

# Moment Constraints in Convex Optimization for Segmentation and Tracking

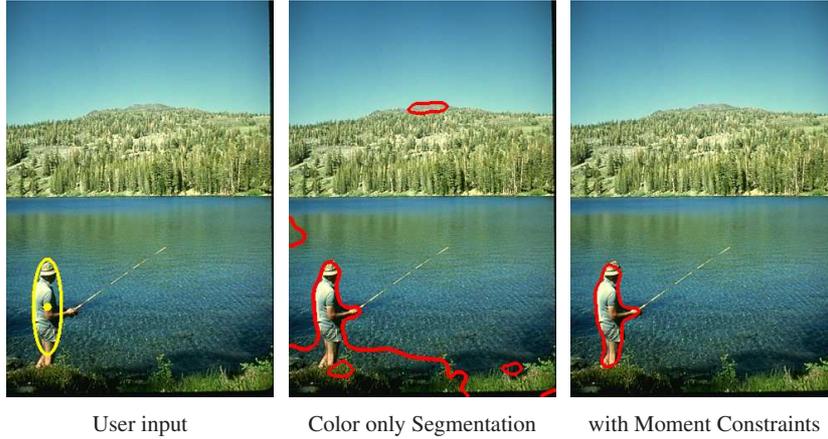
Maria Klodt and Frank Steinbrücker and Daniel Cremers

**Abstract** Convex relaxation techniques have become a popular approach to shape optimization as they allow to compute solutions independent of initialization to a variety of problems. In this chapter, we will show that shape priors in terms of moment constraints can be imposed within the convex optimization framework, since they give rise to convex constraints. In particular, the lower-order moments correspond to the overall area, the centroid, and the variance or covariance of the shape and can be easily imposed in interactive segmentation methods. Respective constraints can be imposed as hard constraints or soft constraints. Quantitative segmentation studies on a variety of images demonstrate that the user can impose such constraints with a few mouse clicks, leading to substantial improvements of the resulting segmentation, and reducing the average segmentation error from 12% to 0.35%. GPU-based computation times of around 1 second allow for interactive segmentation.

## 1 Introduction

Shape optimization is at the heart of several classical computer vision problems such as image segmentation and multi view reconstruction. Following a series of seminal papers [12, 1, 19], functional minimization has become the established paradigm to solve shape optimization problems such as image segmentation or 3D reconstruction. In the spatially discrete setting the study of the corresponding binary labeling problems goes back to the spin-glass models introduced in the 1920's [11]. Popular algorithms to solve the arising shape optimization problems include level set methods [20], graph cuts [9] or convex relaxation [4].

In this chapter, we focus on a class of functionals of the form:



**Fig. 1** We propose a convex formulation for interactive image segmentation which allows to impose constraints on moments of arbitrary order. In particular, constraints on the lower order moments (area, centroid, covariance) are easily transmitted through mouse interaction (left). They allow to stabilize the segmentation process while preserving fine-scale details of the shape (right).

$$E(S) = \int_{int(S)} f(x) dx + \int_S g(x) dA, \quad (1)$$

where  $S$  denotes a hyper surface in  $\mathbb{R}^d$ , i.e. a set of closed boundaries in the case of  $2D$  image segmentation or a set of closed surfaces in the case of  $3D$  segmentation and multi view reconstruction. The functions  $f : \mathbb{R}^d \rightarrow \mathbb{R}$  and  $g : \mathbb{R}^d \rightarrow \mathbb{R}^+$  are application dependent. In a statistical framework for image segmentation, for example,

$$f(x) = \log p_{bg}(I(x)) - \log p_{ob}(I(x)), \quad (2)$$

may denote the log likelihood ratio for observing the color  $I(x)$  at a point  $x$  given that  $x$  is part of the background or the object, respectively. A common choice for  $g$  is the function

$$g(x) = \frac{1}{1 + \beta |\nabla I_\sigma(x)|^2} \quad (3)$$

with suitable choices for the parameters  $\beta$  and  $\sigma$ . Here,  $I_\sigma$  is a Gaussian smoothed version of the input image.

