Automatic User Interaction Correction via Multi-label Graph Cuts

Antonio Hernández-Vela[1,2] ahernandez@cvc.uab.cat
Carlos Primo[2] carlos.pg79@gmail.com
Sergio Escalera[1,2] sergio@maia.ub.es

[1]Computer Vision Center, Edifici O, Campus UAB, 08193 Bellaterra (Cerdanyola), Barcelona, Spain
[2]Dept. MAIA, Universitat de Barcelona, Gran Via 585, 08007 Barcelona, Spain

Abstract

Most applications in image segmentation requires from user interaction in order to achieve accurate results. However, user wants to achieve the desired segmentation accuracy reducing effort of manual labelling. In this work, we extend standard multi-label α-expansion Graph Cut algorithm so that it analyzes the interaction of the user in order to modify the object model and improve final segmentation of objects. The approach is inspired in the fact that fast user interactions may introduce some pixel errors confusing object and background. Our results with different degrees of user interaction and input errors show high performance of the proposed approach on a multi-label human limb segmentation problem compared with classical α-expansion algorithm.

1. Introduction

Object segmentation in images is still a challenging problem that requires from user interaction in order to obtain efficient and successful results. Objects in images use to have different visual structure at different parts. Moreover, same objects suffer from visual changes because of illumination or changes in the point of view, which makes difficult full precision of automatic object segmentation. The basic case of object segmentation consists on a binary labelling –foreground, background– of the pixels of an image. In this scope several algorithms have been proposed. Many recent approaches, formulate image segmentation as an energy minimization problem [2, 3, 9, 6]. These methods define an energy function whose minimum value corresponds to the optimal segmentation, and this energy is optimized via graph optimization. Following this graph-based methods, works like [5, 8] also introduce spectral clustering theory in the framework, which uses eigenvectors and eigenvalues of the similarities between pixels. Some of these techniques have been also developed in order to deal with multi-label image segmentation, where more than one objects (or parts of objects) are segmented at the same time. Some techniques start oversegmenting small uniform regions, called superpixels [4, 10, 7], which preserve the contours of the objects. Then, final segmentation is applied over these dense regions in order to reduce computation time.

One of the main problems we can find when trying to segment an object in an image, is the complexity of the background. In many cases, we have to deal with the problem of camouflage, i.e., some parts of the object –or the object itself– could be confused with the background in terms of colour similarity. Furthermore, we also have to face problems like changing illumination conditions or occlusions, just to mention a few. Because all of these problems, the user is asked to give some clue about the location of the object he/she wants to segment in the scene in order to reduce segmentation ambiguities of the segmentation algorithm.

There are many ways in which the user can interact in order to provide useful information about the desired object to segment. In [9] we can find a brief list of well-known interactive approaches for object segmentation, each one of them with a different kind of human interaction. On one hand, some approaches like Magic Wand, just expect the user to click some points or small regions inside the object. On the other, there also exist some other approaches like Intelligent Scissors or Bayes matting, which ask the user to draw a rough approximation of the contour of the object instead of regions inside it. One method which has been proved to get successful results in image segmentation is Graph-cuts [3]. This method, which was originally designed to work only with gray-scale images, expects the user to draw some small strokes in both the object to segment and the background, similarly as in the case of Magic Wand. An extension to this Graph-cuts method is GrabCut [9], which works with RGB images instead of just gray-scale ones. Furthermore, GrabCut proposes a new interaction system consisting in just selecting a bounding box enclosing the object to segment. However, this method is iterative, and lets the user to interact after each iteration, drawing the same kind of strokes as in the original Graph-cuts in order to correct possible wrong segmented parts.

