Global Optimal Multiple Object Detection Using the Fusion of Shape and Color Information

Marek Schikora

FGAN Research Institute for Communication, Information Processing and Ergonomics (FKIE)
D-53343 Wachtberg, Germany
schikora@fgan.de

Abstract. In this work we present a novel method for detecting multiple objects of interest in one image, when the only available information about these objects are their shape and color. To solve this task we use a global optimal variational approach based on total variation. The presented energy functional can be minimized locally due its convex formulation. To improve the runtime of our algorithm we show how this approach can be scheduled in parallel.Our algorithm works fully automatically and does not need any user interaction. In experiments we show the capabilities in non-artificial images, e.g. aerial or bureau images.

1 Introduction

To detect multiple objects of interest we use the concept of image segmentation. We will segment the image plane into two regions: foreground (objects of interest) and background. In this context we will use the minimization of an energy functional in continuous space introduced in [1] and [2]. The usage of shape information for image segmentation is normally done using the level-set representations (cf. [3] [4] [5] [6]). In this representation a shape is defined as the boundary given by the zero level set of an embedding function $\phi : \mathbb{R}^d \rightarrow \mathbb{R}$:

$$C = \left\{ \boldsymbol{x} \in \mathbb{R}^d \middle| \phi(\boldsymbol{x}) = 0 \right\}. \tag{1}$$

The shape priors in this context are then defined on a space of embedding functions using the space of signed distance functions. Although this formulation has its benefits (independency of parametrization and easy handling of topological changes) there exist two well-known drawbacks: Firstly, the space of signed distance functions is not a linear space, and secondly, the resulting cost or energy functionals are generally not convex.

Recently, an alternative to the continuous level set representation has been proposed, where the segmentation of images is formulated on the basis of convex functional minimization using the concept of *Total Variation*(TV) (c.f [7], [8]). In [9] the formulation of a globally optimal color-based image segmentation using the TV norm was shown. In this paper we extend this work by combining it with shape information.

D. Cremers et al. (Eds.): EMMCVPR 2009, LNCS 5681, pp. 443–454, 2009.

2 Shape Information

In this section we briefly describe the shape prior model, introduced in [10], which will be used in the following because of its convex and continuous formulation.

For the representation of shapes we use the shape space $\mathcal{Q}$:

Definition 1. *A* **shape** *in* $\mathbb{R}^d$ *is a function*

$$q : \mathbb{R}^d \to [0, 1]\,, \tag{2}$$

which assigns to any pixel $\boldsymbol{x} \in \mathbb{R}^d$ *a probability* $q(\boldsymbol{x})$ *that* $\boldsymbol{x}$ *is part of the object. The space of all shapes will be denoted* $\mathcal{Q}$*. In our case we will only consider planar shapes, so we set* $d = 2$*.*

The benefit of this model lies in the independency of any parametrization. So the problem of shape alignment does not require the estimation of point correspondences. Furthermore the values of q can be easily interpreted in a probabilistic sense. Cremers et al. have shown in their paper [10] that the shape space $\mathcal{Q}$ is convex. This characteristic of $\mathcal{Q}$ leads to the conclusion that any convex combination of elements of the set

$$\chi = \{q_1, q_2, ..., q_N\} \tag{3}$$

is a valid shape. With this we can define statistic quantities such as mean, covariance matrices and eigenmodes of a training set χ.

Let $\chi = \{q_1, q_2, ..., q_N\}$ be a set of N training shapes; then the mean value $\mu : \mathbb{R}^2 \to [0, 1]$ of this set is defined through

$$\mu(\boldsymbol{x}) = \frac{1}{N} \sum_{i=1}^{N} q_i(\boldsymbol{x}). \tag{4}$$

