

# System for the Automated Segmentation of Heads from Arbitrary Background

Benjamin Prestele<sup>1</sup>, David C. Schneider<sup>1</sup>, and Peter Eisert<sup>1,2</sup>

<sup>1</sup> Fraunhofer Institute for Telecommunications, Heinrich Hertz Institute,  
Einsteinufer 37, 10587 Berlin, Germany  
<http://www.hhi.fraunhofer.de>

<sup>2</sup> Humboldt University of Berlin, Visual Computing,  
Unter den Linden 6, 10099 Berlin, Germany  
<http://www.informatik.hu-berlin.de>

**Abstract.** We propose a system for the fully automated segmentation of frontal human head portraits from arbitrary unknown background. No user interaction is required at all, since the system is initialized using a standard eye detector. Using this semantic information, the head region is projected into a normalized polar reference frame. To setup an energy function for segmentation, regional and boundary models are learned from the image data. A robust non-local boundary detection scheme is proposed, which minimizes the similarity of fore- and background regions. Additionally a shape model learned from a large set of manually segmented images is employed as prior information to encourage the segmentation of plausible head shapes. Segmentation is performed as an iterative optimization process, using two different graph-based algorithms.

**Keywords:** automatic segmentation, human heads, shape prior, graph-cut optimization, shortest-path search, polar coordinates

## 1 Introduction

Image-based object segmentation forms the basis for a large variety of methods and applications, including 3D reconstruction for human-computer interaction and augmented reality applications, semantic image retrieval, visual effects compositing in movies, as well as different types of security related applications.

The goal is to find a labeling that separates foreground pixels depicting the object from all other pixels in the background. Since pixels do not carry semantic information about the depicted objects, image-based object segmentation is closely related to image understanding – a still unsolved problem and active area of research in computer vision. Most approaches therefore rely on additional constraints in order to be robust. Typically they either derive hints from interactive user input, or they restrict the type of input data to a very specific class of objects and use a predefined model which captures common properties.

Segmentation may then be formulated in terms of an optimization problem over the image and model data, which tries to satisfy all given constraints. The

use of graphcuts has shown to be a very efficient technique for this purpose [1][2][8][9].

This paper focuses on the fully automated segmentation of a very specific class of objects, namely frontal human heads, from a single still image. Given the frontal portrait of an arbitrary person in front of an unknown background, the system detects and cuts out the head of the person, including hair. No interaction at all is required from the user, though the system could easily be extended to allow for a manual refinement stage after the automatic cutout is complete (e.g. using the technique described in [12]).

To demonstrate the performance of the system, we present results from a large database of test images, as well as examples from a real-world use case, where users can upload photos to a website to automatically create personalized movie clips.

### 1.1 Related Work

Segmentation being a long standing problem in computer vision, a comprehensive review of literature in this field would be very extensive. With the focus on recent years, particularly the rediscovery of graphcut optimization for segmentation tasks has drawn much attention by the community. The segmentation process is typically initialized by the user interactively labeling few sample pixels in the image as certain fore- and background [1], or by specifying a single bounding box enclosing the full foreground [11]. A statistical model, e.g. of color, is derived from the samples and the segmentation is then computed by minimizing an error function that penalizes cutting homogeneous regions with respect to the model. For a class of error functions investigated in [8], global optimization can be performed using graphcuts.

The little amount of user interaction required by those approaches make them suitable also for semi- or fully-automated segmentation schemes. Since color and gradient information is however not always sufficient for a plausible segmentation, several attempts were made to integrate shape priors into the optimization process [4, 7, 9, 14]. For the special case of faces, the authors of [10] propose to use an elliptical shape prior.

Our system builds upon the existing work described above and proposes several enhancements to yield a more robust and fully automated solution. Instead of incorporating geometric primitives, such as ellipses, as shape priors, a contour model is learned from a large database of manually segmented frontal head images. This increases robustness of the segmentation by encouraging plausible head shapes. In addition to using local gradient information as part of our segmentation cost function, a more global boundary detection scheme is introduced, which maximizes dissimilarities of fore- and background regions. Finally, the use of a special coordinate system simplifies the shape of the head boundary from a roughly elliptical to a horizontal segmentation path. This enables the application of graph search algorithms that otherwise would be difficult to integrate. Optimization itself is conducted in an iterative manner, in order to successively refine the cutout path.