Going more in detail, Graph-cut and GrabCut methods, the strokes the user draws in order to mark foreground and background pixels are supposed to not contain any errors, i.e., the pixels the user marks as foreground do not contain any pixel of the background, and vice-versa. Making this assumption, the algorithm fixes those pixels to the class the user specified, without any chance of changing their labels during the execution of the algorithm. However, the user could eventually make some mistakes when drawing the initial strokes, resulting in an uncorrectable wrong segmentation. This case, is even more plausible when thinking of a scenario where the user has to segment several images with several objects, eventually increasing the tiredness of the user, and decreasing the attention he/she pays. As a result, the user would probably make some mistakes after a while.

In this work, we extend standard multi-label α-expansion Graph Cut algorithm by analyzing the interaction of the user in order to modify the object model and improve final segmentation of objects. The approach is inspired in the fact that fast user interactions may introduce some pixel errors confusing between object and background. We present both quantitative and qualitative comparison of the original approach and our proposal on the Human Limb data set [1]. Our results with different degrees of user interaction and input errors show higher performance of the proposed approach on a multi-label human limb segmentation problem compared with classical α-expansion algorithm.

The rest of the paper is organized as follows: Section 2 introduces the Multi-label Graph-cuts segmentation framework, Section 3 presents our proposal, Section 4 contains the experimental setup we conducted, and finally, Section 5 concludes the paper.

2. Multi-label Graph-cuts Segmentation

Graph-cuts is an energy minimization framework which has successfully been applied to the problem of image segmentation, in both binary and multi-label cases. This framework defines an energy function specific to the problem we want to solve, in such a way that the minimum value of this energy corresponds to the optimal solution. Therefore, in our case Graph-cuts will find the optimal segmentation.

Given a color image, let us consider the array $z = (z_1, .., z_n, .., z_N)$ of N pixels where $z_i = (R_i, G_i, B_i)$, $i \in [1, .., N]$ in RGB space. The segmentation is defined as an array $\boldsymbol{\alpha} = (\alpha_1, ..\alpha_N)$, $\alpha_i \in \{1, .., L\}$, assigning a label to each pixel of the image indicating the class it belongs to. L is the total number of classes of our problem.

An initial labelling $T = \{T_1, .., T_L\}$ is defined by the user strokes, consisting on L sets of marked pixels –one for each possible label–. These pixels the user has marked, are clamped as their corresponding label –that means Graph-cuts will not be able to modify these labels–, while the rest of unlabelled pixels are actually the ones the algorithm will be able to label.

Color information is introduced by GMMs. A full covariance GMM of K components is defined for each group of pixels labelled as each one of the possible classes, parameterized as follows,

$$\boldsymbol{\theta} = \{\pi(\alpha, k), \mu(\alpha, k), \Sigma(\alpha, k), \alpha \in [1, .., L], k \in [1, .., K]\}, \tag{1}$$

being π the weights, μ the means and Σ the covariance matrices of the model. We also consider the array $\mathbf{k} = \{k_1, .., k_i, ..k_N\}$, $k_i \in \{1, ..K\}$, $i \in [1, .., N]$ indicating the component of the corresponding GMM (according to α_i) the pixel z_i belongs to. The energy function for segmentation is then,

$$\mathbf{E}(\boldsymbol{\alpha}, \mathbf{k}, \boldsymbol{\theta}, \mathbf{z}) = \mathbf{U}(\boldsymbol{\alpha}, \mathbf{k}, \boldsymbol{\theta}, \mathbf{z}) + \mathbf{V}(\boldsymbol{\alpha}, \mathbf{z}), \tag{2}$$

where $\mathbf{U}$ is the likelihood potential, based on the probability distributions $p(\cdot)$ of the GMM:

$$\mathbf{U}(\boldsymbol{\alpha}, \mathbf{k}, \boldsymbol{\theta}, \mathbf{z}) = \sum_i -\log p(z_i | \alpha_i, k_i, \boldsymbol{\theta}) - \log \pi(\alpha_i, k_i) \tag{3}$$

and V is a regularizing prior assuming that segmented regions should be coherent in terms of color, taking into account a neighborhood C around each pixel,

