12th International Conference on Information Fusion
Seattle, WA, USA, July 6-9, 2009

A Convex Formulation for Color Image Segmentation in the Context of Passive Emitter Localization

Marek Schikora, Miriam Häge, Eicke Ruthotto and Klaus Wild
FGAN Research Institute for Communication,
Information Processing and Ergonomics (FKIE)
D-53343 Wachtberg, Germany
{schikora, haege, ruthotto, klaus_wild}@fgan.de

Abstract – *In many tasks in information fusion objects of interest need to be extracted from color images. Often the only available information is the color of a specific object. In this paper we present a novel method for segmenting images into two regions, foreground (e.g. object) and background. We introduce a convex energy functional based on total variation that is subsequently solved using the Euler-Lagrange theorem and a parallel implementation of successive over-relaxation. The main achievement of this formulation is that, due to the convex formulation, the algorithm is guaranteed to find the global optimum from all possible solutions. Furthermore, this algorithm can be heavily parallized using the graphics processing unit (GPU). In the following we will show how to use the thus obtained image information in the context of passive emitter localization from aerial platforms. It will be shown that the fusion of image- and bearing-based localization results can strongly improve the bearings-only results.*

Keywords: image segmentation, emitter localization, parallel computing, total variation

1 Introduction

The problem of extracting relevant objects from images can be seen as the segmentation of an image into two regions, foreground and background. All pixels labeled as foreground count as part of an object and are interesting candidates for further analysis. Image segmentation is a common task in computer vision, and many solutions have been proposed for this problem. A good solution is provided by variational approaches. There exist three main classes of variational approaches for image segmentation, the first one being level sets [1, 2, 3]. The main advantage is that the energy functional being minimized is formulated continuously, so there is no need for discretization. On the other hand, the local optimization of the energy functional does not necessarily lead to globally optimal solutions. The second class are the graph cuts [4, 5, 6, 7] with two main advantages: the computation time is very short, even for large images, and the solution is approximately globally optimal. The main disadvantage of this approach is the discrete formulation on a graph, which leads to discretization errors. A combination of the benefits from these two approaches constitutes the third class: total variation (TV) minimization using the total variation norm. Chan et al. [8] proposed this method in 2004 for image segmentation of intensity-based images using a transformed version of the Mumford-Shah model. In this paper we will present an extension of this approach to color-based segmentation and show that it can be computed in far above real-time using a parallel implementation on the GPU. Then we will present a framework for fusion of the extracted image information with bearing data for the localization of radio emitters. This fusion of bearing- and image-based localization results is accomplished using multiple hypotheses tracking (MHT) [9, 10]. It will be shown that the fusion of image and bearing information can strongly improve the bearings-only results.

2 Color-TV-Segmentation

The segmentation of an image $I : \Omega \to [0,1]^3 \subset \mathbb{R}^3$ with $\Omega \subseteq \mathbb{R}^2$ can be seen as a separation of the image plane into disjoint regions $\Omega_1, \Omega_2, \ldots, \Omega_n$,

$$\Omega = \Omega_1 \cup \ldots \cup \Omega_n \cup \Gamma\,, \tag{1}$$

where Γ denotes the contour of the segmentation, $\Gamma = \partial\Omega_1 \cup \ldots \cup \partial\Omega_n$ and $\Omega_i \cap \Omega_j = \emptyset$ for all $i \neq j \in \{1, \ldots, n\}$.

In the case discussed here there will be only two regions Ω_{obj} and Ω_{bgd}, so we are looking for a binary image $u : \Omega \to \{0, 1\}$.

2.1 Formulation of a convex energy functional

To perform the segmentation we will formulate an energy functional based on the Mumford-Shah model [11]. A transformation using the total variation concept will ensure the

convexity of the functional. The total variation of a function $u : \Omega \rightarrow \mathbb{R}$ is defined as

$$\mathrm{TV}(u) = \sup_{\phi \in \Phi} \left\{ \int_{\Omega} u(\mathbf{x}) \, \mathrm{div}\phi(\mathbf{x}) \, \mathrm{d}\mathbf{x} \right\} \tag{2}$$

with

$$\Phi = \left\{ \phi \in C^1(\Omega, \mathbb{R}) \; : \; |\phi(\mathbf{x})| \leq 1 \quad \forall \mathbf{x} \in \Omega \right\} . \tag{3}$$

If u is continuously differentiable, then

$$\mathrm{TV}(u) = \int_{\Omega} |\boldsymbol{\nabla} u| \; \mathrm{d}\mathbf{x} \, . \tag{4}$$

For binary functions u which are the characteristic function of Ω_{obj} the TV-norm $\mathrm{TV}(u)$ is equal to the length of the contour of Ω_{obj}:

$$\int_{\Omega} |\boldsymbol{\nabla} u| \; \mathrm{d}\mathbf{x} = \mathrm{Per}(\Omega_{\mathrm{obj}}) \tag{5}$$

with $\mathrm{Per}(\Omega_{\mathrm{obj}}) = \int_{\Gamma_{\mathrm{obj}}} \mathrm{d}s$ the length of the contour Γ_{obj}.