## 2 Automatic Segmentation

Segmentation is performed with no user interaction required at all. The process comprises the following steps:

1. System initialization using a standard eye detector to create an initial trimap based on the eyes' location and distance.
2. Transformation of the input image into a rotation and scale normalized polar reference frame.
3. Setup of a color-based regional model using Gaussian Mixtures.
4. Setup of a boundary model using an image-derived non-local contour measure, a learned head contour prior, and local image gradients.
5. Iterative graph-based segmentation using Dijkstra shortest-path search and graphcut optimization.

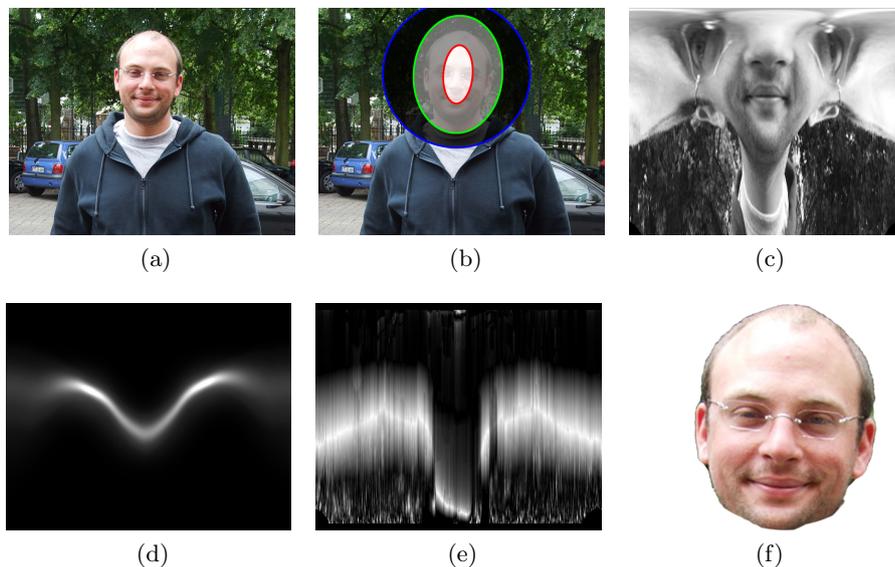


Fig. 1: (a) Original image; (b) Initial trimap with safe foreground (white area/red ellipse), safe background (black area/blue circle), and undefined region (grey/green ellipse); (c) Rotation and scale normalized polar reference frame; (d) Head contour prior map in polar space; (e) Boundary cost map in polar space; (f) Segmentation result.

### 2.1 System Initialization

Similar to [10], we use a standard eye detector, such as [13], to initialize the system and to replace the user input typically required by interactive schemes.

Being a quite reliable feature to detect, the eyes' position and distance are used to estimate the region of interest in the image and to normalize for rotation and scale of the depicted head, as described in Section 2.2.

Additionally, the eye coordinates are used for a conservative estimate of initial "safe" fore- and background regions (shown as black and white areas in Fig. 1b), using two differently sized ellipses. The position and size of the ellipses were determined empirically from a large set of sample images. The boundary of the head is assumed to be located within the undefined region in between, as illustrated as the gray region in Fig. 1b.

## 2.2 Polar Reference Frame

All subsequent processing is performed in a normalized reference frame. Given the eye position and distance from the feature detector, the image is transformed into polar coordinates, denoted by  $(\theta, r)$ . The midpoint of the line segment connecting the eyes is taken as the center, with the zero-angle axis pointing in the direction of the right eye, hence normalizing rotation. Similarly, the eye distance is used to normalize for scale. The region of interest is then resampled with angular resolution  $A$  and radial resolution  $R$  into the polar frame using bilinear interpolation. Figure 1c shows an example of such a projection into the reference frame.