$$\mathbf{V}(\boldsymbol{\alpha}, \mathbf{z}) = \gamma \sum_{\{m,n\} \in C} \Omega\left(\alpha_n, \alpha_m\right) \, \exp\left(-\beta \|z_m - z_n\|^2\right), \tag{4}$$

being $\Omega\left(\alpha_n, \alpha_m\right)$ a function that penalizes relations between pixels z_n and z_m depending on their labellings, assigning some pre-set costs to each possible combination of labels. With this energy minimization scheme and given the initial labellings by the user, the final segmentation is performed using an optimization algorithm. In the case of binary segmentation –$L = 2$–, the min-cut algorithm [2] can be applied in order to find the optimal solution. However, when $L > 2$, min-cut cannot be applied directly and the optimal solution cannot be found. Instead of that, two different algorithms based on min-cut can be applied depending on the nature of the energy function [3]. On one hand, α - β swap algorithm can be applied when the defined energy function is *semi-metric*. On the other hand, α-expansion can find a better approximation of the minimum, but only when the energy function is *metric*. In our case, we base on α-expansion algorithm for our proposal. The method is summarized in Algorithm 1.

3. Human interaction correction

Our proposed interaction correction algorithm acts before initializing the $\boldsymbol{\alpha}$ vector with the initial labelling T

Algorithm 1 Original Graph-cuts algorithm.

```
$T$ initialization with manual annotation.
Initialize $a_i = l$ for $z_i \in T_l$, $l \in [1, .., L]$.
Initialize GMMs from sets $a_n = l$ , $l \in [1, .., L]$, with $k$-means clustering.
Assign GMM components to pixels.
Learn GMM parameters from data z.
Estimate segmentation: $\alpha$-expansion.
```

performed by the user. The method basically checks the "fitness" of the pixels selected by the user for each label against the GMMs of the remaining labels. This way, if we find some inconsistencies between the probabilistic models and the labels for any pixel, we remove it from the set of marked pixels. Given T, we check the pixels marked by the user and compare them to the learnt GMMs for each label. For each pixel in region T_l, we compute its probabilities of belonging to each GMM as follows,

$$p\left(z_i|\alpha_i = l, k_i, \boldsymbol{\theta}^k\right), l \in [1, .., L]. \quad (5)$$

Furthermore, for each pixel we also compare the weight $\pi(\alpha, k)$ of the actual label GMM component this pixel belongs to, and the corresponding weight in the rest of GMMs for all the labels. This comparison is a simple threshold τ over the difference in probabilities, and a direct comparison over GMM component weights. If this difference is greater than τ for at least one of the labels, then this pixel is taken away from its corresponding T_i set, i.e., the user is not trusted for this pixel, and the segmentation algorithm will be free to label this pixel with the appropriate label. Moreover, the GMMs are recomputed with the new reduced set of initial labels in order to remove false positives when building the probabilistic model based on color. The method is summarized in Algorithm 2.

Fig. 1 shows an example of applying Graph-cuts segmentation without and with the proposed human interaction correction method. One can see how the user initially labelled the image. Specifically, we will focus on the labelling of the lower-body. The user has taken a short-cut, and included some background pixels as lower-body. This error results in a wrong segmentation as can be seen in the corresponding resulting mask (top row). When applying our interaction correction approach, not only we delete these wrongly labelled pixels from the interaction, but we also remove them for the estimation of the corresponding GMM. Taking a look at the GMM component coloured in red in Fig. 1 (c) top, we can see it has a significant overlap with the background GMM –Fig. 1 (d) top–, so we can expect this component has been estimated with the erroneous background pixels. Indeed, we can see how this component disappears when applying our correction approach, resulting in a more accurate probabilistic color model.