The Mumford-Shah model for the case of two regions and intensity-based images is

$$\begin{aligned} E(\Gamma, \mu_{\mathrm{obj}}, \mu_{\mathrm{bgd}}) = & \int_{\Omega_{\mathrm{obj}}} \left[I(\mathbf{x}) - \mu_{\mathrm{obj}} \right]^2 \; \mathrm{d}\mathbf{x} \\ & + \int_{\Omega_{\mathrm{bgd}}} \left[I(\mathbf{x}) - \mu_{\mathrm{bgd}} \right]^2 \; \mathrm{d}\mathbf{x} + \nu \, |\Gamma| \; , \end{aligned} \tag{6}$$

where μ_{obj} and μ_{bgd} are average values for the intensities of both regions, object and background, and ν is a parameter weighting the impact of the contour length. The desired segmentation follows from the solution of

$$\frac{\partial E}{\partial \Gamma} = 0 \tag{7}$$

and can be obtained using the Euler-Lagrange theorem.

To perform a segmentation of a color image we first transform the input image I from the RGB to the HSV color space,

$$I^{\mathrm{HSV}} = \mathrm{rgb2hsv}(I). \tag{8}$$

The advantage of the HSV color space lies in its intuitive and simple model, which yields in a much better interaction with the user, then the RGB color space. Now we can rewrite the Mumford-Shah model for color images as follows:

$$\begin{aligned} E(\Gamma, \boldsymbol{\mu}_{\mathrm{obj}}, \boldsymbol{\mu}_{\mathrm{bgd}}) = & \int_{\Omega_{\mathrm{obj}}} \Delta \left(I^{\mathrm{HSV}}(\mathbf{x}), \; \boldsymbol{\mu}_{\mathrm{obj}} \right) \; \mathrm{d}\mathbf{x} \\ & + \int_{\Omega_{\mathrm{bgd}}} \Delta \left(I^{\mathrm{HSV}}(\mathbf{x}), \; \boldsymbol{\mu}_{\mathrm{bgd}} \right) \; \mathrm{d}\mathbf{x} + \nu \, |\Gamma| \end{aligned} \tag{9}$$

with

$$\begin{aligned} \Delta(I^{\mathrm{HSV}}(x), \; \boldsymbol{\mu}) \; &= \; w_{\mathrm{H}} \left(I^{\mathrm{H}}(\mathbf{x}) - \mu^{\mathrm{H}} \right)^2 \\ &+ \; w_{\mathrm{S}} \left(I^{\mathrm{S}}(\mathbf{x}) - \mu^{\mathrm{S}} \right)^2 \\ &+ \; w_{\mathrm{V}} \left(I^{\mathrm{V}}(\mathbf{x}) - \mu^{\mathrm{V}} \right)^2 , \end{aligned} \tag{10}$$

w_{H}, w_{S} and w_{V} being (normalized) weighting parameters for the individual channels. The disadvantage of (9) is that it is not convex, because of the local optimization along the contour Γ. Methods such as gradient descent tend to get stuck in local minima in this case.

To solve this problem we will now transform this functional using the total variation. For binary functions the length of the contour Γ is proportional to its TV norm,

$$|\Gamma| = \int_{\Omega} |\boldsymbol{\nabla} u| \; \mathrm{d}\mathbf{x} \, . \tag{11}$$

This allows us to rewrite (9) as

$$\begin{aligned} & E(u, \boldsymbol{\mu}_{\mathrm{obj}}, \boldsymbol{\mu}_{\mathrm{bgd}}) = \\ & \int_{\Omega_{\mathrm{obj}}} \Delta(I^{\mathrm{HSV}}(\mathbf{x}), \; \boldsymbol{\mu}_{\mathrm{obj}}) \, \mathrm{d}\mathbf{x} \; + \int_{\Omega_{\mathrm{bgd}}} \Delta(I^{\mathrm{HSV}}(\mathbf{x}), \; \boldsymbol{\mu}_{\mathrm{bgd}}) \, \mathrm{d}\mathbf{x} \\ & \qquad + \nu \int_{\Omega} |\boldsymbol{\nabla} u| \, \mathrm{d}\mathbf{x} \end{aligned} \tag{12}$$

$$\begin{aligned} = & \int_{\Omega} \Big[\Delta(I^{\mathrm{HSV}}(\mathbf{x}), \; \boldsymbol{\mu}_{\mathrm{obj}}) \, u(\mathbf{x}) \\ & \quad + \Delta(I^{\mathrm{HSV}}(\mathbf{x}), \; \boldsymbol{\mu}_{\mathrm{bgd}}) \, (1 - u(\mathbf{x})) \Big] \, \mathrm{d}\mathbf{x} + \; \nu \int_{\Omega} |\boldsymbol{\nabla} u| \, \mathrm{d}\mathbf{x} \end{aligned} \tag{13}$$