This is a function that assignes to each pixel $\boldsymbol{x} \in \mathbb{R}^2$ the average of all probabilities. Using principal component analysis (PCA) we compute the eigenmodes of the shape set χ. We use only a subspace of χ spanned by the first $n \leq N$ eigenmodes $\{\psi_1, \psi_2, ..., \psi_n\}$. The size n follows from the cumulative energy content for each eigenmode. In experiments we used a threshold value of about 0.8. Figure 1 shows the normalized cumulative energy for our training set database. Now a subspace χ_n is given by:

$$\chi_n = \left\{ q_\alpha = \mu + \sum_{i=1}^{n} \alpha_i \psi_i \;\middle|\; q_\alpha(\boldsymbol{x}) \in [0, 1]\,,\ \alpha_i \in \mathbb{R} \right\}. \tag{5}$$

In [10] it was shown that χ_n is convex. Now we can generate an shape from this space as

$$q_\alpha = \mu + \boldsymbol{\alpha}^T \boldsymbol{\Psi} \tag{6}$$

With this we can describe every shape only storing the vector $\boldsymbol{\alpha} \in \mathbb{R}^n$. $\boldsymbol{\Psi}$ is a matrix containing the eigenmodes $\psi_1, \psi_2, ..., \psi_n$. Figure 2 shows some examples.

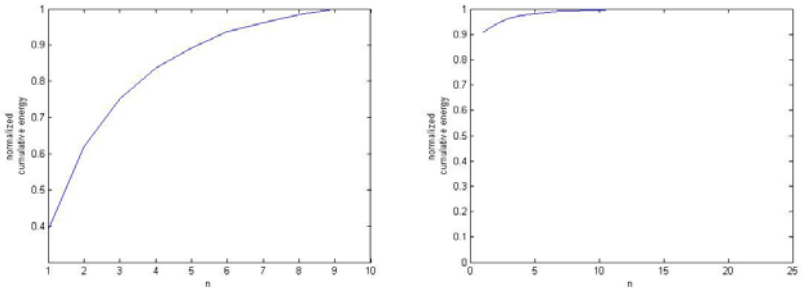


Fig. 1. Normalized cumulative energy content of eigenmodes vs. the number of eigenmodes used for the representation for a database of human hands (left figure) and for a car database segmented manually from aerial images (right).

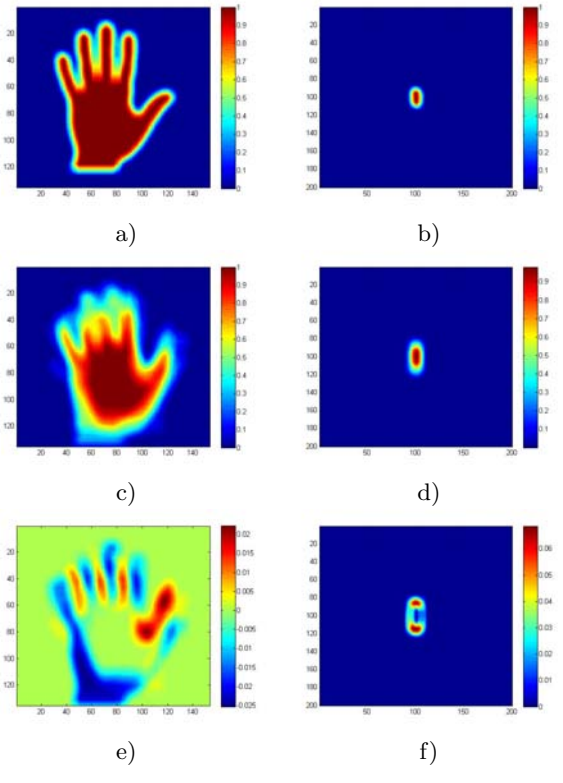


Fig. 2. Shape information: On the left side a hand database is used, on the right side we use a collection of manually-segmented cars from aerial images. a) and b) are example shapes from our database. c) and d) are the mean shapes μ from equation (4). e) and f) represent the first eigenmode ψ_1 of the database.

3 Multiple Object Detection

In this section we describe how to detect multiple object using shape and color information. First we formulate a convex energy function and show that it can be computed efficiently by parallelization. Then we describe the following steps of our algorithm.