Algorithm 2 Multi-label Graph-cuts with interaction correction proposal.

```
$T$ initialization with manual annotation.
Initialize $a_i = l$ for $z_i \in T_l$, $l \in [1, .., L]$.
Initialize GMMs from sets $a_n = l$ , $l \in [1, .., L]$, with $k$-means clustering.
for $l = 1 \to L$ do
    for $z_i \in T_l$ do
        for $m \neq \alpha_i$ do
            if $p(z_i|\alpha_i = l, k_i, \boldsymbol{\theta}^k) < \left(p(z_i|\alpha_i = m, k_i, \boldsymbol{\theta}^k) + \tau\right)$
            and $\pi(\alpha_i = l, k_i) < \pi(\alpha = m, k_i)$ then
                $T_l = T_l \setminus \{z_i\}$
            end if
        end for
    end for
end for
Reinitialize $a_i = l$ for $z_i \in T_l$, $l \in [1, .., L]$.
Renitialize GMMs from new sets $a_n = l$ , $l \in [1, .., L]$, with $k$-means clustering.
Assign GMM components to pixels.
Learn GMM parameters from data z.
Estimate segmentation: $\alpha$-expansion.
```

4. Experimental section

We performed several experiments to evaluate our proposal and compare it with the original Graph-cuts segmentation algorithm. Our objective is to see the reliability of the methods for different degree of interaction. In order to do that, we simulated human interactions by randomly choosing a set of points from each label looking at the ground-truth. This random selection is performed several times in order to get a bank of different interactions, and the number of points selected for each label is a fixed proportion of the number of pixels from that class in the ground-truth.

On one hand, we considered the case where the user does not make any mistake, i.e., there are no errors in the initial labelling. On the other, we also marked some erroneous pixels, in a fixed proportion over the number of previously selected correct pixels. Each one of the computed interactions for each image is then used to compute their corresponding segmentations. Additionally, we also define a new bank of interactions by incrementally combining the previously computed ones. This way, we can see the effect of the complexity of the interactions in the segmentation results. An example of different interactions can be seen in Fig. 4.

Next, we describe the data, methods, and validation protocol of the experiments.

Data: For the evaluation of our method we used the Human Limb data set [1], which contains images from 25 different people in complex backgrounds. This data set provides a ground-truth with the labelling of 14 different body

