SYSTEM FOR THE AUTOMATED SEGMENTATION OF HEADS FROM ARBITRARY BACKGROUND

Benjamin Prestele[*] *David C. Schneider*[†] *Peter Eisert*[‡]

Fraunhofer HHI, Berlin, Germany [*] [†] [‡]
Humboldt Universität zu Berlin, Germany [†] [‡]

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, as the system is initialized using a standard eye detector. Using this semantic information, the head region is projected into a normalized polar reference frame. Regional and boundary models are learned from the image data to setup an energy function for segmentation. 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.

Index Terms— Object segmentation, Optimization, Graphcuts

1. INTRODUCTION AND RELATED WORK

Image-based object segmentation provides the basis for a large variety of methods and applications, including 3D reconstruction, semantic image retrieval, and 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 object segmentation is closely related to the problem of image understanding, robust segmentation techniques typically rely on additional constraints, such as hints derived from interactive user input, or by restraining the type of input data to a very specific class of objects and employing a predefined model that 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.

With the focus on research in recent years, particularly the rediscovery of graphcut optimization for segmentation tasks has drawn much attention by the community and has shown to be a very efficient technique for this purpose [1, 2, 3]. 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 [4]. 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 [2], 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 [3, 5, 6, 7]. For the special case of faces, the authors of [8] 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 as shape priors, such as ellipses, 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 the segmentation cost function, a more global boundary detection scheme is used, 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 an approximately 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.

Examples from a real-world use case of the system are given in Figure 1. In this application, the user can upload a portrait photo to a website, using e.g. his camera-equipped mobile phone, to automatically create a personalized e-card or movie clip with the user's head augmented into the scene. The particular challenge for the system is to cutout the unknown user's head – including the face and hair, but omitting the neck – from an unknown photo with arbitrary background. No user intervention is required, though the system could easily be extended to allow for a manual refinement stage after the automatic cutout is complete – e.g. using the techniques described in [9]. While the automatic segmentation in these examples was achieved using our system, the postprocessing and animation of the cutout were not within the scope of our work.

Fig. 1: Examples from real-world use cases of the system for the creation of personalized e-cards (left) and movie clips (right), using the cutout head from a user-provided photo.

[*]E-mail: benjamin.prestele@hhi.fraunhofer.de
[†]E-mail: david.schneider@hhi.fraunhofer.de
[‡]E-mail: peter.eisert@hhi.fraunhofer.de

2. AUTOMATIC SEGMENTATION

The segmentation process proposed by our system comprises the following steps, which require no manual user interaction at all:

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.

2.1. System Initialization and Polar Reference Frame

Similar to [8], we use a standard eye detector, such as [10], 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 detailed below. 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. 2b), 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 by the gray region in Fig. 2b.

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 (θ, 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 2c 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 approximately 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 2.5.

2.2. Head Shape Prior

In order to encourage a plausible cutout for images 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 (θ, r), the boundary of each head $(\theta_1, r_1), \ldots, (\theta_A, r_A)$ in a discrete raster of angles and radii is determined. Assuming that the angular components θ_i are strictly and regularly increasing, i.e. $\theta_i = \frac{2\pi}{A} i$, the head boundary in sample image j can be represented as a single vector of radii $\mathbf{q}_j = [r_1 \ldots r_A]^T$, 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. 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 shown in Fig. 2d.

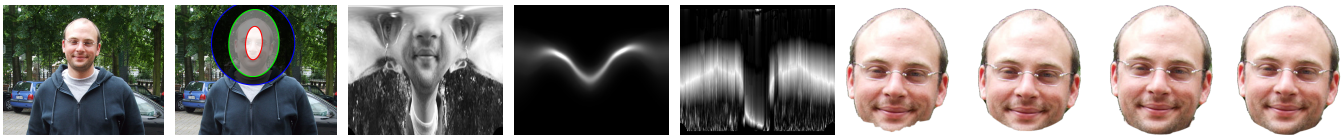
2.3. Color Model

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 and morphological expansion.