$$\begin{aligned} = & \int_{\Omega} \Big[(\Delta(I^{\mathrm{HSV}}(\mathbf{x}), \; \boldsymbol{\mu}_{\mathrm{obj}}) - \Delta(I^{\mathrm{HSV}}(\mathbf{x}), \; \boldsymbol{\mu}_{\mathrm{bgd}}) \Big] \, u(\mathbf{x}) \, \mathrm{d}\mathbf{x} \\ & \quad + \nu \int_{\Omega} |\boldsymbol{\nabla} u| \, \mathrm{d}\mathbf{x} \, . \end{aligned} \tag{14}$$

Until now we have assumed u as a binary function. Since minimization in the space of binary functions is not convex, we will interpret u as real-valued. It is thus possible for u to converge against $+\infty$ or $-\infty$. To eliminate this effect we introduce an additional term into the functional that penalizes all values of u outside of the valid range of $0 \leq u(\mathbf{x}) \leq 1$. An example for such a function is (cf. Figure 1)

$$\Theta(u(\mathbf{x})) = \max \left[0, \; 2 \, |u(\mathbf{x}) - 0.5| - 1 \right] . \tag{15}$$

The final energy functional for globally optimal color image

segmentation is then given by

$$E(u, \boldsymbol{\mu}_{\text{obj}}, \boldsymbol{\mu}_{\text{bgd}}) = \int_\Omega \left[\Delta(I^{\text{HSV}}(\mathbf{x}),\ \boldsymbol{\mu}_{\text{obj}}) - \Delta(I^{\text{HSV}}(\mathbf{x}),\ \boldsymbol{\mu}_{\text{bgd}})\right] u(\mathbf{x})\ \mathrm{d}\mathbf{x} + \nu \int_\Omega |\boldsymbol{\nabla} u|\ \mathrm{d}\mathbf{x} + \alpha \int_\Omega \Theta(u(\mathbf{x}))\ \mathrm{d}\mathbf{x}\,. \tag{16}$$

Theorem 1. *The energy functional (16) is convex.*

Proof. For the sake of conciseness, let us define $D := (\Delta(I^{\text{HSV}}(\mathbf{x}),\ \boldsymbol{\mu}_{\text{obj}}) - \Delta(I^{\text{HSV}}(\mathbf{x}),\ \boldsymbol{\mu}_{\text{bgd}}))$. Now we show that (16) is convex with respect to u. For this purpose we have to show that for all $\lambda \in (0, 1)$:

$$\forall u_1, u_2 : E((1-\lambda)u_1 + \lambda u_2) \leq (1-\lambda)E(u_1) + \lambda E(u_2) \tag{17}$$

So we can write

$$E((1-\lambda)u_1 + \lambda u_2) = \int_\Omega D \cdot ((1-\lambda)u_1 + \lambda u_2)\ \mathrm{d}\mathbf{x} + \nu \int_\Omega |\boldsymbol{\nabla}((1-\lambda)u_1 + \lambda u_2)|\ \mathrm{d}\mathbf{x} + \alpha \int_\Omega \Theta((1-\lambda)u_1 + \lambda u_2)\ \mathrm{d}\mathbf{x} \tag{18}$$

$$= \int_\Omega (1-\lambda)\cdot D \cdot u_1 + \lambda \cdot D \cdot u_2\ \mathrm{d}\mathbf{x} + \nu \int_\Omega |(1-\lambda)\boldsymbol{\nabla} u_1 + \lambda \boldsymbol{\nabla} u_2|\ \mathrm{d}\mathbf{x} + \alpha \int_\Omega \Theta((1-\lambda)u_1 + \lambda u_2)\ \mathrm{d}\mathbf{x} \tag{19}$$

$$\leq \int_\Omega (1-\lambda)\cdot D \cdot u_1 + \lambda \cdot D \cdot u_2\ \mathrm{d}\mathbf{x} + \nu \int_\Omega |(1-\lambda)\boldsymbol{\nabla} u_1| + |\lambda \boldsymbol{\nabla} u_2|\ \mathrm{d}\mathbf{x} + \alpha \int_\Omega |(1-\lambda)\Theta(u_1) + \lambda \Theta(u_2)\ \mathrm{d}\mathbf{x} \tag{20}$$

$$= (1-\lambda)\int_\Omega D \cdot u_1\ \mathrm{d}\mathbf{x} + \lambda \int_\Omega D \cdot u_2\ \mathrm{d}\mathbf{x} \tag{21}$$