Working in the polar frame has several advantages. It is a straightforward approach to normalize for rotation and scale, and all subsequent processing can be performed on images of the same size (i.e.  $A \times R$  pixels); computation time is thus controllable despite of varying input image sizes. A common reference frame also simplifies computation of probabilistic information from a set of training images and relating this information to a new image. Finally, the use of polar coordinates changes the topology of the search space, as the regions inside and outside of the head will be on opposite sides at the top and the bottom of the polar frame. This reduces the complexity of the segmentation boundary from an elliptical shaped path in Cartesian space into a roughly horizontal path in polar space. This will be quite handy during the graph-based segmentation discussed in Section 2.6.

## 2.3 Color Model

Energy functions for segmentation tasks typically comprise a regional term, as well as a boundary term. While the regional term models – for each pixel independently – the probability of belonging to either fore- or background, the boundary term favors a homogeneous labeling among adjacent pixels.

For the regional term, distinct color models for fore- and background regions are learned from the image data. As the segmentation process is performed iteratively, corresponding samples are selected using either the initial trimap constructed after eye detection, or a trimap constructed from the binary cutout mask of a previous segmentation iteration after morphological expansion. After evaluating several parametric and non-parametric model types empirically, we

chose Gaussian Mixture Models (GMMs) over  $L^*a^*b$  color space as the color model; the evaluation is described below.

Standard Expectation Maximization (EM) is used to learn a foreground and a background GMM with typically  $K = 7$  components each. For this training step, only corresponding pixels labeled by the trimap as either certain fore- or background are used. Denoting  $\mathbf{c}_{\theta,r}$  as the color of a pixel at  $(\theta, r)$  in the image, and  $L_{\theta,r} \in \{\text{'fg'}, \text{'bg'}\}$  as its label, the GMMs give us conditional probabilities  $p(\mathbf{c}_{\theta,r} | L_{\theta,r})$  for both possible values of  $L_{\theta,r}$ . This probability will later be used to derive the regional term of our segmentation energy function.

To select a color model, we tested several model types – parametric and non-parametric – on a set of 40 highly diverse images that were segmented manually into a head region (with hair) and a background. Each tested model type yields two numeric values for each pixel indicating the degree to which the pixel belongs to the foreground and to the background class; for some but not all models these are probabilities. A binary segmentation map was obtained by taking the larger value as the winner. The map was then compared to the ground truth. In total, seven model types were tested: Gaussian Mixture Models (GMMs) in RGB (1) as well as  $L^*a^*b$  color space (2), both normalized;  $k$ -nearest-neighbor models in normalized RGB (3) and  $L^*a^*b$  color space (4). Support vector machine (SVM) classification in normalized RGB (5) and  $L^*a^*b$  space (6), as well as in a combined, six-dimensional RGB+ $L^*a^*b$  space (7).

Fig. 2 shows the results. A second evaluation was computed after a morphological refinement of the class maps. In conclusion, the choice of model has no large effect on the color classification performance. GMMs in  $L^*a^*b$  color space perform best and have a reasonably small variance. They have the additional advantage of allowing a direct probabilistic treatment of color as described above.

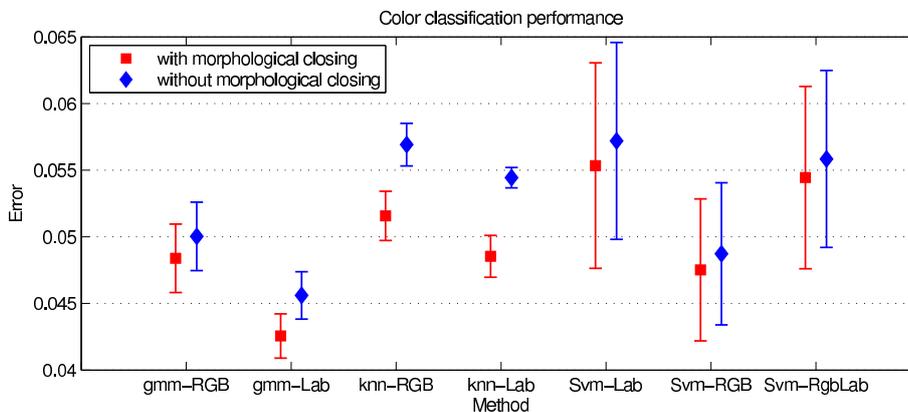


Fig. 2: Results of the color model type evaluation. Error measured as fraction of misclassified pixels.