The second term in (1) corresponds to the area (for  $d = 3$ ) or the boundary length (for  $d = 2$ ), measured in a metric given by the function  $g$ . In the context of image segmentation,  $g$  may be a measure of the local edge strength – as in the geodesic active contours [3, 13] – which energetically favors segmentation boundaries along strong intensity gradients. In the context of multi view reconstruction,  $g(x)$  typically measures the photo-consistency among different views of the voxel  $x$ , where low

values of  $g$  indicate a strong agreement from different cameras on the observed patch intensity.

### 1.1 Shape Priors for Image Segmentation

There has been much research on imposing prior shape knowledge into image segmentation. While it was shown that segmentation results can be substantially improved by imposing shape priors [10, 5, 8], existing approaches typically suffer from the following problems:

- Apart from a few exceptions such as [23] – computable solutions are only *locally* optimal. As a consequence, one typically needs appropriate initializations and solutions may be arbitrarily far from the globally optimal ones. Other notable exceptions were designed by Veksler and coworkers for specific scenarios such as “compact objects” [6] or “star-shaped objects” [25].
- Many shape priors have a rather fine granularity in the sense that they impose the object silhouette to be consistent with those silhouettes observed in a training set [5, 7]. The degree of abstraction is typically rather small. In particular, deviations of the observed shape from the training shapes are (elastically) suppressed by the shape prior. This is particularly undesirable in medical image segmentation where malformations of organs (that make it deviate from the training shapes of healthy organs) should be detected rather than ignored. It may therefore be of interest to merely impose some coarse-level shape information rather than imposing the exact form of the object.

An alternative approach that may provide a remedy for both of the above problems is to impose moment constraints. In particular, the lower-order moments allow to constrain the area/volume, the centroid and the size or covariance of objects without imposing any constraints on their local shape. A related idea of using *Legendre moments* (albeit in a local optimization scheme) was developed in [8].

In a convex formulation of multiple view 3D reconstruction, it was recently shown [15] that one can impose additional convex constraints which assure that the computed minimal surfaces are silhouette-consistent. Essentially this constraint can be seen as a *volume constraint*: The volume along any ray from the camera center must be at least 1 if that ray passes through the silhouette and zero otherwise. In the two-dimensional case, a related constraint was recently proposed as a bounding box prior for image segmentation [17].

### 1.2 Contribution

We show that one can impose an entire family of moment constraints in the framework of convex shape optimization, thereby generalizing from the zeroth order mo-

ment (volume) to higher order moments (centroid, scale, covariance, etc). In particular, all moment constraints - both soft or hard - correspond to convex constraints. As a consequence we can compute moment-constrained shapes which are independent of initialization and lie within a bound of the optimum.

The outline of the paper is as follows. In Section 2, we will briefly review a framework for convex relaxation and thresholding which allows to efficiently compute global minima of the above energies in a spatially continuous setting. In Section 3, we will then show that moment constraints can be imposed as convex constraints within the optimization framework. In Section 4 we show how the arising optimization problem can be minimized using efficient GPU-accelerated PDE solving. We will furthermore show that computing projections onto the moment constraint sets reduces to solving systems of linear equations. Section 5 shows how the presented method for image segmentation with moment constraints can be extended to a method for object tracking in videos. In Section 6 we present experimental results and a quantitative evaluation showing that interactive segmentation results can be drastically improved using moment constraints.

## 2 Shape Optimization via Convex Relaxation

Functionals of the form (1) can be globally optimized in a spatially continuous setting by means of convex relaxation and thresholding [4]. To this end, one reverts to an implicit representation of the hyper surface  $S$  using an indicator function  $u \in BV(\mathbb{R}^d; \{0, 1\})$  on the space of binary functions of bounded variation, where  $u = 1$  and  $u = 0$  denote the interior and exterior of  $S$ . The functional (1) defined on the space of surfaces  $S$  is therefore equivalent to the functional

$$E(u) = \int_{\Omega} f(x)u(x) dx + \int_{\Omega} g(x)|Du(x)|, \quad (4)$$

where the second term in (4) is the weighted total variation. Here  $Du$  denotes the distributional derivative which for differentiable functions  $u$  boils down to  $Du(x) = \nabla u(x)dx$ . By relaxing the binary constraint and allowing the function  $u$  to take on values in the interval between 0 and 1, the optimization problem becomes that of minimizing the convex functional (4) over the convex set  $BV(\mathbb{R}^d; [0, 1])$ . Global minimizers  $u^*$  of this relaxed problem can therefore efficiently be computed, for example by a simple gradient descent procedure.