$$+\nu(1-\lambda)\int_\Omega |\boldsymbol{\nabla} u_1|\mathrm{d}\mathbf{x} + \nu\lambda \int_\Omega |\boldsymbol{\nabla} u_2|\ \mathrm{d}\mathbf{x} + \alpha(1-\lambda)\int_\Omega \Theta(u_1)\mathrm{d}\mathbf{x} + \alpha\lambda \int_\Omega \Theta(u_2)\ \mathrm{d}\mathbf{x} \tag{22}$$

$$= (1-\lambda)E(u_1) + \lambda E(u_2) \tag{23}$$

□

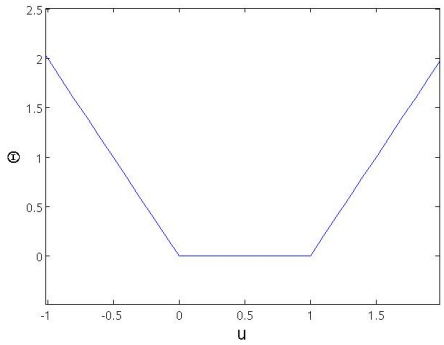

Figure 1: Plot of the regularization function $\Theta(u(\mathbf{x}))$ from Equation (15).

2.2 Efficient computation of the optimal segmentation

The segmentation algorithm consists of two steps:

1. Find global minimum of (16) for $u : \Omega \to [0, 1]$.
2. Find the desired binary function by threshholding: $\Omega_{\text{obj}} = \{\mathbf{x} \in \Omega | u(\mathbf{x}) > \lambda, \lambda \in (0, 1)\}$.

λ is an arbitrary threshold from the open interval (0,1). Due to the convexity of (16) the global minimum can be found with local optimization methods. Here, we use a parallel implementation of successive over-relaxation (SOR).

The Euler-Lagrange equation of (16) is

$$\frac{\partial E}{\partial u} = \frac{\partial \mathcal{L}}{\partial u} - \frac{\mathrm{d}}{\mathrm{d}\mathbf{x}}\frac{\partial \mathcal{L}}{\partial \boldsymbol{\nabla} u} \tag{24}$$

$$= -\Delta(I^{\text{HSV}},\ \boldsymbol{\mu}_{\text{obj}}) + \Delta(I^{\text{HSV}},\ \boldsymbol{\mu}_{\text{bgd}}) - \nu\,\text{div}\left(\frac{\boldsymbol{\nabla} u}{|\boldsymbol{\nabla} u|}\right) - \alpha\,\Theta'(u) \tag{25}$$

In this case the Lagrange function $\mathcal{L}$ is defined as:

$$\mathcal{L} = \left[\Delta(I^{\text{HSV}}(x),\ \boldsymbol{\mu}_{\text{obj}}) - \Delta(I^{\text{HSV}}(x),\ \boldsymbol{\mu}_{\text{bgd}})\right] u(x) + \nu\,|\boldsymbol{\nabla} u| + \alpha\Theta(u(x)) \tag{26}$$

To use the SOR formalism we need to generate a linear system of equations of the form $\mathbf{A}\,\mathbf{u} = \mathbf{b}$. We write u as a vector $\mathbf{u}$, so 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 the Euler-Lagrange equation (25).

$$b_i = -\Delta(I^{\mathrm{HSV}},\, \boldsymbol{\mu}_{\mathrm{obj}}) + \Delta(I^{\mathrm{HSV}},\, \boldsymbol{\mu}_{\mathrm{bgd}}) \qquad (27)$$

Accordingly, $\mathbf{A}$ contains the u-dependent part of (25). It is useful to replace the function $\Theta(u)$ in the actual implementation with a simple thresholding. We obtain for $\mathbf{A} = (a_{ij})$:

$$a_{ij} = \begin{cases} g_{i\sim j}, & \text{if} \quad j \in \mathcal{N}(i) \\ -\sum\limits_{k\in\mathcal{N}(i)} g_{i\sim k}, & \text{if} \quad i = j \\ 0, & \text{otherwise} \end{cases}, \qquad (28)$$

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. On the main diagonal we have the sums of all diffusions to all neighboring pixels. For each neighbor there exists another diagonal. In our experiments we use a 4-connected neighborhood, so we get only five diagonals with content. All other entries of $\mathbf{A}$ are zero.

With this information the SOR-method can easily be implemented. Since 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 assumtion that the diffusion is constant, and we perform a new computation only every K iterations.

For a speedup in the computation time we use the red-black computation scheme for SOR (see [12] for details). With this scheme we can parallelize the computation, so that we create a separate thread for every pixel which computes the solution using the latest information from its neighbors. For the computation we use the NVIDIA CUDA framework, so the main computing is done in parallel on the GPU.

