ROBUST SEGMENTATION OF RELEVANT REGIONS IN LOW DEPTH OF FIELD IMAGES

Franz Graf, Hans-Peter Kriegel, Michael Weiler

Ludwig-Maximilians-Universität München
Oettingenstr. 67, 80538 Munich, Germany

ABSTRACT

Low depth of field (DOF) is an important technique to emphasize the object of interest (OOI) within an image. When viewing a low depth of field image, the viewer implicitly segments the image into region of interest and non regions of interest which has major impact on the perception of the image. Thus, robust algorithms for the detection of the OOI in low DOF images provide valuable information for subsequent image processing and image retrieval. In this paper we propose a robust and parameterless algorithm for the fully automatic segmentation of low depth of field images. We compare our method with three similar methods and show the superior robustness even though our algorithm does not require any parameters to be set by hand. The experiments are conducted on a real world data set with high and low depth of field images.

Index Terms— Image Segmentation, Low Depth of Field, Object of Interest

1. INTRODUCTION

In photography low depth of field (DOF) is an important technique to emphasize the object of interest (OOI) within an image. Low DOF images are well known from portrait, sports or macro photography where only a part of the image should attract most of the users' attention. The OOI is thereby displayed sharp while other areas like the background appears blurred, so that the viewer automatically focuses on the sharp areas of the image. When viewing a low depth of field image, the viewer implicitly segments the image into region of interest and regions of less interest (usually background). As the implicit segmentation has major impact on the perception of the image, this information is valuable feature for the subsequent image processing chain like an adaptive image compression [1] or image retrieval aspects as the similarity of images can be considerably influenced by the image's DOF.

In this paper we propose a novel robust and parameterless algorithm for the fully automatic segmentation of low depth of field images. The segmentation of low DOF images has gained some interest in the research community in past years. In [2, 3] first approaches to segment low DOF images were presented. In [4, 5] high frequency wavelets are used to determine the segmentation which have the drawback of being not too robust. In [6], localized blind deconvolution is used to determine the OOI, yet the authors do not propose a pure image segmentation algorithm. The works proposed in [7, 8, 9] are consecutive works for segmentation of image sequences which address a similar topic. However, the method is not fully applicable to single image segmentation.

We compare our algorithm with the works of [10] which inspired our algorithm, with [11] where single frames of videos are segmented and to [12], where a fuzzy segmentation approach was proposed. The test image dataset consists of a set of various photos and comprises several categories from high to low depth of field images.

This research has been supported in part by the THESEUS program in the CTC and Medico projects. They are funded by the German Federal Ministry of Economics and Technology under the grant number 01MQ07020. The responsibility for this publication lies with the authors.

2. ALGORITHM

The proposed algorithm consists of five stages which will be discussed in the following sections.

2.1. Deviation Scoring

First, sharp pixels are identified. Let I be the set of pixels $I = \{\mathrm{p(x,y)} | 1 \leq \mathrm{x} \leq \mathrm{width}, 1 \leq \mathrm{y} \leq \mathrm{height}\}$ of the image. For each pixel $p\,(x,y)$ the mean color from its r-neighbourhood $\eta^r_{I(x,y)} = \{p\,(x',y') \mid |x'-x| \leq r \wedge |y'-y| \leq r\}$ is calculated. As $p(x,y)$ is represented in the $L^*a^*b^*$ color space, the mean neighbourhood color of $p(x,y)$ in the L^*-band is determined by

$$L_{\eta^r_{I(x,y)}} = \frac{\sum\limits_{p \in \eta^r_{I(x,y)}} \left(L^*_p\right)}{\left|\eta^r_{I(x,y)}\right|} \tag{1}$$