## 2.4 Boundary Model

The boundary term of the segmentation energy function is typically derived only from the gradient magnitude of adjacent pixels. This has the disadvantage that local noisy structures in the image, such as hair, can have a strong impact on the segmentation boundary. More noise-tolerant edge detection schemes, such as the Canny edge detector, can be used to avoid this problem, but the edge information will still be very local. We therefore propose to additionally use a more global boundary measure.

The key idea is that a good segmentation boundary will provide a separation of the head from the background in a non-local optimum sense. Assuming that the trained foreground color model is sufficiently distinct, the optimum segmentation boundary should, with respect to the color model, maximize the distance of probabilities accumulated over all pixels on both sides of the boundary. In the polar frame this can be easily modeled independently for each column  $\theta$  of the polar image, which corresponds to finding an optimum split on a ray in Cartesian space, cast from the eye center into the background.

Let the current segmentation boundary be represented by a vector of radii  $\mathbf{q} = [r_1 \dots r_A]^T$  as described above. For a pixel at location  $(\theta, r)$  this implies the labeling

$$L_{\theta, r | \mathbf{q}} = \begin{cases} \text{'fg' } & \text{if } r \leq \mathbf{q}_\theta \\ \text{'bg' } & \text{otherwise} \end{cases} \quad (1)$$

Now let  $\mathbf{f}$  be a normalized histogram describing the distribution of foreground GMM probabilities in the foreground given segmentation boundary  $\mathbf{q}$ . Let similarly be  $\mathbf{b}$  the normalized histogram of GMM *foreground* probabilities in the background implied by  $\mathbf{q}$ . The optimum boundary  $\mathbf{q}$  is found by maximizing the  $\chi^2$ -distance between  $\mathbf{f}$  and  $\mathbf{b}$  defined as

$$\chi^2 = \sum_{i=1}^K \frac{(\mathbf{f}_i - \mathbf{b}_i)^2}{\mathbf{f}_i + \mathbf{b}_i} \quad (2)$$

with  $K$  the number of histogram bins.

Maximization is done iteratively: in each iteration, the elements of  $\mathbf{q}$  – i.e. the radii of the segmentation boundary – are updated in order to increase  $\chi^2$ . Therefore it is not necessary to recompute the full histograms. Instead, a pixel  $(\theta, \mathbf{q}_\theta)$  on the boundary is updated according to the rule

$$(\theta, \mathbf{q}_\theta) \leftarrow \begin{cases} (\theta, \mathbf{q}_\theta + 1) & \text{if } \mathbf{f}[\theta, \mathbf{q}_\theta + 1] > \mathbf{b}[\theta, \mathbf{q}_\theta + 1] \\ (\theta, \mathbf{q}_\theta - 1) & \text{if } \mathbf{f}[\theta, \mathbf{q}_\theta - 1] < \mathbf{b}[\theta, \mathbf{q}_\theta - 1] \end{cases} \quad (3)$$

where  $\mathbf{f}[\theta, r]$  and  $\mathbf{b}[\theta, r]$  are the histogram values for the pixels in the brackets. Also, the histograms can be updated along with the local changes. Maximization is repeated until convergence, i.e. until the change of  $\chi^2$  falls below a threshold. An example of such a boundary cost map is given by Fig. 1e.

## 2.5 Head Shape Prior

In order to encourage a plausible cutout for images that are difficult to segment, a linear shape model of the head is obtained from a set of manually segmented images. To build the model, the segmentation map of each image is first projected into the polar reference frame. Denoting polar coordinates as  $(\theta, r)$ , the boundary of each head  $(\theta_1, r_1), (\theta_2, r_2), \dots, (\theta_R, r_A)$  in a discrete raster of angles and radii is determined.