2.3 Results

The results in this section were computed on a Intel Core2Quad Q8200 CPU with 4GB RAM and a NVIDIA GeForce GTX280 with 1GB RAM. Table 1 shows the running times for the segmentation of color images of different sizes and the number of frames or images you can compute per second (FPS) on the GPU. In this configuration we are far above real-time image processing, which is usually at 20–30 FPS. This algorithm could even be used with a high-speed camera.

Table 1: Running times for color image segmentation.

size	CPU (sec)	GPU (sec)	FPS
256×192	1.039	0.004	250
512×384	4.181	0.004	250
1024×768	16.783	0.005	200

In Figure 2 we present some segmentation results for images obtained from a aerial platform equipped with a steerable camera. The only prior information about the objects of interest is the color: yellow.

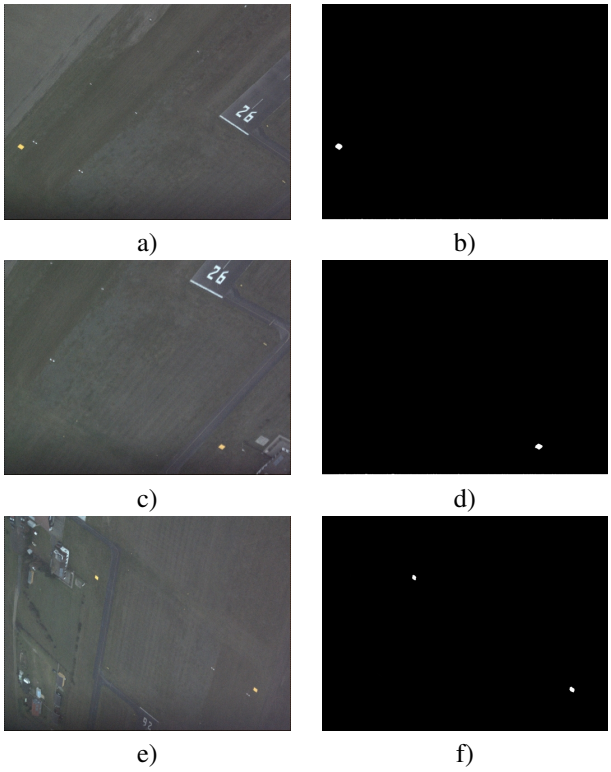

Figure 2: Segmentation results for aerial color images. Task: find yellow object. a), c) and e) are the input images, b), d) and f) the segmentation results.

3 Fusion of Image and Bearing Data

In this section we will show how to use the segmentation results from aerial images to improve passive emitter localization. Traditionally, emitters are localized using the signal information received by an antenna array, which is subsequently condensed into bearing angles. To improve the accuracy of bearing-only localization results we will combine the available bearing data with image data. Therefore we have to extract bearing-like information from the image segmentation. For each object we need to calculate two angles, azimuth and elevation, in the local coordinate system of the bearing sensor. The workflow of our algorithm is presented in Figure 3.

Having accumulated a number of bearings and having obtained a first localization result based on this data, e.g. by Maximum Likelihood estimation, a stearable camera on the

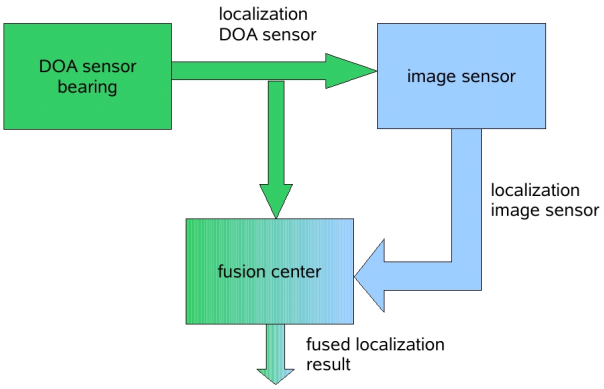

Figure 3: Workflow of a DOA sensor and image sensor fusion algorithm

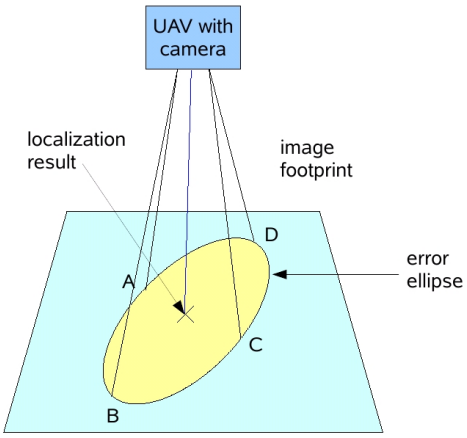

Figure 4: Error ellipse in the image footprint

aerial platform is pointed to the estimated emitter position. The thus obtained image is then subject to the segmentation algorithm, and the optical bearings to the extracted objects are determined. In the next step another bearing measurement is performed and a new localization, based on the larger measurement set, is carried out. This scheme iterates, as long as signals from the emitter are received. A closer look at the extration of image information is presented in the following, and a brief formulation of the localization algorithm will conclude this section.