. The values for the a^*- and b^*-band are denoted by $a_{\eta^r_{I(x,y)}}$ and $b_{\eta^r_{I(x,y)}}$ respectively, so that the mean neighbourhood color $Lab_{\eta^r_{I(x,y)}}$ is defined by $\left(L_{\eta^r_{I(x,y)}}, a_{\eta^r_{I(x,y)}}, b_{\eta^r_{I(x,y)}}\right)$. For each $p\,(x,y)$, the neighbour difference is defined by

$$\Delta\eta^r_{I(x,y)} = min\left(255 \cdot \frac{\Delta E^*\left(Lab_{\eta^r_{I(x,y)}}, p\right)}{\Delta E^{max}}, 255\right) \quad (2)$$

with ΔE^{max} being the maximum possible distance in the $L^*a^*b^*$ space and $\Delta E^*(u,v)$ being the euclidean distance of the color values u, v in the $L^*a^*b^*$ space. To compute the score $\mu \in [0,255]$ of $p(x,y)$ we first convolve I with a gaussian kernel with radius σ and denote the result by I_σ. The score $\mu(x,y)$ for $p(x,y)$ is calculated by $\mu(x,y) = \left(\Delta\eta^r_{I(x,y)} - \Delta\eta^r_{I_\sigma(x,y)}\right)^2$ and identifies pixels having a high probability of belonging to the focused region, which is the case, if μ exceeds the threshold $\Theta_{score} \in [0, 255]$. For each pixel $p(x,y) \in I$, $I_{score}(x,y)$ is defined as follows:

$$I_{score}(x,y) = \begin{cases} 0 & \mu(x,y) < \Theta_{score} \\ \mu(x,y) & else \end{cases} \quad (3)$$

Finally I is scaled by half to improve the processing speed of the subsequent steps without major impact to accuracy.

2.2. Score Clustering

In this stage, clusters are generated from all $p \in I_{score}$ by applying DBSCAN [13], which is based on the two parameters ε and $minPts$. To provide highest flexibility with respect to the different occurrences of the focused area, we do not apply absolute values, but compute them relatively to the size of the image and its score distribution. Thus, ε is calculated by $\varepsilon = \sqrt{|I|} \cdot \Theta_\varepsilon$, with $|I|$ denoting the total amount of pixels in I and $\Theta_\varepsilon \in [0,1]$. $minPts$ is defined as in Equation 4.

$$minPts = \left\lfloor \frac{|\{(x,y) \mid I_{score}(x,y) > 0\}|}{|I|} (\varepsilon + 1)^2 \right\rfloor \quad (4)$$

The weight w_p for a point p is then defined as $w_p = min\left\{\frac{I_{score}(x,y)}{\Theta_{dbscan}}, 1\right\}$, with the threshold $\Theta_{dbscan} \in [0,255]$. The result of the clustering is a cluster set $C = \{c_1, \ldots, c_n\}$, with each $c_i \in C$ representing a subset of pixels $p(x,y) \in I$. Due to our assumption that small isolated sharp areas are treated as noise, we define the relevant score cluster set $\hat{C} = \left\{c \in C \mid |c_i| \geq \frac{max_C}{2}\right\} \subseteq C$ with $max_C = max\{|c_1|, \ldots, |c_n|\}$ being the amount of pixels of the largest cluster.

2.3. Mask Approximation

The relevant cluster set $\hat{C}$, is a good reference point of the OOI's location and distribution. In general however, there exists no contiguous area, but individual regions of interest. This stage of the algorithm connects all clusters $c \in \hat{C}$ to a contiguous area which represents an approximate binary mask of the OOI. Therefore, we first generate the convex hull for all points in the ε-neighbourhood $N_{Eps}(p)$ of each core point p of the cluster set. Let $K = \{k_1, \ldots, k_j\}$ be the set of all core points from the score clusters in $\hat{C}$ and let $convex(P)$ be the convex hull from a point set P. Then we can define the set of convex hull polygons by $H = \{convex(N_{eps}(k_1)), \ldots, convex(N_{eps}(k_j))\}$ which is used to generate a contiguous area. Therefore each $p(u,v) \in I$ is checked, if it is located within one of the convex hull polygons of H. If that is the case, we mark this pixel with 1, otherwise with 0. The binary approximation mask I_{app} is then given by

$$I_{app}(x,y) = \begin{cases} 1 & if\ \exists H_i : p(x,y) \in H_i \\ 0 & otherwise \end{cases}.$$