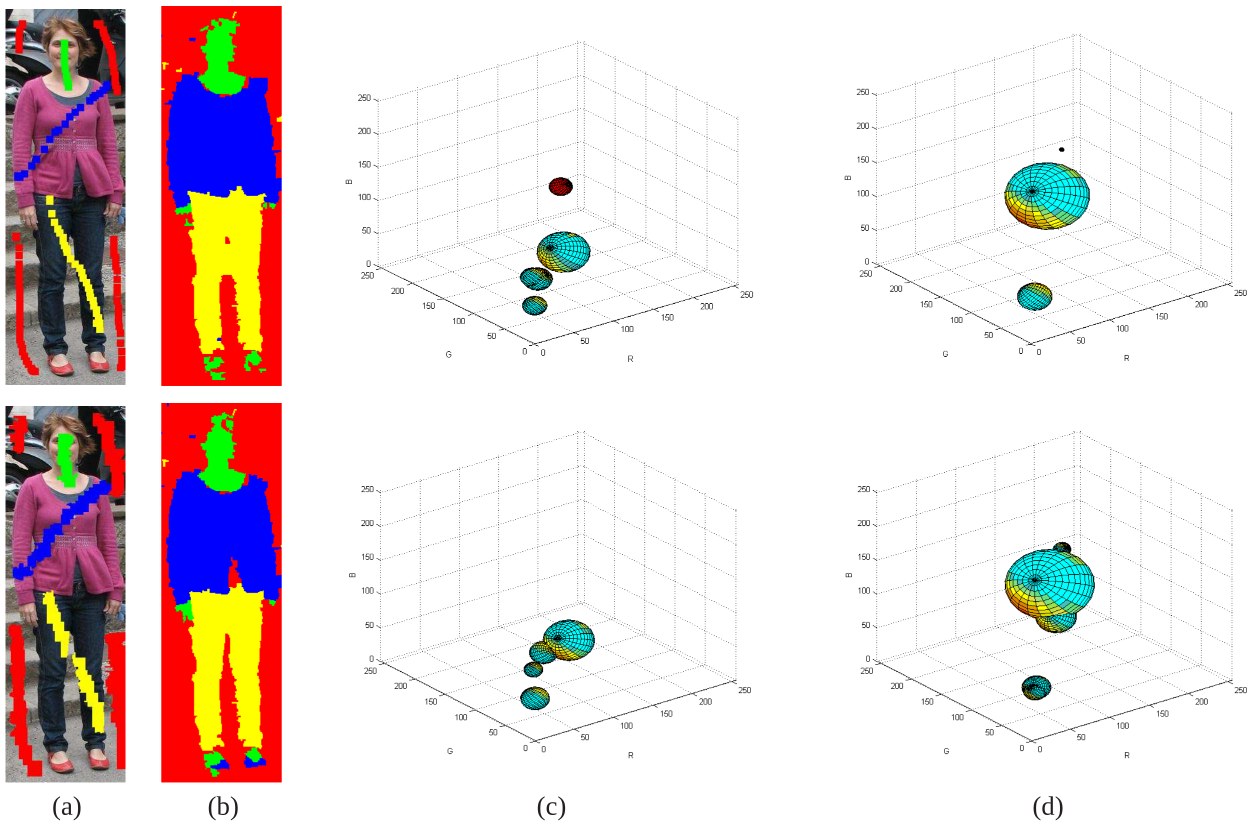


Figure 1. An example of interaction correction. (a) Initial pixel labelling by the user for the 4 labels, (b) Final segmentation mask, (c) GMM for the lower-body, (d) GMM for the background. First row shows an example using the original Graph-cuts approach, and second row shows the same example with or proposal.

parts for each image, as shown in Fig. 2(a). From the total 227 images, we randomly selected 10 of them –assuring we take only one image from each person– to make the comparison. Moreover, we re-grouped the original labels of the ground-truth in 4 groups: upper-body, lower-body, head & hands, and background, as shown in Fig. 2(b).

Method: We evaluated the performance of the original α-expansion method, and our interaction correction approach previous to segmentation. In both cases we set the λ parameter to 50, and the number of GMM components to $K = 5$. The $\Omega\left(\alpha_n, \alpha_m\right)$ function was defined as follows:

$$\Omega\left(\alpha_n, \alpha_m\right) = \left\{ \begin{array}{lll} 0 & \text{for} & \alpha_n = \alpha_m \\ 1 & \text{for} & \alpha_n \neq \alpha_m \end{array} \right. \tag{6}$$

This considers the basic case in which the inter-label costs are the same for all the possible combinations of different labels. Moreover, we pre-processed the images computing superpixels with the method of [7] in order to reduce computation time and memory requirements. Once computed the superpixels, we take the mean RGB value of each one of them as the information used for the computation of the Graph-cuts potentials. Finally, the best threshold we found for the comparison of GMM pixel probabilities was $\tau = 0.1$.