3.1 Extracting angles from a binary image

In practice we do not need to perform the segmentation on the whole image, since a traditional bearing-/localization-system produces, apart from the position of an object, also a confidence region in form of a covariance error ellipse. Projecting this ellipse into the image plane, the search space is considerably reduced. Because of the arbitrary position of the ellipse in a image we approximate it through a bounding, so that no information is cut away. The only information of this ellipse are the max/min angles of the ellipse axes, see Figure 4.

Let $\mathbf{P}^k$ be the 3×4 projection matrix at time t_k,

$$\mathbf{P}^k = \mathbf{K} \cdot \mathbf{R}^k \cdot \left[\mathbf{I} \mid -\mathbf{x}_0^k \right] . \tag{29}$$

$\mathbf{K}$ is the 3×3 calibration matrix, $\mathbf{I}$ is the 3×3 identity matrix, $\mathbf{R}^k$ and $\mathbf{x}_0^k$ describe the position and attitude of the camera in the local coordinate system of the bearing sensor: $\mathbf{R}^k$ is a 3×3 rotation matrix and $\mathbf{x}_0^k$ is a 3×1 vector with cartesian position information. They can be obtained from a GPS/INS system on the platform. The matrix $\mathbf{K}$ can be estimated with calibration measurements in a laboratory. So we can calculate the projection of the extremum points { A, B, C, D } of the ellipse into the image plane. For the point A writen in homogeneous coordinates as $\mathbf{a}$ you get:

$$\mathbf{a}' = \mathbf{P}^k \cdot \mathbf{a} = \mathbf{K} \cdot \mathbf{R}^k \cdot \left[\mathbf{I} \mid -\mathbf{x}_0^k \right] \cdot \mathbf{a} \tag{30}$$

$$a_x = \frac{a_1'}{a_3'} \tag{31}$$

$$a_y = \frac{a_2'}{a_3'} \tag{32}$$

The maximal/minimal axe coordinates are given by

$$x_{\max} = \max\{a_x, b_x, c_x, d_x\} \tag{33}$$

$$x_{\min} = \min\{a_x, b_x, c_x, d_x\} \tag{34}$$

$$y_{\max} = \max\{a_y, b_y, c_y, d_y\} \tag{35}$$

$$y_{\min} = \min\{a_y, b_y, c_y, d_y\} . \tag{36}$$

If the angle between one axis of the ellipse and the x-axis of the image is close to $\frac{\pi}{4}$, then a simple bounding box with the size $(x_{\max} - x_{\min}) \times (y_{\max} - y_{\min})$ will cut off information, see Figure 5. To solve this we expand the x and y size with a factor $q(\alpha)$, with α the angle between the x axis of the image and a axis of the ellipse. So we get the new sizes

$$s_x = (x_{\max} - x_{\min})\,(1 + q(\alpha)) \tag{37}$$

$$s_y = (y_{\max} - y_{\min})\,(1 + q(\alpha)) . \tag{38}$$

The function $q(\alpha)$ is given by a Gaussian centered around $\frac{\pi}{4}$. The center of the bounding box follows from the projection of the last localization result.

Within this bounding box we need to compute the optical bearings to all the extracted objects. To this end we first calculate the barycenter of each object in the binary segmentation image. Let u_c be the binary image inside the bounding box, then all barycenters can be calculated as

$$\mathbf{c}_i = \frac{1}{\Omega_i} \int_{\Omega_i} u_c(\mathbf{x}) \cdot \mathbf{x} \, \mathrm{d}\mathbf{x} . \tag{39}$$

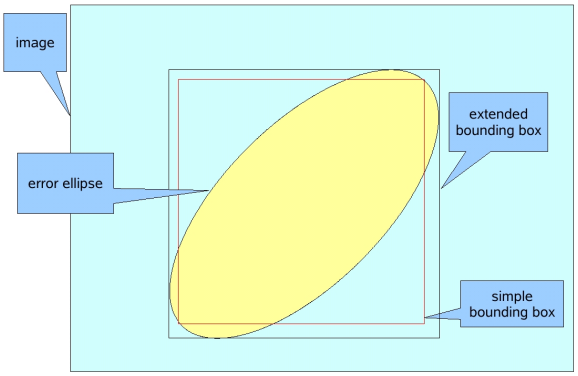

Figure 5: Bounding boxes for error ellipse projected into the image.

So we obtain a set $\mathbf{c} = \{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_m\}$ of barycenters for m found objects. Ω_i is the region containing object i.