Afterwards we apply the morphological filter operations *closing* and *dilation by reconstruction* to I_{app} for smoothing and closing small holes. The filters are based on the operations *dilation* $\delta_H(I)$ and *erosion* $\varepsilon_H(I)$, where $H(i,j) \in \{0,1\}$ denotes the *structuring element.* $\varepsilon_H(I)$ can be extended to a *basic geodesic erosion* $\varepsilon^1(I, I')$ of size 1, so that $\varepsilon^1(I, I')(u,v) = max\{\varepsilon_H(I)(u,v), I'(u,v)\}$. The geodesic *erosion* of infinite size ε^∞, called *reconstruction by erosion* φ^{rec} is defined as $\varphi^{rec}(I, I') = \varepsilon^\infty(I, I') = \varepsilon^1 \circ \varepsilon^1 \circ \ldots \circ \varepsilon^1(I, I')$. Note that that $\varphi^{rec}(\cdot,\cdot)$ converges and achieves stability after a certian number of iterations.

In our approach, we primarily apply a *morphological closing* operation $\varphi_H(I_{app}) = \varepsilon_H(\delta_H(I_{app}))$ to the approximate mask. The dimension of the structuring element H therefore is discussed later. Afterwards we use $\varphi^{rec}(I_{app}, \delta_{H'}(I_{app}))$ to close holes in the approximate mask I_{app}. The dimension of the structuring element H' is $h \times h$, where h is calculated relatively to the total pixel count $|I|$ of image I, so that $h = \sqrt{|I|} \cdot \Theta_{rec}$, with $\Theta_{rec} \in [0,1]$. After this morphological processing, the approximate mask I_{app} covers the OOI quite well. In gerneral however, it includes boundary regions that exceed the borders of the OOI and tend to surround it with a thick border. The following two stages of our algorithm refine the mask by erasing all these surrounding border regions.

2.4. Color Segmentation

In this stage, the pixels from the approximate mask I_{app} are divided into groups, so that each group contains pixels that correspond to similar colors in I. Therefore we process each $p(u,v) \in I_{app}$ and iteratively include all its neighbours $n \in \left\{(s,t) \in \eta^1_{I_{app}(u,v)} \mid I_{app}(s,t) = I_{app}(u,v) = 1\right\}$ with similar color to p, so that $\Delta E^*(p,n) < \Theta_{dist}$. The threshold $\Theta_{dist} \in [0,100]$ is an internal parameter, which specifies the maximum distance between two color values u, v in the $L^*a^*b^*$ color space.

Therefore a method $expand(x,y,R)$ is called for each $p(x,y) \in \{(x,y) \in I_{app} \mid I_{app}(x,y) = 1\}$, which is not marked as visited. $R = \{(x,y)\}$ thereby defines

a new color region formed by the point $p_1(x, y)$. The method $expand(x, y, R)$ then proceeds as follows: For all $p_2(u, v) \in \{(s, t) \in \eta^1_{I_{app}(x,y)} \mid I_{app}(s, t) = 1\}$ with $\Delta E^*(p_1, p_2) < \Theta_{dist}$ we add p_2 to R and mark $p(u, v)$ as visited. Then $expand(u, v, R)$ is called recursively. The resulting set of regions is called R_{color}.

2.5. Region Scoring

In this step, a relevance value μ is calculated for each region $r \in R_{color}$. Low relevant regions are removed afterwards which causes an update of μ in the neighbouring regions and thus possibly trigger another deletion if the relevance of an updated region is not high enough after the according update.