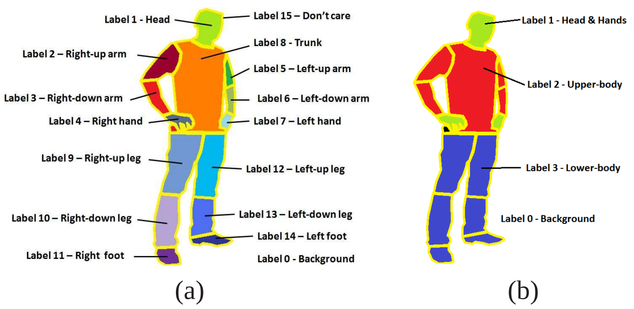


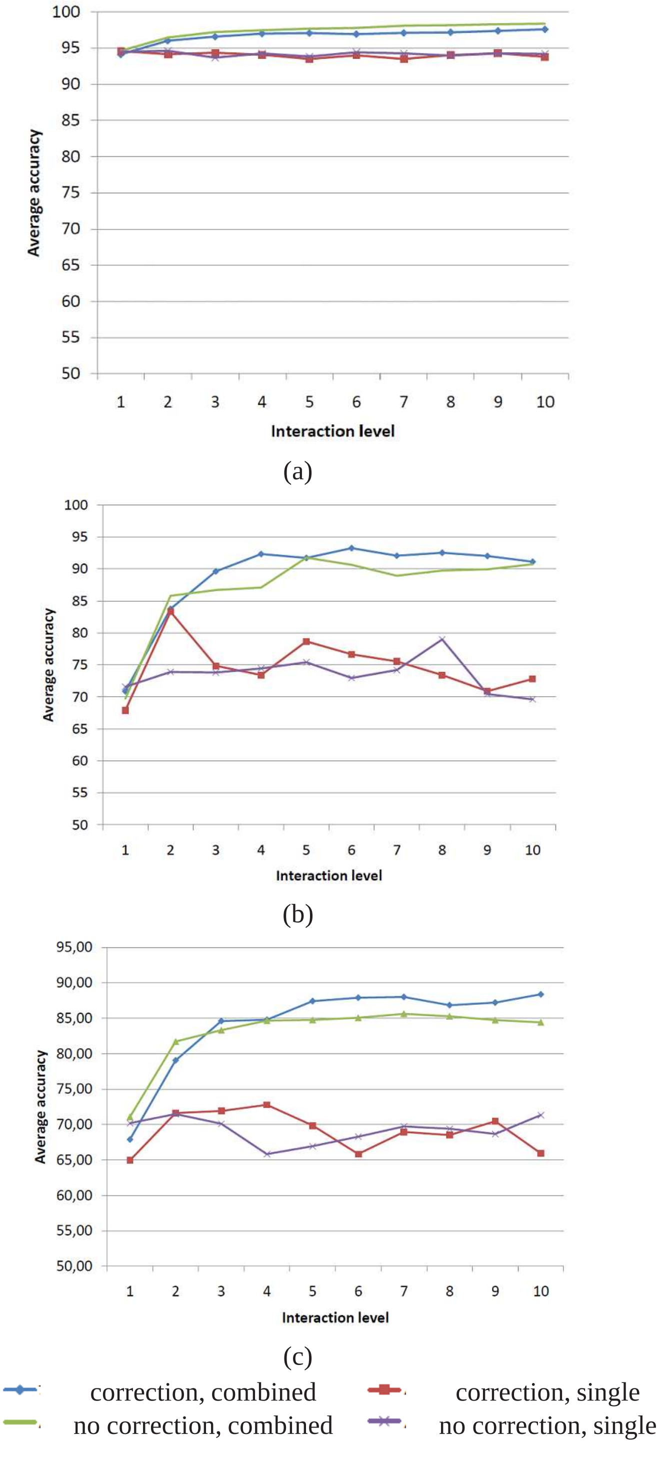
Figure 2. Label rearrangement of Human Limb data set.

Validation: As a measurement for the validation of the method, we evaluated the mean average overlap accuracy for the 4 defined labels. For all the interaction scenarios –with and without error, single and combined interaction masks– we used a bank of 10 different interactions. At each one of these interactions, we randomly selected for each label the 0.2% of the total number pixels for that label in the ground-truth. In the case of erroneous interactions, we additionally selected the 25% of correctly marked pixels as new erroneous interactions, choosing a random label different from the actual one.