3.1 Convex Functional

The energy function used in [9] was formulated for color-based image segmentation.We extend this approach to incorporate also shape information. The general form of a functional for a desired segmentation $u : \mathbb{R}^2 \rightarrow [0, 1]$ is

$$E(u) = E_{\text{img}}(u) + \beta \cdot E_{\text{shape}}(u). \tag{7}$$

The color based energy function is of the form

$$E_{\text{img}}(u) = \int_{\Omega} f(\boldsymbol{x})u(\boldsymbol{x}) \ \mathrm{d}\boldsymbol{x} + \gamma \int_{\Omega} |\boldsymbol{\nabla} u(\boldsymbol{x})| \ \mathrm{d}\boldsymbol{x} + \rho \int_{\Omega} \xi(u(\boldsymbol{x}))\mathrm{d}\boldsymbol{x}, \tag{8}$$

where $\Omega \subseteq \mathbb{R}^2$ denotes the image plane and $\beta, \gamma, \rho \in \mathbb{R}$ are weighting parameters.The function ξ penalizes values of u lying outside of the valid range of $[0, 1]$. f can be an arbitrary function which measures the consistency of a point $\boldsymbol{x}$ with the foreground. In our work we used the following function for f:

$$f(\boldsymbol{x}) = \Delta\left(I^{\text{HSV}}(\boldsymbol{x}), \boldsymbol{\nu}_{\text{obj}}\right) - \Delta\left(I^{\text{HSV}}(\boldsymbol{x}), \boldsymbol{\nu}_{\text{bgd}}\right). \tag{9}$$

Here, I^{HSV} is the input image I transformed into the HSV color space. The function Δ computes the squared distances of the single channels of I^{HSV} to the mean value $\boldsymbol{\nu}$ of a region.

$$\Delta\left(I^{\text{HSV}}(\boldsymbol{x}), \boldsymbol{\nu}\right) = w_{\text{H}}\left(I^{\text{H}}(\boldsymbol{x}) - \nu^{\text{H}}\right)^2 + w_{\text{S}}\left(I^{\text{S}}(\boldsymbol{x}) - \nu^{\text{S}}\right)^2 + w_{\text{V}}\left(I^{\text{V}}(\boldsymbol{x}) - \nu^{\text{V}}\right)^2 \tag{10}$$

$w_{\text{H}}, w_{\text{S}}$ and w_{V} being (normalized) weighting parameters.

The term introducing the shape information into the segmentation is $E_{\text{shape}}(u)$:

$$E_{\text{shape}}(u) = \int_{\Omega} |u(\boldsymbol{x}) - \tilde{q}_{\alpha}(\boldsymbol{x})| \, \mathrm{d}\boldsymbol{x} \tag{11}$$

with

$$\tilde{q}_{\alpha} = \sum_{k=1}^{K} \Phi_u\left(q_{\boldsymbol{\alpha}k}, \Theta_k\right). \tag{12}$$

The function Φ_u projects the shape $q_{\boldsymbol{\alpha}k}$ into the image plane of u using the transformation vector $\Theta_k = (t_x, t_y, \phi, \lambda)$ for every object k. K is the number of object in the image I. This number is estimated automatically. Details on this will be given later in this paper. The transformation vector Θ_k contains

two parameters for the translation (t_x and t_y), one for rotation (ϕ), and one for scaling (λ). With these parameters we can perform any similarity transformation of a planar shape q. $q_{\boldsymbol{\alpha}k}$ is a shape generated from our database given the vector $\boldsymbol{\alpha}_k$:

$$q_{\boldsymbol{\alpha}k} = \mu + \sum_{i=1}^{n} \boldsymbol{\alpha}_k(i) \cdot \psi_i. \tag{13}$$

Let us define the transformed version of $q_{\boldsymbol{\alpha}k}$ as:

$$q_{\boldsymbol{\alpha}_k}^{\Theta_k} = \Phi_u(q_{\boldsymbol{\alpha}k}, \Theta_k). \tag{14}$$

Since it was shown in [9] that E_{img} is a convex functional, what remains to be shown is that $E_{\text{shape}}(u)$ is also convex.