Assuming that the angular components  $\theta_i$  are strictly and regularly increasing — i.e.  $\theta_i = \frac{2\pi}{R}i$  where  $R$  is the angular resolution of the polar reference frame —, the boundary of sample image  $j$  can be represented as a single vector of radii  $\mathbf{q}_j = [r_1 \dots r_A]^T$ , which has the same length for all images. This entails a simplification of the boundary shape, which is however tolerable for head shapes. To cover more variations and increase robustness with respect to the precision of the eye detection, rotated and scaled copies of the training examples are included in the training set, yielding a total of  $N$  boundaries. Note that in polar coordinates scaling is achieved by simply adding a constant to the angular components  $r_i$ , while rotation amounts to a circular shift of elements in the radius vector.

The relative frequency of a pixel being a boundary point can finally be determined over all boundaries  $\mathbf{q}_j$  in the training set, and interpreted as a probability. The resulting head contour prior is illustrated in Fig. 1d.

## 2.6 Graph-based Segmentation

Since the segmentation process starts with models initialized from only few foreground and background samples as defined by the initial trimap, it will be performed as an iterative optimization process that successively refines the head boundary. After each intermediate segmentation step, the resulting binary mask is again expanded into a trimap, using a sequence of morphological operations. The models are then re-trained, using the foreground and background samples defined by the new trimap.

Treating the image as a graph enables us to apply algorithms from graph theory to our segmentation problem.

**Graph Construction in Polar Space** Segmentation is performed in the polar reference frame using graph search algorithms. For this purpose, the input image, as well as the cost and prior maps are interpreted as a 2D-grid like graph structure. This provides a natural representation of pixels as vertices, and their spatial neighborhood relation as edges of the graph. In our system, the edges between vertices are defined by the 8-connected neighborhood relation of the pixels, denoted as  $N$ .

The use of the polar frame provides some interesting properties for our segmentation problem. As can be seen from Fig. 1c, regions inside and outside of the head will always be located on the top and the bottom of the polar frame, respectively. The goal of the segmentation algorithm will therefore be to find a

roughly horizontal path that is optimal with respect to the regional and boundary costs defined above. As will be discussed below, this allows us to use graph search algorithms that are not truly agnostic to the shape of the path.

In polar space, specific regions can easily be enforced as being fore- or background, by limiting the search to a narrow horizontal band in the middle of the polar image. Note that it is therefore not necessary to create graph vertices for all pixels. Instead, only vertices for pixels marked by the trimap as safe foreground/background, and located at the border to the undefined region need to be constructed.

In order to yield a closed boundary in Cartesian space, the segmentation path must start and end on about the same row in the polar image, i.e.  $|r_1 - r_A| \leq 1$ . Depending on the graph search algorithm being used, this can be ensured by adding edges between the vertices on the left and right side of the 2D-grid graph in polar space, effectively yielding a torus. For algorithms that do not allow for cyclic graphs, such as the Dijkstra shortest-path search, a straightforward solution is to select the global optimum path from an iterated search over all rows in the search band, defining  $(r_i, \theta_1)$  as the source node and  $(r_{i-1}, \theta_N), \dots, (r_{i+1}, \theta_N)$  as the target nodes.

**Iterative Graph Search** Two different graph search algorithms are used in our system, namely Dijkstra shortest-path search [6] and the Kolmogorow max-flow/min-cut algorithm [3]. This is done for practical reasons: in the very first iteration of the segmentation, the color models learned from the samples given by the initial elliptical trimap will capture only a small subset of color shades from the face and head. When emphasizing on the regional term of the energy function, graphcut optimization will naturally exclude pixels not fitting the color model very well.

One possible solution would be to make the color model more tolerable, e.g. to train the GMM with less components. However, this would also reduce the distinction between the fore- and background model and may therefore introduce artifacts in images difficult to segment.

A second approach would be to reduce the influence of the regional term in graphcut optimization, thus allowing for more color variations in the foreground. However, since this likewise increases the influence of the boundary term, it will tend to produce groups of disconnected pixel “islands” or holes in regions with strong gradients, such as in the hair or in areas with hard shadows. Again, this would exclude significant parts of the foreground and will not help to improve the color models in subsequent iteration steps.