For each $\mathbf{c}_i$ we have to calculate the corresponding angle in the coordinate system of the bearing sensor. We can rewrite the projection matrix P into a 3×3 matrix $\mathbf{S}$ and a 3×1 vector $\mathbf{v}$,

$$\mathbf{P}^k = [\,\mathbf{S} \mid \mathbf{v}\,]\,. \tag{40}$$

With this we can now write

$$\lambda \cdot \mathbf{c}_i = [\,\mathbf{S} \mid \mathbf{v}\,] \begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix} = \mathbf{S}\mathbf{x} + \mathbf{v} \tag{41}$$

$$\mathbf{x} = \mathbf{S}^{-1}(\lambda \mathbf{c}_i - \mathbf{v})\,. \tag{42}$$

In this case λ is the distance from the camera to the object. This distance can be obtained from a digital elevation model (DEM). $\mathbf{x}$ is the direction vector from the camera to the extracted object in the real world.

3.2 Localization algorithm

The bearing-only localization is based on the algorithm proposed in [13]. When using automated image segmentation it is possible that the object of interest is not detected (probability of detection $P_D < 1$ in the parlance of target tracking) and/or that unwanted objects are extracted (false alerts). Therefore, a fusion algorithm must be able to handle such ambiguous situations; this is accomplished by including the complete set of data interpretations in the likelihood function, which leads data fusion by multiple hypotheses tracking (MHT).

If we have one emitter and, at time t_k, n_k measurements, then there exist n_k+1 possible data interpretations (hypotheses):

h_k^0 : emitter was not detected; n_k false reports (1 hypothesis)

h_k^j : emitter detected; measurement z_k^j is the emitter; $n_k - 1$ false reports (n_k hypotheses) for $j = 1, ..., n_k$.

Using the Kalman filter with the MHT likelihood, we obtain the probability density function (pdf) for the state of the emitter $\mathbf{x}_k$ based on all measurements $\mathcal{Z}^k$ up to the time t_k as:

$$p(\mathbf{x}_k|\mathcal{Z}^k) = \sum_{i=1}^{\hat{n}_{k-1}} \sum_{j=0}^{n_k} p_k^{ij}\, \mathcal{N}(\mathbf{x}_k;\, \mathbf{x}_{k|k}^{ij},\, \mathbf{P}_{k|k}^{ij}). \tag{43}$$

Here, the index i labels the $\hat{n}_{k-1}$ track hypotheses from time step $k-1$ and $\mathbf{x}_{k|k}^{ij}$, $\mathbf{P}_{k|k}^{ij}$ are the filtering results (mean and covariance) of hypothesis i with measurement j. p_k^{ij} are the normalized hypotheses weights. So, at each time step, the pdf is given by a weighted sum of gaussian hypotheses for the emitter state.

Since in every iteration step new hypotheses are generated, it is necessary to avoid a combinatorial disaster by dropping hypotheses with weights below a given threshold (pruning), combining similar hypotheses (merging) and taking only those measurements into account that fall within a certain expectation region around the prediction (gating). For details on MHT we refer to [14] and [9].

3.3 Simulation results

In the simulations we carried out, an emitter was positioned at the origin of the coordinate system and a Poisson-distributed number of false targets were placed randomly in the field of interest (FOI) around the true emitter position. In order to test the robustnes of our algorithm, we simulated false reports of the image sensor by scattering, at each time step, a Poisson-distributed number of clutter measurements evenly in the FOI. For the sake of simplicity we simulated a circular flight course at constant height. With this choice of the observer motion, the observability requirements for bearing-only location are satisfied from the beginning. Localization results at different stages of the simulation are shown in Figure 7 in form of the (multi-modal) probability density function (pdf) over the field of interest. In the begining there are three main hypotheses for the emitter location. After some iterations the false hypotheses are depleted and nearly allprobability mass is concentrated at the true target position. The simulation was repeated a thousand times, and the mean error of the localization is shown in Figure 6. It can be seen that the fusion of bearing and image data can considerably increase the localization accuracy.

4 Conclusion

We have presented in this paper a novel approach for color image segmentation based on the variational concept of total variation. Due to the parallel implementation we achieve very fast running times even for large images. Because of the convex formulation our algorithm always finds the globally optimal solution. Furthermore, we have outlined how to use these segmentation results to improve the accuracy of bearing-based emitter localization with aerial platforms.

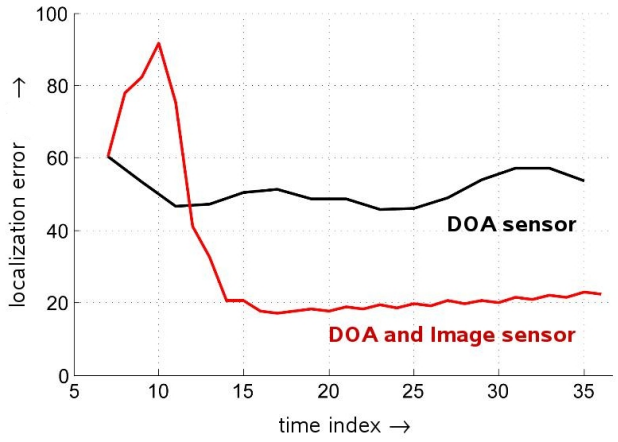

Figure 6: Comparison of performance for bearing-only (DOA sensor) and fused bearing-image (DOA and Image sensor) localization.