The thresholding theorem [4] assures that thresholding the solution  $u^*$  of the relaxed problem preserves global optimality for the original binary labeling problem. We can therefore compute global minimizers for functional (4) in a spatially continuous setting as follows: Compute a global minimizer  $u^*$  of (4) on the convex set  $BV(\mathbb{R}^d; [0, 1])$  and threshold the minimizer  $u^*$  at any value  $\mu \in (0, 1)$ .

The following theorem [4] assures that thresholding the solution  $u^*$  of the relaxed problem provides a minimizer of the original binary labeling problem (4), in

other words the convex relaxation preserves global optimality for the original binary labeling problem.

**Theorem 1.** *Let  $u^* \in BV(\mathbb{R}^d; [0, 1])$  be a global minimizer of the functional (4). Then all upper level sets (i.e. thresholded versions)*

$$\Sigma_{\mu, u^*} = \{x \in \mathbb{R}^d \mid u^*(x) > \mu\}, \quad \mu \in (0, 1), \quad (5)$$

*of  $u^*$  are minimizers of the original binary labeling problem (1).*

*Proof.* Using the layer cake representation of the function  $u^* \in BV(\mathbb{R}^d; [0, 1])$ :

$$u^*(x) = \int_0^1 1_{\Sigma_{\mu, u^*}} d\mu \quad (6)$$

we can rewrite the first term in the functional (4) as

$$\int_{\mathbb{R}^d} f u^* dx = \int_{\mathbb{R}^d} f \left( \int_0^1 1_{\Sigma_{\mu, x}} d\mu \right) dx = \int_0^1 \int_{\Sigma_{\mu, u^*}} f(x) dx \quad (7)$$

As a consequence, the functional (4) takes on the form:

$$E(u^*) = \int_0^1 \left\{ \int_{\Sigma_{\mu, u^*}} f dx + |\partial \Sigma_{\mu, u^*}|_g \right\} d\mu \equiv \int_0^1 \hat{E}(\Sigma_{\mu, u^*}) d\mu, \quad (8)$$

where we have used the coarea formula to express the weighted total variation norm in (4) as the integral over the length of all level lines of  $u$  measured in the norm induced by  $g$ . Clearly the functional (8) is now merely an integral of the original binary labeling problem  $\hat{E}$  applied to the upper level sets of  $u^*$ .

Assume that for some threshold value  $\tilde{\mu} \in (0, 1)$  theorem 1 was not true, i.e. there exists a minimizer  $\Sigma^*$  of the binary labeling problem with smaller energy:

$$\hat{E}(\Sigma^*) < \hat{E}(\Sigma_{\tilde{\mu}, u^*}). \quad (9)$$

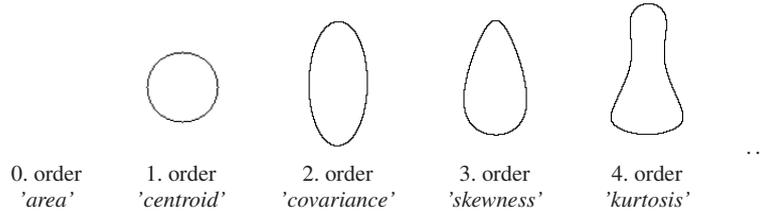
Then – using continuity arguments – we have for the indicator function  $1_{\Sigma^*}$  of the set  $\Sigma^*$ :

$$E(1_{\Sigma^*}) = \int_0^1 \hat{E}(\Sigma^*) d\mu < \int_0^1 \hat{E}(\Sigma_{\mu, u^*}) d\mu = E(u^*), \quad (10)$$

which contradicts the assumption that  $u^*$  was a global minimizer of (4).  $\square$

The above theorem allows to compute global minimizers of the functional (4) in a spatially continuous setting as follows:

1. Compute a global minimizer  $u^*$  of (4) on the convex set  $BV(\mathbb{R}^d; [0, 1])$ .
2. Threshold the minimizer  $u^*$  at any value  $\mu \in (0, 1)$  to obtain a binary solution of the original shape optimization problem.



**Fig. 2** 2D central moments of a shape. With increasing order of the moment more details of the shape can be described.