Lemma 1. *The energy functional (11) is convex.*

Proof (of lemma 1). To show that (11) is convex with respect to u, we have to show that for all $\rho \in (0,1)$ holds

$$\forall u_1, u_2 : E_{\text{shape}}\left((1-\rho)u_1 + \rho \cdot u_2\right) \leq (1-\rho)E_{\text{shape}}(u_1) + \rho \cdot E_{\text{shape}}(u_2). \tag{15}$$

So we can write

$$\begin{aligned} E_{\text{shape}}\left((1-\rho)u_1 + \rho \cdot u_2\right) &= \int_\Omega |(1-\rho)u_1 + \rho \cdot u_2 - \tilde{q}_\alpha| \, \mathrm{d}\boldsymbol{x} && (16)\\ &\leq \int_\Omega (1-\rho)\,|u_1 - \tilde{q}_\alpha| + \rho \cdot |u_2 - \tilde{q}_\alpha| \, \mathrm{d}\boldsymbol{x} && (17)\\ &= \int_\Omega (1-\rho)\,|u_1 - \tilde{q}_\alpha| \, \mathrm{d}\boldsymbol{x} + \int_\Omega \rho \cdot |u_2 - \tilde{q}_\alpha| \, \mathrm{d}\boldsymbol{x} && (18)\\ &= (1-\rho) \cdot E_{\text{shape}}(u_1) + \rho \cdot E_{\text{shape}}(u_2) && (19) \end{aligned}$$

□

For the sake of completeness we write down the complete energy functional:

$$\begin{aligned} E(u) = &\int_\Omega f(\boldsymbol{x})u(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} + \gamma \int_\Omega |\boldsymbol{\nabla} u(\boldsymbol{x})| \, \mathrm{d}\boldsymbol{x} \\ &+ \rho \int_\Omega \xi(u(\boldsymbol{x})) \, \mathrm{d}\boldsymbol{x} + \beta \int_\Omega |u(\boldsymbol{x}) - \tilde{q}_\alpha(\boldsymbol{x})| \, \mathrm{d}\boldsymbol{x}. \end{aligned} \tag{20}$$

Since the norm function is not continuously differentiable we will replace it with a smoothed version by introducing a small offset $\epsilon \in \mathbb{R}$:

$$|u|_\epsilon = \sqrt{u^2 + \epsilon^2}. \tag{21}$$

In experiments we often used $\epsilon = 0.001$.

Now we can formulate the Euler-Lagrange equation of (20):

$$\frac{\partial E}{\partial u} = f - \gamma \mathrm{div}\left(\frac{\boldsymbol{\nabla} u}{|\boldsymbol{\nabla} u|_\epsilon}\right) + \rho\xi'(u) + \beta\frac{u-\tilde{q}_\alpha}{\sqrt{(u-\tilde{q}_\alpha)^2+\epsilon^2}} = 0 \tag{22}$$

Without the shape term you can solve equation (22) as a system of linear equations, e.g. with successive over-relaxation (SOR). Details on this can be found in [9]. We write the new shape term in equation (22) as:

$$s(u) = \frac{u-\tilde{q}_\alpha}{\sqrt{(u-\tilde{q}_\alpha)^2+\epsilon^2}} \tag{23}$$

Due the fact that $s(u)$ is not linear in u we have to perform a linearization by first-order Taylor expansion:

$$s(u_t) = s\left(u^{t-1}\right) + s'\left(u^{t-1}\right)\cdot\left(u^t - u^{t-1}\right) \tag{24}$$

$$= s\left(u^{t-1}\right) + \frac{\epsilon^2}{((u^{t-1}-\tilde{q}_\alpha)+\epsilon^2)^{3/2}}\cdot(u^t - u^{t-1}) \tag{25}$$