The mask boundary overlap BO^R_r of a region r is a measure for the adjacence of r to the approximate mask I_{app} and is defined by $BO^R_r = |\{(u, v) \in B_r \mid \exists r' \in R : (u, v) \in r'\}|$, where B_r is the difference of r to its dilation. The mask boundary overlap MBO_r of r is then defined as $MBO_r = \frac{BO^{R_{color}}_r}{|B_r|}$. MBO_r specifies the ratio of the number of outline points located in other regions to the number of all outline points of r. The score boundary overlap $SBO_r = \frac{BO^{\hat{C}}_r}{|B_r|}$ of r is a measure for the adjacence of r to the corresponding score values μ. A large SBO_r indicates, that r has a neighbourhood with large corresponding score values μ. The mask relevance for a given region r can then finally be defined as $MR_r = SBO_r \cdot MBO_r$. Afterwards, we eliminate all regions r with a mask relevancy value which is too low. The calculation of MR_r is executed iteratively: Let MR^i_r and MBO^i_r denote the values MR_r and MBO_r of a region r during the i-th iteration. One iteration cycle computes the corresponding μ for each region r and deletes r from the approximate mask I_{app} if the condition $MBO^i_r \leq \lambda^{MBO}_i \wedge MR^i_r \leq \lambda^{MR}_i \wedge |r| \leq \lambda^{MS}$ holds, where λ^{MBO}_i and λ^{MR}_i are both calculated relatively to the iteration i, so that an increase of i results in an increase of both values. λ^{MS} defines the maximum size of a region that can be deleted if its relevancy is low. The condition $|r| \leq \lambda^{MS}$ ensures that large regions are unaffected by the refinement because they already cover a huge area of the OOI. The precise assignment of the threshold values is discussed later. Once a region r satisfies the condition X^i_r at iteration i, it will be erased from I_{app}, so that $\forall (x, y) \in r : I_{app}(x, y) = 0$. The calculation of MR^i_r continues for $i = 1, \ldots m$ and terminates as soon as there are no more regions to delete. This is the case, as soon as $MR^i_r = MR^{i-1}_r$so that $\exists m \geq 1 \mid \forall r \in R_{color} : MR^i_r = MR^{i-1}_r$.

3. EXPERIMENTAL RESULTS

A major contribution of this algorithm is that none of the parameters introduced in the previous section needs to be hand tuned for an image as all parameters are either independent of the image or determined fully automatically. An overview of the implicit parameters can be seen in Table 1. To determine the quality of a segmentation mask I we use the *spatial distortion* $d(I, I_r)$ as proposed in [10]:

$$d(I, I_r) = \left(\sum I(x, y) \otimes I_r(x, y)\right) / \sum I_r(x, y),$$

where $\otimes$ is the binary *XOR* operation and I_r being the manually generated reference mask.

Θ_{score}	10	Θ_{rec}	$\frac{1}{3}$
Θ_{ε}	0.025	Θ_{dist}	20
Θ_{dbscan}	128	λ^{MBO}_i	$0.85 - \frac{1}{3}i$
σ	1.25	λ^{MR}_i	λ^{MR}_i
H	25	λ^{MS}	$0.2 \sum_{c \in \hat{C}} \lvert c \rvert$

Table 1: Parameters used in the algorithm.

All experiments were conducted on a diverse dataset of 65 images mostly downloaded from Flickr. The images are from different categories with strong variations in the amount of depth of field, same as the fuzziness of the background. Also the selection of the images does not focus on certain sceneries, topics or coloring schemes in order to avoid overfitting to certain types of images. In our experiments, we compare the spatial distortion of the proposed algorithm with reimplementations of the works presented in [10], [11] and [12]. The parameters for all algorithms were optimized to achieve the best average spatial distortion over the complete test set.

Fig. 1 compares the performance of the reference algorithms with our proposed method. In case that an algorithm performed rather poor and produced a segmentation with $d(I, I_r)$> 2, we defined this image to be an outlier and also present the error values in brackets where the outliers were removed. It can be seen that even though the computation time of the proposed algorithm is greater than two of the three reference algorithms, it outperforms the reference algorithms in terms of spatial distortion in all cases. Also, our algorithm has a maximum spatial distortion of 1.58 over the complete test set which means that it managed to produce a reasonable segmentation in all cases whereas the reference algorithms performed rather poor in 2, 7 and 20 cases. It should also be noted that in contrast to the reference algorithms, the proposed algorithm can take advantage of larger images whereas the competitors loose accuracy with growing size of the image.