### 3 Moment Constraints for Segmentation

In the following, we will describe the moment constraints presented in [14]. We will successively constrain the moments of the segmentation and show how all of these constraints give rise to nested convex sets. To this end we will represent shapes in  $d$  dimensions as binary indicator functions  $u \in BV(\Omega; \{0, 1\})$  of bounded variation on the domain  $\Omega \subset \mathbb{R}^d$ . We will denote the convex hull of this set by  $\mathcal{B} = BV(\Omega; [0, 1])$ .

Figure 2 shows examples for the first five 2D central moments of a shape. The first order moment describes the centroid of a shape, the second describes the covariance, i.e. the relation between width and height of an object. While with the third order moment egg-shapes of shapes can be described, for the fourth order moments the effect on the shape already becomes less intuitive. The figure shows how the higher the order of a moment, the more sophisticated properties can be imposed on the corresponding shape. In the following, we will show how these moments can be used to constrain properties of shapes for segmentation in a convex framework, and how especially the lower order moments can constrain shape optimization in a very intuitive way.

#### 3.1 Area Constraint

We can impose that the area of the shape  $u$  to be bounded by a constant  $c \in \mathbb{R}^+$  by constraining  $u$  to lie in the set:

$$\mathcal{C}_0 = \left\{ u \in \mathcal{B} \mid \int_{\Omega} u \, dx = c \right\}. \quad (11)$$

In the case of a 3 dimensional domain  $\Omega$  the constant  $c$  constrains the volume of a shape.

**Proposition 1.** *For any constant  $c \geq 0$ , the set  $\mathcal{C}_0$  is convex.*

*Proof.* Let  $u_1, u_2 \in \mathcal{C}_0$  be two elements from this set. Then for any convex combination  $u_\alpha = \alpha u_1 + (1 - \alpha)u_2$ ,  $\alpha \in [0, 1]$  of these elements we have:

$$\int_{\Omega} u_\alpha dx = \alpha \int_{\Omega} u_1 dx + (1 - \alpha) \int_{\Omega} u_2 dx. \quad (12)$$

As a consequence we have  $\int_{\Omega} u_\alpha dx = c$  such that  $u_\alpha \in \mathcal{C}_0$ .

In practice, we can either impose an exact area or we can impose upper and lower bounds on the area with two constants  $c_1 \leq c_2$  and constrain the area to lie in the range  $[c_1, c_2]$

Alternatively, we can impose a soft area constraint by enhancing the functional (4) as follows:

$$E_0(u, \lambda_0) = E(u) + \lambda_0 \left( \int_{\Omega} u dx - c \right)^2, \quad (13)$$

which imposes a soft constraint with a weight  $\lambda_0 > 0$  favoring the area of the estimated shape to be near  $c \geq 0$ . Clearly, the functional (13) is also convex.

### 3.2 Centroid Constraint

Assume that someone gave us some bounds about the centroid (center of gravity) for the object we want to reconstruct. We can impose the centroid of the shape by constraining the solution  $u$  to the set  $\mathcal{C}_1$ :

$$\mathcal{C}_1 = \left\{ u \in \mathcal{B} \mid \frac{\int_{\Omega} x u dx}{\int_{\Omega} u dx} = \mu \right\}, \quad (14)$$

where equality is to be taken point wise and  $\mu \in \mathbb{R}^d$ . Alternatively, we can impose the centroid to lie between two constants  $\mu_1, \mu_2 \in \mathbb{R}^d$ . For  $\mu_1 = \mu_2$ , the centroid is fixed.  $\mathcal{C}_1$  contains all shapes whose first order moment is  $\mu$ . In probability theory the first order moment is also called the expected value.

**Proposition 2.** *For any constant  $\mu \geq 0$ , the set  $\mathcal{C}_1$  is convex.*

*Proof.* The equality constraint in (14) is equivalent to

$$\int_{\Omega} x u dx = \mu \int_{\Omega} u dx, \quad (15)$$

which is clearly a linear constraint.