Since we use a iterative solver such as SOR we know the solution of u from the last timestep $t-1$ and denote it here as u^{t-1}. The value of $s(u^{t-1})$ can then be seen as a constant. With this we can generate a system of linear equations. For the SOR formalism we need a linear system of equations of the form $\mathbf{Au} = \mathbf{b}$. For this we write u as a vector $\mathbf{u}$, such that the columns of the image matrix are concatenated to an N-dimensional column vector with N the number of pixels. The vector $\mathbf{b}$ is given by the constant part of (22),

$$b_i = -f - \beta\cdot s\left(u^{t-1}(i)\right) - \beta\cdot s'\left(u^{t-1}(i)\right)\cdot u^{t-1}(i). \tag{26}$$

Accordingly, $\mathbf{A}$ contains the u^t-depended part (22). It is useful to replace the function $\xi(u)$ in the actual implementation with a simple thresholding. We obtain for $\mathbf{A} = (a_{ij})$:

$$a_{ij} = \begin{cases} g_{i\sim j} & \text{if} \quad j \in \mathcal{N}(i) \\ \beta\cdot s'(u^{t-1}(i)) - \sum\limits_{k\in\mathcal{N}(i)} g_{i\sim k} & \text{if} \quad i = j \\ 0 & \text{otherwise} \end{cases} \tag{27}$$

where $g_{i\sim j}$ is the diffusivity between pixel i and its neighbor j. $\mathcal{N}(i)$ denotes the neighborhood of pixel i. The Matrix $\mathbf{A}$ is diagonally dominant. In our experiments we use a 4-connected neighborhood, so we get only five non-zero diagonals. All other entries of $\mathbf{A}$ are zero. Because the diffusivity $g = \frac{1}{|\boldsymbol{\nabla} u|}$ depends on the actual solution for u, we do not really have a linear system of equations, but we make the assumption, that the diffusion is constant, and we perform a new computation of it only every L iterations.

For a speedup in the computation time we use the red-black computation scheme for SOR (see [11] for details). With this we schedule the computation

parallel, so that we create a separate thread for every pixel that computes the solutions using the latest information from its neighbor. For this computation we use the NVIDIA CUDA framework, so the main computing is done in parallel on the GPU.

3.2 Estimation of the Optimal Transformation Parameters for Every Shape

Given an initial solution of u we need to determine the number of object candidates in the segmented image. Since u is almost binary this can be solved easily, e.g. through connected components. This gives us the number K of possible objects in the input image I. For each of these candidates we need to know its transformation parameters Θ_k.

Using a parallel framework we can compute the residuum

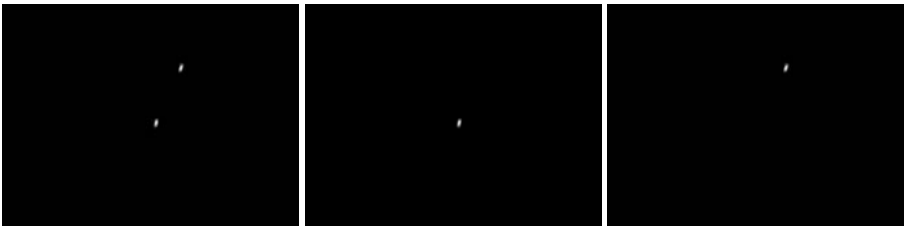
$$\mathbf{r} = \mathbf{b} - \mathbf{A}\mathbf{u}. \tag{28}$$

given the actual solution u, all transformation parameters Θ_k and all shape parameters $\boldsymbol{\alpha}_k$ for $k = 1, 2, ..., K$. The estimation of the optimal Θ_k for all k is done by computing a "branch & bound" search on the space of valid transformations parameters. For initialization we set the values of the translation parameters to the barycenter of each candidate. The norm of $\mathbf{r}$ indicates the correctness of the found parameters. In every node in the branch & bound searching tree we save the actual intervals for all parameters, the norm of $\mathbf{r}$ and an indicator holding the information which interval of a parameter has to be divided for the next level of the search. The search is stopped if a satisfying accuracy is achieved, e.g. when the residuum does not change any more. It was shown in [10] that this approach leads to a globally optimal solution. Although our derivation is more general, the extension of the proof shown there is straight-forward and will not be presented here.

