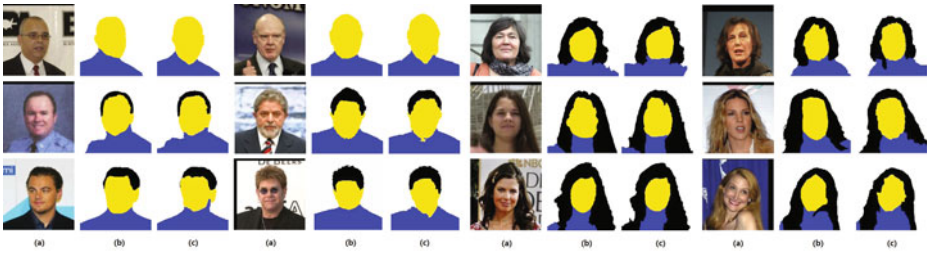

Fig. 6. Examples of Segmentation result. (a) Original Image (b) Manually labeled result (c) Our Segmentation Result. Color code for labels are: white - background, yellow - face, black -hair and blue - clothes.

where $r(j)$ is the rating of the j-th image and C_m is the normalization constant to make that a perfect ordering gets NDCG scores 1. In our experiment, $r(j) = C(P_0, P_j)$. In Figure 7(a), we illustrate our result of $m = 1$ in each location with comparison with a straightforward ranking algorithm using histogram and

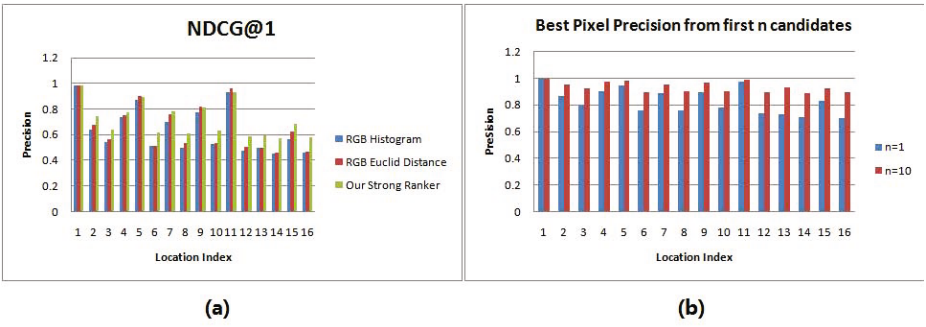

Fig. 7. (a) NDCG at the first image. (b) Best Pixel Precision from the first and the first ten candidate exemplars.

Euclid distance in RGB space. Our algorithm outperforms the straightforward one in almost every location. Especially in the difficult patches, the precision can be improved by 8% to 10%.

And we also test the pixel precision of the best candidate obtained by our strong rankers. Pixel level segmentation accuracy defined as:

$$presion = \frac{\sum_i^n \delta\left(L_i = \tilde{L}_i\right)}{n} \tag{15}$$

where n is the size of the current image. L_i and $\tilde{L}_i$ denotes the label of algorithm result and ground truth of pixel respectively. In Figure 7(b), we show the pixel precision of the best candidate in each location respectively. For most patches, our ranker can find acceptable exemplars for them.

In Figure 8, we give a qualitative CEM example with and without neighborhood consistency constraint. As explained before, independent search for exemplar patch can cause ambiguity inevitably. Neighborhood consistency constraint enforces the continuity between overlapped patches to improve the model robustness for ambiguity.

CEM without neighborhood consistency constraint can achieve a pixel precision of 84.6%. Incorporate the constraint into CEM can improve the precision to 86.3%. As numerous work [15] [16] [17] suggested, incorporating segments prior benefits the segmentation accuracy and robustness. In our problem, the precision of final segmentation result by Dual-Level CRF can reach 89.1%, which outperform Single-Level one by 1.5%. Although they bring only a slight increase in the segmentation accuracy quantitatively, they contribute significantly to subjective quality improvement on segmentation, just as stated in [15], a small increase in the pixel-wise accuracy will actually make a large improvement on the quality of segmentation.