In Fig. 3 we can see some quantitative results corresponding to different levels of erroneous interactions. We can firstly see how, when no errors exist in the interactions, our proposal gets similar results –96.71% accuracy in combined mode, 94.04% in single mode– as the classical approach where no correction is performed –97.44% combined, 94.21% single–. This small decrease in the accuracy is caused by the possible erasing of correctly labelled pixels, resulting in a poorer interaction and thus, in a slightly worse result. Apart from that, we can see how the addition of pixels in the interactions –combined mode– incrementally improves the segmentation results compared to the single mode.

Taking a look at the case where errors are introduced in the interactions, one can see that our approach gets better results in almost all the levels of interaction, specially in the case of the combined mode. Moreover, the improvements in the obtained results are higher in the case of 25% of error rather than in the case of 45%. This tells us that the method can deal with a small quantity of incorrectly labelled pixels, but when this error is high, the method cannot work as expected.

More specific qualitative results can be found in Fig. 4. In these examples we can clearly see how the method can correct some erroneous labelled pixels, which can lead to bigger wrongly segmented areas, as in the two first examples. Although the proposed method is able to correct a high number of mistaken interactions, we can see in those specific examples how some of them still remain uncorrected. Furthermore, we can appreciate how the massive addition of wrong interactions –last 4 columns, combined mode of interactions– leads to noisy segmentations in the classic approach, and how they can be smothered with our proposal.



(a)

(b)

(c)

Figure 3. Mean Average Accuracy: (a) No errors, (b) 25% of error, (c) 45% of error. X axis represents the different interactions from the bank, Y axis shows the mean average accuracy for all labels.

5. Conclusion

We proposed a method for the correction of wrong human interactions in the problem of multi-label image segmentation, and applied it to the segmentation of human body parts in the Human Limb data set. The interaction correction is based on analyzing, for each pixel, the probabilities of belonging to each one of the GMMs corresponding to each existing label together with the GMM density information. The proposed approach reassign GMM components based on low confidence probabilities and recompute RGB models to improve final segmentation.

Results are presented both qualitative and quantitatively, showing the improvement of our proposal when erroneous labellings are present. Moreover, an extensive validation has been performed by automatically generating random initial labellings simulating human interaction, and combining them in order to see the influence of the addition of correct and incorrect initial labellings in the results obtained with the Graph-cuts segmentation framework.