3.3 Estimating the Optimal Shape Representation

Knowing the actual solution of u and the optimal transformation parameters Θ_k we have to estimate the optimal shape parameters $\boldsymbol{\alpha}_k$ for every candidate $k = 1, 2, ..., K$. This can be summarized in three steps:

1. divide $\tilde{q}_\alpha$ into $q_{\boldsymbol{\alpha}_1}^{\Theta_1}, q_{\boldsymbol{\alpha}_2}^{\Theta_2}, ..., q_{\boldsymbol{\alpha}_K}^{\Theta_K}$, such that each $q_{\boldsymbol{\alpha}_k}^{\Theta_k}$ only contains information of candidate k (cf. Figure 3),
2. transform the eigenmodes $\psi_1, ..., \psi_n$ with $\Phi_u(\psi_i, \Theta_k)$ for each eigenmode $i = 1, .., n$ and each candidate $k = 1, ..., K$, so that you get a transformed set of eigenmodes $\boldsymbol{\Psi}_k$ for each candidate,
3. solve:
$$\min_{\boldsymbol{\alpha}_k} \left\| \boldsymbol{\Psi}_k^T \cdot \boldsymbol{\alpha}_k - \left(q_{\boldsymbol{\alpha}_k}^{\Theta_k} - \Phi_u(\mu, \Theta_k) \right) \right\| \tag{29}$$
for all $k = 1, ..., K$.



a) b) c)

Fig. 3. Examples of Step 1 in section 3.3: a) $\tilde{q}_\alpha$, b) $q_{\boldsymbol{\alpha}_1}^{\Theta_1}$, c)$q_{\boldsymbol{\alpha}_2}^{\Theta_2}$

The first two steps can be easily implemented. The third step can be solved in different ways. We use in our experiments a singular value decomposition (SVD) to obtain $\boldsymbol{\alpha}_k$. Due the fact that n is a very small number (in our case 3 or 5) the computation time of the SVD is short. So we do not need a more complex solving algorithm. If the training set database contains many dissimilar shapes then n will be larger and a different computation strategy for step 3 would be probably faster.

3.4 Algorithm Summary

Now we can summarize the whole algorithm:

1. solve (22) with $\beta = 0$ to get an initial solution for u only based on the color information,
2. determine the number of object candidates K,
3. estimate the optimal translation parameters Θ_k for $k = 1, 2, ..., K$ using branch & bound,
4. estimate the optimal shape parameters $\boldsymbol{\alpha}_k$ solving (29) for $k = 1, 2, ..., K$,
5. check for each candidate $k = 1, 2, ..., K$ whether the segmented object matches the found shape representation $q_{\boldsymbol{\alpha}_k}^{\Theta_k}$ and discard false responses,
6. solve (22) with $\beta \neq 0$ to get a optimal solution for u based on color and shape information
7. if the accuracy is sufficient stop, else return to step 2.

Step 5 can be realized with the following procedure. First divide the segmentation u into disjoint images $u_1, u_2, ..., u_K$, so that each u_k contains only the information of u that corresponds to candidate k. Since we have already found the optimal translation and shape parameters of the corresponding shape $q_{\boldsymbol{\alpha}_k}^{\Theta_k}$, we can now simply compute the difference of u_k and $q_{\boldsymbol{\alpha}_k}^{\Theta_k}$:

$$d_k\left(u_k, q_{\boldsymbol{\alpha}_k}^{\Theta_k}\right) = \left|u_k - q_{\boldsymbol{\alpha}_k}^{\Theta_k}\right|. \tag{30}$$