Reformulating the constraint equation in (14) leads to

$$\int_{\Omega} (\mu - x) u dx = 0, \quad (16)$$

allowing us to impose the centroid as a soft constraint by minimizing the energy:

$$E_1(u, \lambda_1) = E(u) + \lambda_1 \left( \int_{\Omega} (\mu - x)u \, dx \right)^2. \quad (17)$$

Energy (17) is also convex in  $u$ . Interestingly this soft constraint does not minimize the quadratic difference to the specified center  $\mu$ ; the latter would not be convex. In contrast to the hard constraint, this soft constraint unfortunately exhibits a preference toward smaller shapes as it vanishes with decreasing object size:

$$\left( \int_{\Omega} (\mu - x)u \, dx \right)^2 = \left( \mu - \frac{\int_{\Omega} xu \, dx}{\int_{\Omega} u \, dx} \right)^2 b^2, \quad (18)$$

where  $b = \int u \, dx$  denotes the size of the object.

### 3.3 Covariance Constraint

The proposed concept can be generalized to moments of successively higher order, where we shall focus on so-called central moments (i.e. moments with respect to a specified centroid). In particular, the respective structures will generally be tensors of higher dimension. One can impose the covariance structure by considering the following convex set:

$$\mathcal{C}_2 = \left\{ u \in \mathcal{B} \mid \frac{\int_{\Omega} (x - \mu)(x - \mu)^{\top} u \, dx}{\int_{\Omega} u \, dx} = A \right\}, \quad (19)$$

where the equality constraint should be taken element wise. Here  $\mu \in \mathbb{R}^d$  denotes the center and  $A \in \mathbb{R}^{d \times d}$  denotes a symmetric matrix. Alternatively, we can impose the covariance to lie in a range between two symmetric matrices  $A_1, A_2 \in \mathbb{R}^{d \times d}$  such that  $A_1 \leq A_2$  element-wise. Constraining the covariance, i.e. the second order moment, we are able to constrain the relation between width and height of an object. This constraint is particularly meaningful if one additionally constrains the centroid to be  $\mu$ , i.e. considers the intersection of the set (19) with a set of the form (14).

**Proposition 3.** *For any constant  $A \geq 0$ , the set  $\mathcal{C}_2$  is convex.*

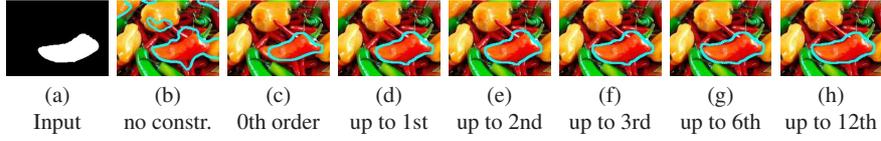
*Proof.* The proof is analogous to that of proposition 2.

We can derive the corresponding soft constraint on the covariance matrix by adding a respective term to the original energy:

$$E_2(u, \lambda_2) = E(u) + \lambda_2 \left( \int_{\Omega} \left( A - (x - \mu)(x - \mu)^{\top} \right) u \, dx \right)^2. \quad (20)$$

Note that this allows, in particular, to constrain the scale  $\sigma$  of the object, because:

$$\sigma^2 = \frac{\int_{\Omega} (x - \mu)^2 u \, dx}{\int_{\Omega} u \, dx} = \text{tr} \frac{\int_{\Omega} (x - \mu)(x - \mu)^{\top} u \, dx}{\int_{\Omega} u \, dx}. \quad (21)$$



**Fig. 3** Segmentation results with higher order moment constraints: By imposing constraints of increasing order (up to 12th order) more and more fine scale details of the shape are restored.

From the constraint in (19) it follows that:

$$\text{tr}(A_1) \leq \sigma^2 \leq \text{tr}(A_2). \quad (22)$$

Here,  $\text{tr}$  denotes the trace of a matrix.

### 3.4 Higher Order Moment Constraints

In general, we can impose constraints on moments of any order  $k \in \mathbb{N}$ :

$$\mathcal{C}_k = \left\{ u \in \mathcal{B} \mid \frac{\int_{\Omega} (x_1 - \mu_1)^{i_1} \cdots (x_d - \mu_d)^{i_d} u \, dx}{\int_{\Omega} u \, dx} = a_{i_1 \dots i_d} \right\}, \quad (23)$$

where  $i_1 + \cdots + i_d = k$  and  $a_{i_1 \dots i_d}$  can be chosen arbitrarily to constrain the moment tensor of order  $k$ . Here  $x_i$  and  $\mu_i$  denotes the  $i$ -th component of  $x$  and  $\mu$  respectively.