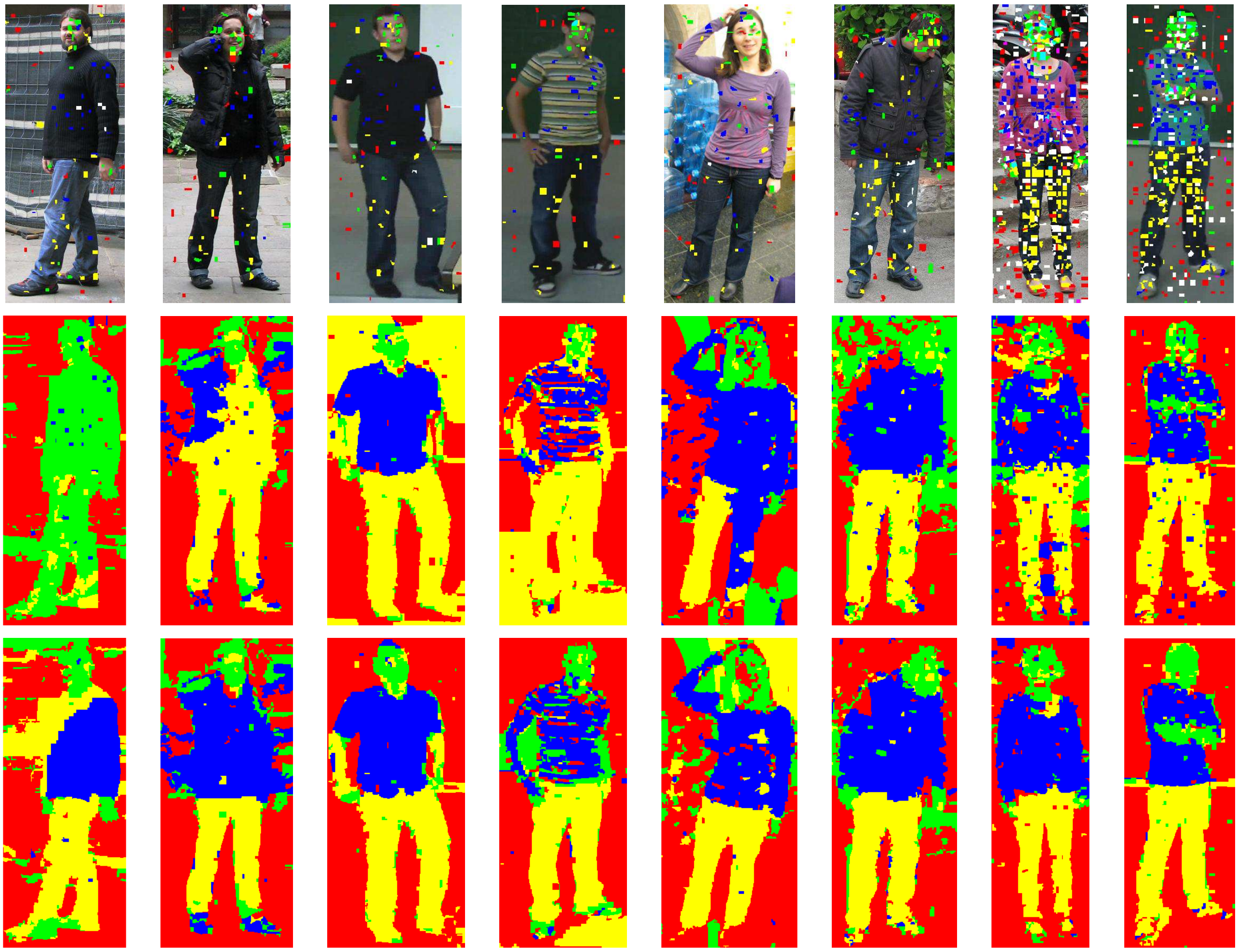


Figure 4. Qualitative results. First row shows the original images with the initial interaction. Second and Third rows show the obtained results for the classical Graph-cuts approach without and with our interaction correction proposal, respectively. Results in columns 1 to 4 correspond to the simple interactions mode, and columns 4 to 8 show results using the combined interactions mode.

Acknowledgements

This work has been supported in part by the projects: La Marató de TV3 082131, TIN2009-14404-C02, and CONSOLIDER-INGENIO CSD 2007-00018.

References

[1] Human limb data set. http://www.maia.ub.es/∼sergio/linked/humanlimbdb.zip.

[2] Y. Boykov and G. Funka-lea. Graph cuts and efficient n-d image segmentation. *International Journal of Computer Vision*, 70:109–131, 2006.

[3] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(11):1222 –1239, nov 2001.

[4] D. Comaniciu and P. Meer. Mean shift: a robust approach toward feature space analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(5):603 –619, may 2002.

[5] L. Grady. Random walks for image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(11):1768 –1783, nov. 2006.

[6] A. Hernandez, M. Reyes, S. Escalera, and P. Radeva. Spatio-temporal grabcut human segmentation for face and pose recovery. In *CVPR Workshops*, pages 33 –40, june 2010.

[7] A. Moore, S. Prince, and J. Warrell. Lattice cut - constructing superpixels using layer constraints. In *CVPR*, pages 2117 –2124, june 2010.

[8] H. Qiu and E. R. Hancock. Image segmentation using commute times. In *In BMVC*, pages 929–938, 2005.

[9] C. Rother, V. Kolmogorov, and A. Blake. ”GrabCut”: interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph.*, 23(3):309–314, Aug. 2004.

[10] J. Shi and J. Malik. Normalized cuts and image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):888 –905, aug 2000.