In contrast to graphcuts, the Dijkstra shortest-path algorithm will find only a single continuous path, i.e. it can not produce islands of pixels or holes in the segmentation. The Dijkstra algorithm, however, does not consider vertices at all but merely optimizes for accumulated edge weights. This makes it much more difficult to incorporate a local smoothness constraint for the path, or to setup a cost function that is truly agnostic to the spatial length and overall shape of the

path. The polar representation comes in very handy in this case, as enforcing a roughly elliptical shaped path is much easier in polar than in Cartesian space.

To exploit the strength of both algorithm, Dijkstra shortest-path search is chosen for the first iteration of the segmentation process, while graphcuts are used to iteratively refine the boundary.

The Dijkstra algorithm operates on edge weights only, as already noted. Given the edge that connects vertex  $p$  and  $q$  in a neighborhood  $N$  of the graph, the weight is derived from a linear combination of the head contour prior at this intermediate pixel position, the magnitude of the image gradient, and the boundary cost defined in Section 2.4. Denoting this linear combination as  $z_{\{p,q\}}$ , the edge weight for the Dijkstra algorithm is computed as  $-\log(z_{\{p,q\}})$ .

For a detailed explanation of the graph construction process in graphcut image segmentation, we refer to [1]. In short, pixels marked by the trimap as safe foreground/background and located at the border to the undefined region need to be constructed and linked to special foreground/background terminal nodes with maximum link capacity as the edge weight. Vertices corresponding to pixels in the undefined region of the trimap must be connected with both terminal nodes, and the weight will be determined by the regional term of our energy function. In the following, we focus on the construction of the energy function.

Let  $\mathbf{L} = \{\mathbf{L}_1, \dots, \mathbf{L}_{A \times R}\}$  be a one-dimensional vector, assigning a binary label  $\mathbf{L}_p \in \{\text{'fg'}, \text{'bg'}\}$  to each pixel (vertex)  $p \in P$  of the polar image. The solution to the segmentation problem is a labeling that minimizes the energy

$$E(\mathbf{L}) = \sum_{p \in P} R_p(\mathbf{L}_p) + \lambda \sum_{\{p,q\} \in N} B_{\{p,q\}}, \quad (4)$$

with parameter  $\lambda$  controlling the relative importance of the regional and boundary term  $R_p(\mathbf{L}_p)$  and  $B_{\{p,q\}}$ , respectively.

The regional term  $R_p(\mathbf{L}_p) = -\log p(\mathbf{c}_{\theta,r} | \text{'bg'})$  is assigned to edges connecting vertex  $p$  with the *foreground* terminal node, and  $R_p(\mathbf{L}_p) = -\log p(\mathbf{c}_{\theta,r} | \text{'fg'})$  is assigned to edges connecting vertex  $p$  and the *background* terminal node.

The boundary term  $B_{\{p,q\}}$  determines the weight of an edge connecting two vertices  $p$  and  $q$  in neighborhood  $N$ . It is again derived from the linear combination of costs  $z_{\{p,q\}}$  described above and computes as

$$B_{\{p,q\}} = \exp\left(-\frac{z_{\{p,q\}}^2}{2\sigma^2}\right) \frac{1}{\text{dist}(p,q)}. \quad (5)$$

### 3 Results

The system has been tested with several hundred images showing a single person in front of arbitrary unknown background. The images cover large variations in terms of image resolution, foreground and background complexity, as well as illumination conditions. Figure 3 shows some results produced with our system. Note that no user interaction at all was required to achieve those results.