4. CONCLUSION

In this paper a new robust algorithm for the segmentation of low depth of field images is proposed which does not need to set any parameters by hand as all necesssary parameters are determined fully automatically. Experiments are conducted on set of real world low depth of field images from various

(a) Original image. (b) Proposed algorithm. (c) Result of fuzzy segmentation [12] (d) Result of morphological segmentation [10] (e) Result of video segmentation [11]

Fig. 1: Comparing the results of the different algorithms.

	proposed	[10]	[12]	[11]
min	**0.01**	0.03	0.02	0.07
max	**1.58**	>43 (2)	>8 (2)	>20 (2)
median	**0.10**	0.33	0.98	1.36
average	**0.22**	1.4 (0.5)	1.2 (1.0)	2.2 (1.3)
std.dev.	**0.26**	6.0 (0.4)	1.2 (0.5)	3.1 (0.7)
time	35s	9s	54.2s	**2.7s**
# images	**65**	65 (63)	65 (58)	65 (45)

Table 2: Spatial distortion and run time of the proposed algorithm compared to the reference algorithms. The numbers in brackets denote values with removed outliers $(d(I, I_r) > 2)$.

categories and the algorithm is compared to three refernce algorithms. The experiments show that the algorithm is more robust than the reference algorithms on all tested images and it performs well even if the depth of field is growing larger so that the background begins to show considerable contours. In Future work, we plan to improve processing speed and accuracy of the algorithm. A demo of the algorithm can be tested online[1].

5. REFERENCES

[1] S. Kavitha, S. Roomi, and N. Ramaraj, "Lossy compression through segmentation on low depth-of-field images," *Digital Signal Processing*, vol. 19, no. 1.

[2] J. Li, J.Z. Wang, RM Gray, and G. Wiederhold, "Multiresolution object-of-interest detection for images with low depth of field," in *Image Analysis and Processing, 1999. Proc. International Conference on*.

[3] D.M. Tsai and H.J. Wang, "Segmenting focused objects in complex visual images," *Pattern Recognition Letters*, vol. 19, no. 10, pp. 929–940, 1998.

[4] Z. Ye and C.C. Lu, "Unsupervised multiscale focused objects detection using hidden Markov tree," in *Proceedings of the International Conference on Computer Vision, Pattern Recognition, and Image Processing (IEEE Press, 2002)*, pp. 812–815.

[5] C.S. Won, K. Pyun, and R.M. Gray, "Automatic object segmentation in images with low depth of field," in *Image Processing. 2002. Proceedings. 2002 International Conference on*. IEEE, 2002, vol. 3, pp. 805–808.

[6] L. Kovács and T. Szirányi, "Focus area extraction by blind deconvolution for defining regions of interest," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 29, no. 6, pp. 1080–1085, 2007.

[7] C. Kim, J. Park, J. Lee, and J.N. Hwang, "Fast extraction of objects of interest from images with low depth of field," *ETRI journal*, vol. 29, no. 3, pp. 353–362, 2007.

[8] J. Park and C. Kim, "Extracting focused object from low depth-of-field image sequences," in *Proceedings of SPIE*, 2006, vol. 6077, pp. 578–585.

[9] C. Kim and J.N. Hwang, "Video object extraction for object-oriented applications," *The Journal of VLSI Signal Processing*, vol. 29, no. 1, pp. 7–21, 2001.

[10] K. Ramar N. Santhi, "Image Segmentation Using Morphological Filters and Region Merging," *Asian Journal of Information Technology*, vol. 6, no. 3.

[11] H. Li and K.N. Ngan, "Unsupervized video segmentation with low depth of field," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 12, pp. 1742–1751, 2007.

[12] K.D. Zhang, H.Q. Lu, Z.Y. Wang, Q. Zhao, and M.Y. Duan, "A Fuzzy Segmentation of Salient Region of Interest in Low Depth of Field Image," *Advances in Multimedia Modeling*, pp. 782–791.

[13] M. Ester, H.P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. KDD*, vol. 96.

[1] http://www.dbs.ifi.lmu.de/research/ICIP-ImageSegmentation/