Our simulation results show that the fusion of both systems can increase the accuracy of the localization by a factor of two.

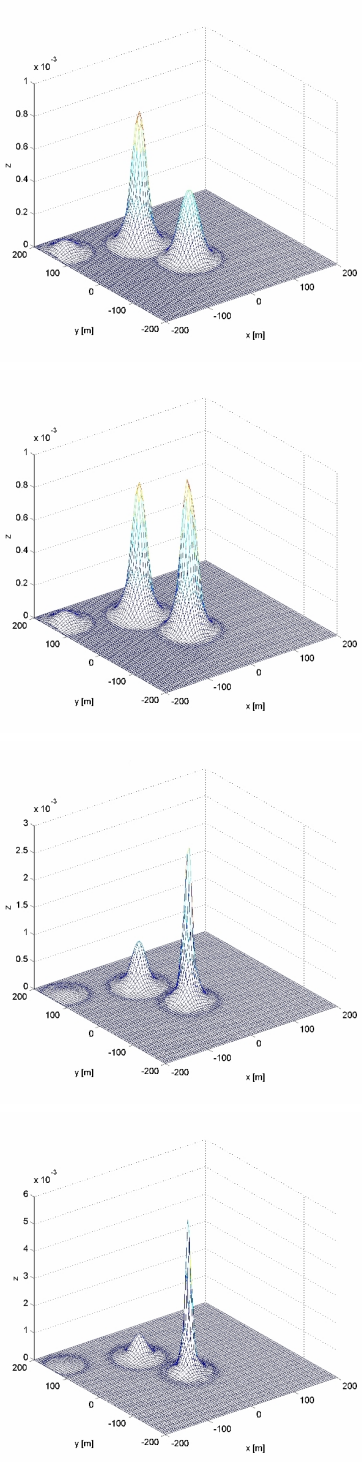

Figure 7: Simulation results: pdf for the emitter location at diffrent timesteps. In the begining there are three hypotheses of different weights. In the end nearly all probability mass is concentrated at the true emitter position.

References

[1] S. Osher and J. A. Sethian. "Fronts propagating with curvature-dependent speed: Algorithms based on hamilton-jacobi formulations". *Journal of Computational Physics*, 79: pp. 12–49, 1988.

[2] T. F. Chan and L. A. Vese. "Active contours without edges". *IEEE Transactions on Image Processing*, 10(2): pp. 266–277, 2001.

[3] L. A. Vese and T. F. Chan. "A multiphase level set framework for image segmentation using the mumford and shah model". *International Journal of Computer Vision*, 50(3): pp. 271–293, dec 2002.

[4] D. M. Greig, B. T. Porteous, and A. H. Seheult. "Exact maximum a posteriori estimation for binary images". *Journal of the Royal Statistical Society, Series B*, 51(2): pp. 271–279, 1989.

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

[6] Y. Boykov and V. Kolmogorov. " An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision". In *Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages pp. 359–374, 2001.

[7] V. Kolmogorov and R. Zabih. “What energy functions can be minimized via graph cuts?”. In *ECCV*, pages pp. 65–81, 2002.

[8] T. F. Chan, S. Esedoglu, and M. Nikolova. “ Algorithms for finding global minimizers of image segmentation and denoising models”. Technical Report 54, UCLA, September 2004.

[9] W. Koch. “Advanced target tracking techniques”. *Advanced Radar Signal and Data Processing*, RTO-EN-SET-086 AC/323(SET-086)TP/54, September 2006.

[10] W. Koch. “On bayesian tracking and data fusion: A tutorial introduction with examples”. *IEEE, Aerospace and Electronic Systems Magazine, Tutorials IV*, October 2008. to be published.

[11] D. Mumford and J. Shah. “Optimal approximations by piecewise smooth functions and variational problems”. *CPAM*, XLII(5): pp. 577–685, 1989.

[12] M. Klodt, T. Schoenemann, K. Kolev, M. Schikora, and D. Cremers. “An experimental comparison of discrete and continuous shape optimization methods”. In *European Conference on Computer Vision (ECCV)*, Marseille, France, October 2008.

[13] K. Becker. “Three-dimensional target motion analysis using angle and frequency measurements”. *IEEE Transactions on Aerospace and Electronic Systems*, 41, issue 1: pp.284–301, January 2005.

[14] W. Koch. *“Advanced Signal Processing Handbook”*, chapter Target Tracking. Boca Raton, 2001.