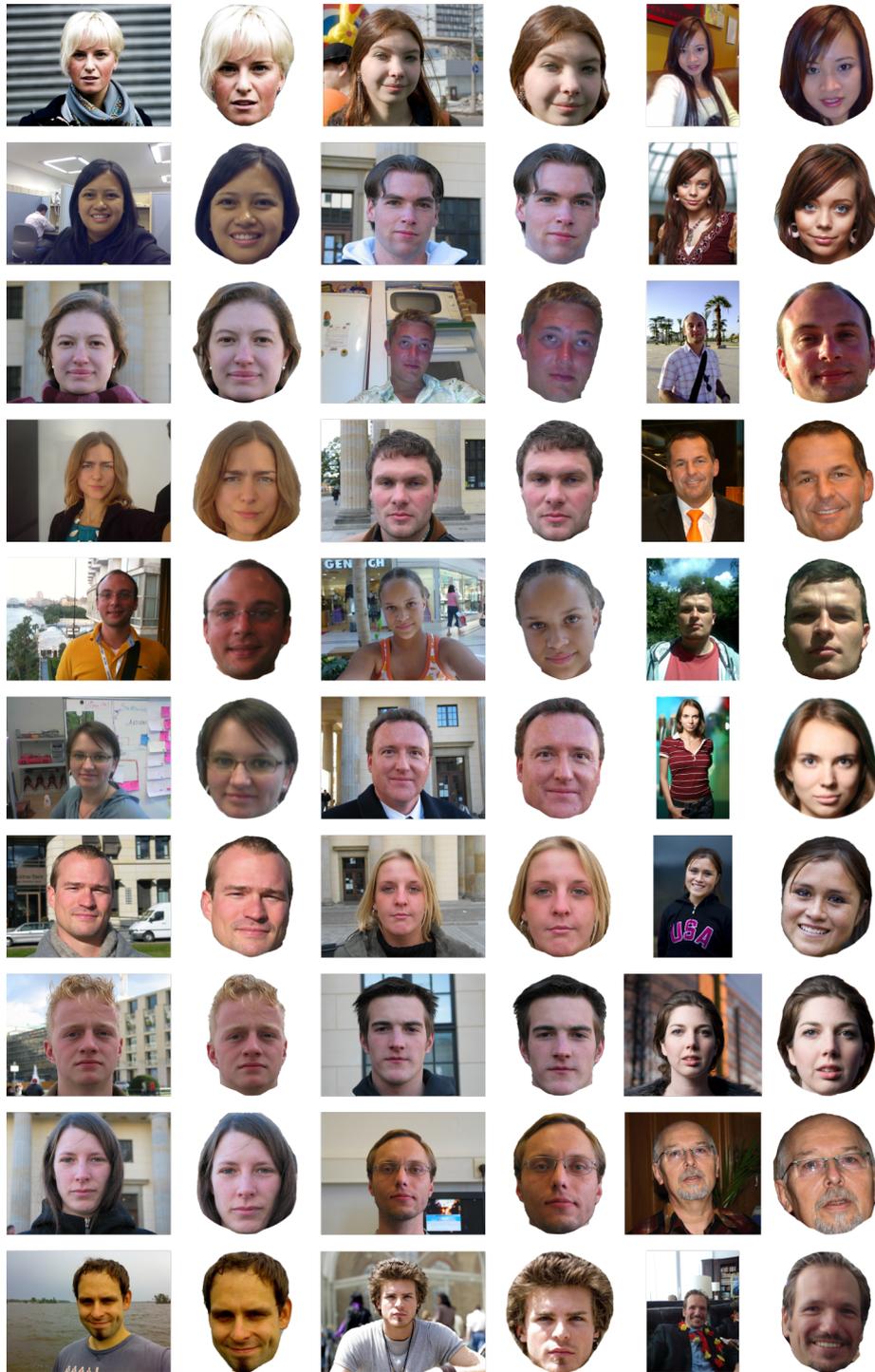


Fig. 3: Results produced by the automatic segmentation system.

Examples from a real-world use case of the system are given in Figure 4. In this application, the user can upload a portrait photo to a website to automatically create a personalized e-card or movie clip. While the automatic segmentation of the head from the input image was achieved using our system, the postprocessing and animation of the cutout were not within the scope of our work.



Fig. 4: Examples from a real-world use case of the system: the user create a personalized movie clip or e-card by uploading a portrait photo to a website. The images show a personalized e-card created as part of an election campaign (left), and a screenshot from an interactive online version of a TV series (right).

The segmentation shows to be fairly robust, however it strongly depends on the person truly facing frontal to the camera, and a good initialization in terms of the eye detector providing a precise localization of the eyes. For all examples given in Fig. 5, the detector returned the location of the brows instead of the eyes, resulting in cut-off chins.



Fig. 5: Example of failure cases due to an imprecise eye detection.