If the cumulated and normalized difference is bigger than a threshold $\tau \in \mathbb{R}$, then the candidate is discarded, and we save this information, such that the candidate will not re-appear in the segmentation. This can be realized with:

$$I(\boldsymbol{x}) = \begin{cases} u_k(\boldsymbol{x}) \cdot \boldsymbol{\nu}_{\text{bgd}} + (1 - u_k(\boldsymbol{x})) \cdot I(\boldsymbol{x}) & , \frac{1}{\|\Omega\|} \cdot \int\limits_{\Omega} d_k(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} > \tau \\ I(\boldsymbol{x}) & \text{otherwise} \end{cases} \tag{31}$$

A more precise shape verification strategy, e.g. shape matching, can be applied to step 5, but was not needed in our experiments. Since shape matching generally needs high computation times we solved this problem here in a simpler way to save runtime. Some fast algorithms for shape matching are described in [12] and [13].

Fig. 4. Object detection results using shape and color information. Left column: input images. Right column: Detection results presented as colored version of the segmentation result u. Each color represents a label for a pixel.

4 Results

In this section we present the results obtained with the proposed algorithm. We performed the presented experiments on a Intel Core2Quad 8200 CPU with 4GB RAM and a NVIDIA GeForce GTX280 with 1GB RAM. As already described in Section 2, we use a hand database which we test on bureau images. In addition to this we created a car database from manually segmented aerial images. These images were taken from a height of about 500 meters obove ground with opening angles of 13.6 and 10.4 degrees. The resolution of both image categories is 1024 $\times$ 768 pixels.

Results can be seen in Figure 4. The first input image shows a bureau scene with hands in it. The challenge with this image is the high level of noise and

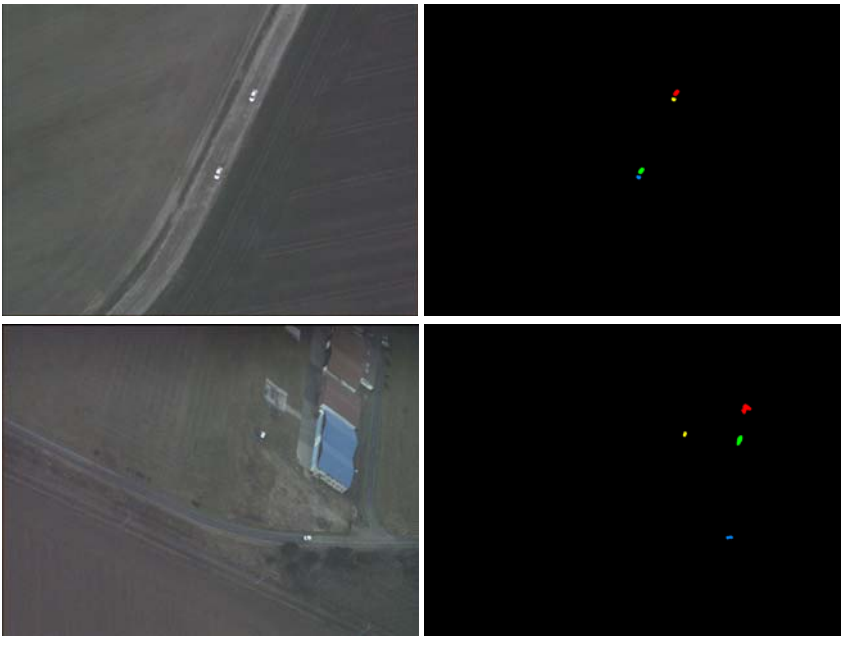


Fig. 5. Object detection results using only color information. Left column: input images. Right column: Detection results presented as colored version of the segmentation result u. Each color represents a label for a pixel.