We also test images not included in our manually labeled library. Some of the results are given in Figure 9 to demonstrate its generalization ability. Due to lack

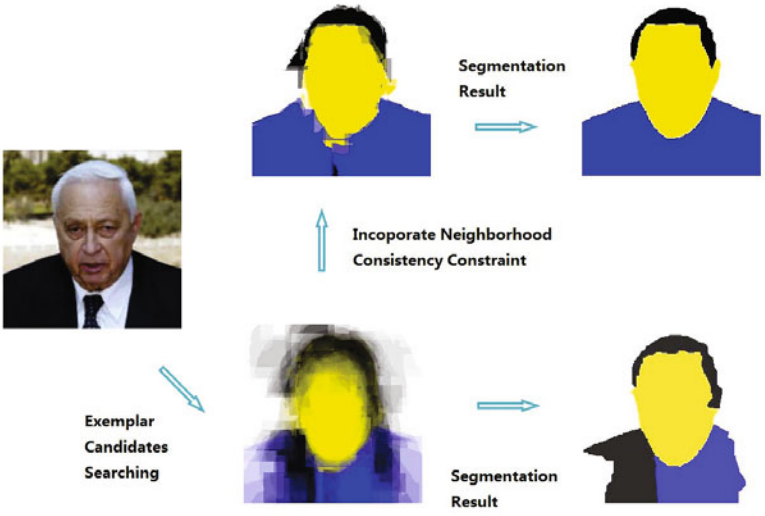

Fig. 8. Comparison between CEM with and without NLC Constraint

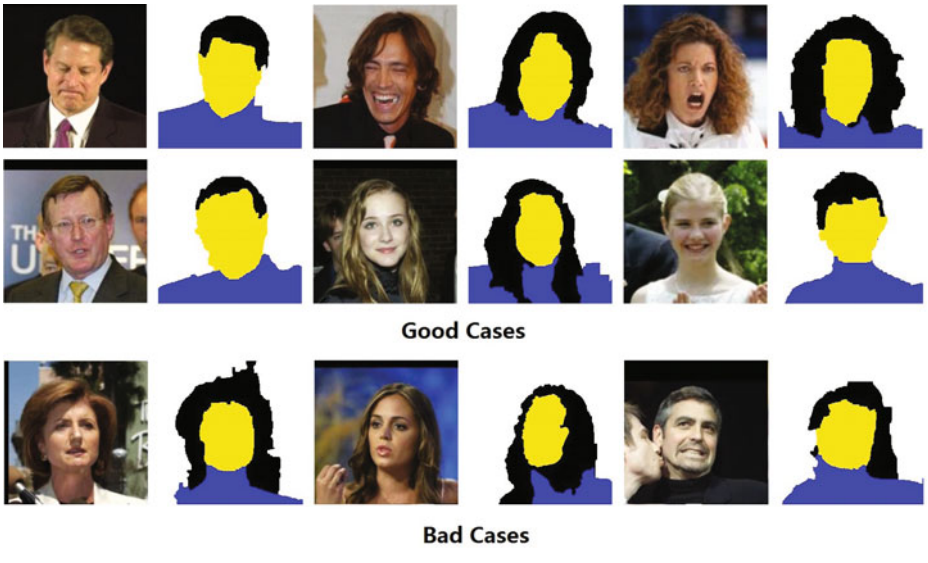

Fig. 9. More Segmentation Results

of technique details of [5], we have not tried to compare with it. Nevertheless we think our method is more powerful in dealing with various hair styles. We tested on 1000 selected face images in front view that are somewhat similar to the exemplars, and about 80% of segmentation results are subjectively acceptable. Unsatisfactory cases occur where hair is confused with background or shadows.

However, if a test image exists in the exemplar library, it will get the exact result. This characteristics guarantees our approach's extensibility since a new hair style can be easily extended by adding its manually labeled result into the exemplar library.

5 Conclusion

In this paper, we propose a novel Compositional Exemplar-based Model for hair style representation and segmentation. CEM generates hair style for the input image in the Divide-and-Conquer manner, which can be divided into the decomposition stage and composition stage naturally. For the decomposition stage, we design a group *Semantic Layout Similarity* features and combine them into a strong ranker by RankBoost algorithm. In the composition stage, we introduce the *Neighbor Label Consistency* Constraint to CEM and define the *consistency function* skillfully to ensure its *regularity*. Final segmentation result is obtained by the inference of Dual-Level Conditional Random Field. Experiment results on face images from Labeled Faces in the Wild data set show its effectiveness. In future, we will try to include side views into the library and speed up the searching procedure.