Furthermore, the treatment of hair remains difficult, due to the large amount of possible variations in shape and color. As can be seen on the left of Fig. 6, the distinction between hair and background can easily fail, particularly in front of highly structured and similarly colored backgrounds. With increasing complexity of the hair and decreasing separability from the background, it also shows quite difficult to capture long hair and fine hair strands, while still retaining an overall smooth segmentation boundary. Since we prefer visually smooth cutouts in our system, this may result in portions of the hair being removed, as shown in the center and on the right of Fig. 6.

For head images with low foreground/background separation, the use of the polar coordinate system provides an additional benefit. In such cases, the segmentation path in polar space will tend towards a straight line, yielding a rounder, visually smoother cutout in Cartesian space.



Fig. 6: Example of wrong or incomplete segmentation due to complex background (left) or hair structure (center and right).

### 3.1 Conclusion

We presented a system for the fully automated segmentation of frontal human head portraits from arbitrary unknown background. The robustness of the system has been demonstrated with images that cover a large variety of head shapes, backgrounds and illumination conditions.

Our approach is currently restricted to frontal views, primarily due to the construction of the shape model and the use of eye detection. In principle, a shape model could be set up for other views as well. Also, the result depends on eye detection which could, however, be replaced by user interaction if necessary. As many other algorithms that rely on color models, our method requires the image to be principally color-separable. However, the shape model and the non-local boundary measure have a stabilizing effect if color similarities between foreground and background are local. The use of polar coordinates additionally helps to produce visually more appealing, rounder cutouts in such difficult cases.

Regarding future work, we want to further improve the shape model and the way it is incorporated into the energy function. Also, segmentation is currently treated as a binary problem. Incorporating alpha matting to create a smooth transition between foreground and background should greatly improve the visual quality of the results; see, for example, the Bayesian method proposed in [5].

### References

1. Boykov, Y., Jolly, M.P.: Interactive Graph Cuts For Optimal Boundary & Region Segmentation Of Objects In N-D Images. In: Proc. IEEE International Conference on Computer Vision (ICCV) (2001)
2. Boykov, Y., Funka-Lea, G.: Graph Cuts and Efficient N-D Image Segmentation. *International Journal of Computer Vision* 70(2), 109–131 (2006)
3. Boykov, Y., Kolmogorov, V.: An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *Proc. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)* 26(9), 1124–1137 (2004)

4. Chang, H., Yang, Q., Parvin, B.: A Bayesian approach for image segmentation with shape priors. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2008)
5. Chuang, Y.Y., Curless, B., Salesin, D.H., Szeliski, R.: A Bayesian Approach to Digital Matting. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2001)
6. Dijkstra, E.W.: A note on two problems in connexion with graphs. *Numerische Mathematik* 1(1), 269–271 (1959)
7. Freedman, D., Zhang, T.: Interactive graph cut based segmentation with shape priors. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2005)
8. Kolmogorov, V., Zabih, R.: What Energy Functions Can Be Minimized via Graph Cuts? *IEEE Transactions On Pattern Analysis And Machine Intelligence (PAMI)* 26, 147–159 (2004)
9. Malcolm, J., Rathi, Y., Tannenbaum, A.: Graph Cut Segmentation With Nonlinear Shape Priors. In: Proc. IEEE International Conference on Image Processing (ICIP) (2007)
10. Rihan, J., Kohli, P., Torr, P.: OBJCUT for Face Detection. In: *Lecture Notes in Computer Science*, pp. 576–584. Springer (2006)
11. Rother, C., Kolmogorov, V., Blake, A.: Grabcut: Interactive Foreground Extraction Using Iterated Graph Cuts. *ACM Transactions on Graphics* 23, 309 – 314 (2004)
12. Schneider, D., Prestele, B., Eisert, P.: Precise head segmentation on arbitrary backgrounds. In: Proc. IEEE International Conference on Image Processing (ICIP). pp. 2381 –2384 (Nov 2009)
13. Viola, P., Jones, M.: Rapid Object Detection using a Boosted Cascade of Simple Features. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2001)
14. Vu, N., Manjunath, B.: Shape prior segmentation of multiple objects with graph cuts. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2008)