Table 1. Running times for multiple object detection based on aerial images

size	sec
256×192	0.475
512×384	1.077
1024×768	3.192

strongly varying color distribution of both hands. Despite these difficulties our method yields the correct segmentation and the corresponding positions of objects of interest in this scene. The next images show aerial scenes in which cars shall be detected. Here, the challenging point is the fact that often the color of the car windows differ strongly from the rest of the car. This leads a algorithm only controlled by color information to the belief, that a car in a scene consists of two objects. Examples for this behavior can be seen in Figure 5. But the fusion of color and shape information yields the correct segmentation. Furthermore, the man-made objects in this scene (e.g. houses) have the same color distribution as the car, so they will appear as object candidates when using color information only (c.f Figure 5). In addition to this the shape representation of a car is quite unspecific (c.f Figure 2), so a shape-only algorithm will not work properly. The benefit lies here in the fusion of both approaches.

Table 1 displays the running times for our algorithm with a GPU-based solution of (22). Since these times depend on the number of objects found, we used the first aerial image from Figure 4 with different resolutions for our time measurements. We did not use a parallel version of SVD to solve (29). This would further decrease the computation time.

5 Conclusion

In this work we presented a novel method for a globally optimal multiple object detection using shape and color information. The proposed method is based on a convex energy functional for image segmentation. We showed how this functional can be parallized to improve the computation time. In experiments we demonstrated the capabilities of this approach with challenging scenes.

In future work we indend to decrease the computation time through a parallel solving of (29) and a faster solving method for the transition parameters of the objects.

References

1. Kass, M., Witkin, A., Terzopoulos, D.: Snakes: Active contour models. International Journal of Computer Vision 1(4), 321–331 (1988)
2. Mumford, D., Shah, J.: Optimal approximations by piecewise smooth functions and variational problems. CPAM XLII(5), 577–685 (1989)
3. Cremers, D.: Dynamical statistical shape priors for level set based tracking. IEEE Trans. on Patt. Anal. and Mach. Intell. 28(8) (August 2006)
4. Cremers, D., Tischhuser, F., Weicker, J., Schnörr, C.: Diffusion snakes: Introducing statistical shape knowledge into the mumford-shah functional. International Journal of Computer Vision 50(3), 295–313 (2002)
5. Rousson, M., Paragios, N.: Prior knowledge, level set representations and visual grouping. International Journal of Computer Vision 76(3), 231–243 (2008)
6. Rousson, M., Paragios, N., Deriche, R.: Implicit active shape models for 3D segmentation in mri imaging. In: MICAI. LNCS, vol. 2217, pp. 209–216. Springer, Heidelberg (2004)

7. Chambolle, A.: Total variation minimization and a class of binary MRF models. In: Rangarajan, A., Vemuri, B.C., Yuille, A.L. (eds.) EMMCVPR 2005. LNCS, vol. 3757, pp. 136–152. Springer, Heidelberg (2005)
8. Chan, T.F., Esedoglu, S., Nikolova, M.: Algorithms for finding global minimizers of image segmentation and denoising models. SIAM Journal on Applied Mathematics 66(5), 1632–1648 (2006)
9. Schikora, M., Häge, M., Ruthotto, E., Wild, K.: A convex formulation for color image segmentation in the context of passive emitter localization. In: International Conference on Information Fusion (July 2009)
10. Cremers, D., Schmidt, F.R., Barthel, F.: Shape priors in variational image segmentation: Convexity, lipschitz continuity and globally optimal solutions. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Anchorage, Alaska (June 2008)
11. Klodt, M., Schoenemann, T., Kolev, K., Schikora, M., Cremers, D.: An experimental comparison of discrete and continuous shape optimization methods. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 332–345. Springer, Heidelberg (2008)
12. Schmidt, F.R., Töppe, E., Cremers, D.: Efficient planar graph cuts with applications in computer vision. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Miami, Florida (June 2009)
13. Schmidt, F.R., Farin, D., Cremers, D.: Fast matching of planar shapes in sub-cubic runtime. In: IEEE International Conference on Computer Vision (ICCV), Rio de Janeiro, Brazil (October 2007)