References

1. Paris, S., Briceo, H.M., Sillion, F.X.: Capture of hair geometry from multiple images. In: SIGGRAPH, Los Angeles, CA, United states, vol. 23, pp. 712–719 (2004)
2. Paris, S., Chang, W., Kozhushnyan, O.I., Jarosz, W., Matusik, W., Zwicker, M., Durand, F.: Hair photobooth: Geometric and photometric acquisition of real hairstyles. In: SIGGRAPH, vol. 27 (2008)
3. Ward, K., Bertails, F., Kim, T.Y., Marschner, S.R., Cani, M.P., Lin, M.C.: A survey on hair modeling: Styling, simulation, and rendering. IEEE Transactions on Visualization and Computer Graphics 13, 213–233 (2007)
4. Yacoob, Y., Davis, L.S.: Detection and analysis of hair. PAMI 28, 1164–1169 (2006)
5. chih Lee, K., Anguelov, D., Sumengen, B., Gokturk, S.B.: Markov random field models for hair and face segmentation. In: AFG, Amsterdam, pp. 1–6 (2008)
6. Shotton, J., Winn, J., Rother, C., Criminisi, A.: Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3951, pp. 1–15. Springer, Heidelberg (2006)
7. Borenstein, E., Ullman, S.: Combined top-down/bottom-up segmentation. PAMI 30, 2109–2125 (2007)
8. Wang, X., Tang, X.: Face photo-sketch synthesis and recognition. PAMI 31, 1955–1967 (2009)
9. Jojic, N., Perina, A., Cristani, M., Murino, V., Frey, B.: Stel component analysis: Modeling spatial correlations in image class structure. In: CVPR (2009)
10. Kolmogorov, V., Zabin, R.: What energy functions can be minimized via graph cuts? PAMI 26, 147–159 (2004)
11. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. PAMI 23, 1222–1239 (2001)

12. Boykov, Y., Kolmogorov, V.: An experimental comparison of min-cut/max- flow algorithms for energy minimization in vision. PAMI 26, 1124–1137 (2004)
13. Zhang, W., Shan, S., Gao, W., Chen, X., Zhang, H.: Local gabor binary pattern histogram sequence (lgbphs): A novel non-statistical model for face representation and recognition. In: ICCV (2005)
14. Freund, Y., Iyer, R., Schapire, R.E., Singer, Y.: An efficient boosting algorithm for combining preferences. Journal of Machine Learning Research 4, 933–969 (2004)
15. Kohli, P., Ladick, L., Torr, P.H.S.: Robust higher order potentials for enforcing label consistency. In: CVPR, Anchorage, AK, United states (2008)
16. Larlus, D., Jurie, F.: Combining appearance models and markov random fields for category level object segmentation. In: CVPR, Anchorage, AK, pp. 1–7 (2008)
17. Pantofaru, C., Schmid, C., Hebert, M.: Object recognition by integrating multiple image segmentations. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part III. LNCS, vol. 5304, pp. 481–494. Springer, Heidelberg (2008)
18. Deng, Y., Manjunath, B.S.: Unsupervised segmentation of color-texture regions in images and video. PAMI 23, 800–810 (2001)
19. Huang, G.B., Berg, T., Ramesh, M.: Labeled faces in the wild: A database for studying face recognition in unconstrained environments, University of Massachusetts, Amherst, Technical Report (2007)
20. Huang, C., Ai, H., Li, Y., Lao, S.: High-performance rotation invariant multiview face detection. PAMI 29, 671–686 (2007)
21. Zhang, L., Ai, H., Xin, S., Huang, C., Tsukiji, S., Lao, S.: Robust face alignment based on local texture classifiers. In: ICIP, vol. 2, pp. 354–357 (2005)
22. Jarvelin, K., Kekalainen, J.: Cumulated gain-based evaluation of ir techniques. ACM Transactions on Information Systems 20, 422–446 (2002)