After evaluating several parametric and non-parametric model types empirically on a set of 40 highly diverse images, we chose Gaussian Mixture Models (GMMs) over L*a*b color space as the color model, since they showed reasonable small variance for the evaluated samples, and also allow for a direct probabilistic treatment of color. Standard Expectation Maximization (EM) is used to learn a foreground and a background GMM with typically 7 components each. For this training step, only corresponding pixels labeled by the trimap as either certain fore- or background are used. Denoting $\mathbf{c}_{\theta,\mathrm{r}}$ as the color of a pixel at (θ, r) in the image, and $\mathsf{L}_{\theta,\mathrm{r}} \in \{'\mathrm{fg}', '\mathrm{bg}'\}$ as its label, the GMMs give us conditional probabilities $p(\mathbf{c}_{\theta,\mathrm{r}}|\mathsf{L}_{\theta,\mathrm{r}})$ for both possible values of $\mathsf{L}_{\theta,\mathrm{r}}$. This probability will later be used to derive the regional term of our segmentation energy function.

2.4. Boundary Model

In most approaches, the boundary term of the segmentation energy function is directly derived from the local gradient magnitude of adjacent pixels. This has the disadvantage that 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 use additionally 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 optimal sense. Assuming that the trained foreground color model is sufficiently distinct, the optimum segmentation boundary should, with respect to the color model, therefore 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 θ 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 $\mathbf{q} = [r_1 \ldots r_A]^T$, as described in section 2.2. For a pixel at



(a) (b) (c) (d) (e) (f) (g)

Fig. 2: (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) Iterative refinement of the segmentation boundary; (g) Final cutout.

location (θ, r) this implies the labeling

$$\mathsf{L}_{\theta,r|\mathbf{q}} = \begin{cases} '\mathrm{fg}' & \text{if } r \leq \mathbf{q}_\theta \\ '\mathrm{bg}' & \text{otherwise} \end{cases}. \tag{1}$$

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

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

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 χ^2. Maximization is repeated until convergence, i.e. until the change of χ^2 falls below a threshold. An example of such a boundary cost map is given by Fig. 2e.

2.5. Iterative Graph-based Segmentation

The graph-based segmentation is performed on the polar representations of the input image, cost maps and prior maps. Pixels in the polar frame are represented as graph-vertices, and their 8-connected neighborhood relations, denoted as N, define the graph-edges. As can be seen from Fig. 2c, the goal of the segmentation will be to find a roughly horizontal path that is optimal with respect to the regional and boundary costs defined below. Please note that 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. $|r_1 - r_A| \leq 1$. 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, a straightforward solution is to select the global optimum path from an iterated search over all rows $i \in [2, R-1]$, defining (θ_1, r_i) as the source node, and $(\theta_A, r_{i-1}), \ldots, (\theta_A, r_{i+1})$ as the target nodes.

The segmentation starts with models initialized from only few fore- and background samples, as defined by the initial trimap, and performs an iterative optimization to successively refine 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 fore- and background samples defined by the new trimap.

Two different graph search algorithms are used in our system, namely Dijkstra shortest-path search [11] and the Kolmogorow max-flow/min-cut algorithm [12]. 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. In the graphcut optimization framework, this may introduce segmentation artifacts in regions with strong gradients, such as disconnected pixel "islands" or holes in the hair region or in areas with hard shadows. Training more tolerable GMMs or balancing the boundary and regional terms to allow for more variation showed to be too unreliable for our automated system.

As an alternative optimization method, the Dijkstra shortest-path algorithm is not the first choice for graph-based segmentation tasks, as it merely optimizes for accumulated edge weights and does not consider graph vertices. This makes it difficult to incorporate e.g. 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 easer in polar than in Cartesian space. Most importantly however, the Dijkstra algorithm optimizes globally for a single continuous path, i.e. it cannot produce holes or pixel islands in the segmentation, as it is the case with graphcuts. Dijkstra shortest-path search is therefore chosen for the very first iteration of the segmentation process, while graphcuts are used to iteratively refine the boundary until convergence (see Fig. 2f).

The Dijkstra algorithm operates on edge weights only, as already noted. Given the edge that connects vertex s and t 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 $z_{\{s,t\}}$, the edge weights for the Dijkstra algorithm are computed as $-log(z_{\{s,t\}})$.

For a detailed explanation of the graph construction process in graphcut image segmentation, we refer to [1]. In short, vertices corresponding to pixels marked by the trimap as safe fore- or background and located at the border to the undefined region are linked to special fore-/background terminal nodes with maximum link capacity as the edge weight. Vertices representing the undefined region of the trimap must be connected with both terminal nodes, and the edge weights are given by the regional term of our energy function. The boundary term finally determines the weight of edges representing the neighborhood relation N in the undefined region. In the following, we focus on the construction of the energy function.

Let $\mathsf{L} = \{\mathsf{L}_1, \ldots, \mathsf{L}_{A \times R}\}$ be a one-dimensional vector, assigning a binary label $\mathsf{L}_s \in \{'\mathrm{fg}', '\mathrm{bg}'\}$ to each pixel (vertex) $s \in S$ at location (θ, r) of the polar image. The solution to the segmentation problem is a labeling that minimizes the energy

$$E(\mathsf{L}) = \sum_{s \in S} R_s(\mathsf{L}_s) + \lambda \sum_{\{s,t\} \in N} B_{\{s,t\}}, \tag{3}$$

with parameter λ controlling the relative importance of the regional and boundary term $R_s(\mathsf{L}_s)$ and $B_{\{s,t\}}$, respectively.

The regional term $R_s(\mathsf{L}_s) = -\log\, p(\mathbf{c}_{\theta,\mathrm{r}}|'\mathrm{bg}')$ is assigned to edges connecting vertex s with the *forgeground* terminal node, and $R_s(\mathsf{L}_s) = -\log\, p(\mathbf{c}_{\theta,\mathrm{r}}|'\mathrm{fg}')$ is assigned to edges connecting vertex s and the *background* terminal node.

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

$$B_{\{s,t\}} = \exp\left(-\frac{z^2_{\{s,t\}}}{2\sigma^2}\right)\frac{1}{dist(s,t)}. \qquad (4)$$

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 exemplary results produced with the system. Note that no user interaction at all was required to achieve those results.

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

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 a precise localization by the eye detector (see Fig. 4a for a typical failure case). Furthermore, the treatment of hair in front of unknown background remains difficult, due to the large amount of possible variations in shape and color. As can be seen in Fig. 4b, the separation can easily fail especially in front of highly structured and similarly colored backgrounds. It also showed 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 Fig. 4c. 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.

4. CONCLUSION

We presented a system for the fully automated segmentation of frontal human head portraits from arbitrary unknown background.

(a) (b) (c)

Fig. 4: Examples of poor segmentation due to: (a) imprecise eye detection; (b) complex background; (c) complex hair shape.

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 the precision of eye detection, which could however be corrected by user interaction, if necessary. Like 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 used have a stabilizing effect when foreground and background colors are locally very similar. The use of polar coordinates additionally helps to produce visually more appealing, rounder cutouts in such difficult cases.

5. REFERENCES

[1] Y. Boykov and M.-P. Jolly, "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] Vladimir Kolmogorov and Ramin Zabih, "What Energy Functions Can Be Minimized via Graph Cuts?," *IEEE Transactions On Pattern Analysis And Machine Intelligence (PAMI)*, vol. 26, pp. 147–159, 2004.

[3] James Malcolm, Yogesh Rathi, and Allen Tannenbaum, "Graph Cut Segmentation With Nonlinear Shape Priors," in *Proc. IEEE International Conference on Image Processing (ICIP)*, 2007.

[4] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake, "Grabcut: Interactive Foreground Extraction Using Iterated Graph Cuts," *ACM Transactions on Graphics*, vol. 23, pp. 309 – 314, 2004.

[5] Hang Chang, Qing Yang, and B. Parvin, "A Bayesian approach for image segmentation with shape priors," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[6] D. Freedman and Tao Zhang, "Interactive graph cut based segmentation with shape priors," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[7] Nhat Vu and B.S. Manjunath, "Shape prior segmentation of multiple objects with graph cuts," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[8] Jonathan Rihan, Pushmeet Kohli, and Philip Torr, "OBJCUT for Face Detection," in *Lecture Notes in Computer Science*, pp. 576–584. Springer, 2006.

[9] D.C. Schneider, B. Prestele, and P. Eisert, "Precise head segmentation on arbitrary backgrounds," in *Proc. IEEE International Conference on Image Processing (ICIP)*, Nov. 2009, pp. 2381 –2384.

[10] Paul Viola and Michael Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2001.

[11] Edsger Wybe Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, no. 1, pp. 269–271, 1959.

[12] Yuri Boykov and Vladimir Kolmogorov, "An experimental comparison of min-cut/max- flow algorithms for energy minimization in vision," *Proc. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 26, no. 9, pp. 1124–1137, 2004.