**Proposition 4.** For all  $i_1, \dots, i_d \in \mathbb{N}$  and for any constants  $a_{i_1 \dots i_d} \leq b_{i_1 \dots i_d}$ , the set  $\mathcal{C}_{i_1 \dots i_d}$  is convex.

*Proof.* The proof is analogous to that of proposition 2.

The above properties allow to impose various constraints on the shape associated with the indicator function  $u$ . Imposing more and more constraints of increasingly higher order leads to a smaller and smaller intersection of the associated convex sets as a feasible domain of the shape and a corresponding hierarchy of shape details being imposed in the segmentation. How much shape detail can one impose in this manner?

**Proposition 5.** Similarity to any given shape can be imposed at arbitrary detail by imposing convex moment constraints of increasingly higher order.

*Proof.* According to the uniqueness theorem of moments [21], the function  $u$  is uniquely defined by its moment sequence.

Figure 3 shows an example of segmentations with high order moment constraints: While the higher-order moments allow to recover fine-scale shape details, the shape

improvements due to higher order constraints are fairly small. Furthermore imposing moments of higher order is not very practical: Firstly the user cannot estimate these moments visually. Secondly the user cannot transmit respective higher-order tensors through a simple mouse interaction. Instead, having the image data determine the shape's fine scale structure turns out to be far more useful.

Fig. 3 (a) shows the manually segmented input shape which is used to compute color histograms for foreground and background, as well as the moments that constrain the subsequent image segmentation. 3 (b) shows that color-only segmentation based on color histograms for foreground and background is not sufficient in this example image to correctly segment one red pepper from the others. Fig. 3 (c)-(e) show that the first three moment constraints – area, centroid and covariance – are able to substantially improve segmentations, however are in this hard case, not enough for an exact segmentation. In Fig. 3 (f)-(h) we can see how with increasing order of moment constraints more and more fine scale details of the shape can be reconstructed. Even the 12th order moment constraint still results in an improvement of segmentation.

In the example of Fig. 3 the respective higher order moment constraints have been learned from a training shape. For interactive applications however the higher order moments are too less intuitive to be applicable. Therefore, for the application we consider – namely interactive image segmentation – we shall in the following limit ourselves to imposing moments up to 2nd order (area/volume, center of mass, scale and covariance).

## 4 Shape Optimization with Moment Constraints

Shape optimization and image segmentation with respective moment constraints can now be done by minimizing convex energies under respective convex constraints.

Let  $\mathcal{C}$  be a specific convex set containing knowledge about respective moments of the desired shape – given by an intersection of the above convex sets. Then we can compute segmentations by solving the convex optimization problem

$$\min_{u \in \mathcal{C}} E(u), \quad (24)$$

with  $E(u)$  given in (4). In this work we solve the Euler-Lagrange equation:

$$0 = \operatorname{div} \left( g \frac{\nabla u}{|\nabla u|} \right) - f. \quad (25)$$

The equation system can be solved using gradient descent, or the lagged diffusivity approach that was presented in [16]. We use the latter because in our experiments it achieves a speed up of computation times of a factor of  $\sim 5$ . An update step for  $u$  at pixel  $i$  and time step  $k$  yields

$$u_i^{l,k+1} = (1 - \omega)u_i^{l,k} + \omega \frac{\mathbf{v} \sum_{j \in \mathcal{N}(i), j < i} g \cdot g_{i \sim j}^l u_j^{l,k+1} + \mathbf{v} \sum_{j \in \mathcal{N}(i), j > i} g \cdot g_{i \sim j}^l u_j^{l,k} - f_i}{\mathbf{v} \sum_{j \in \mathcal{N}(i)} g \cdot g_{i \sim j}^l}, \quad (26)$$

where  $g_{i \sim j}^l$  is the diffusivity between pixel  $i$  and  $j$ , and  $\mathcal{N}(i)$  is the 4-connected neighborhood around  $i$ . The method converges for overrelaxation parameters  $\omega \in (0, 2)$ . We obtained the fastest convergence rate for  $\omega = 1.85$ .

In the case of segmentation without moment constraints, there is only the constraint that  $u \in \mathcal{B}$ , where we project  $u$  such that  $u \in [0, 1]$ . This constraint can be enforced by clipping the values of  $u$  at every point  $x$  after each iteration:

$$u(x) = \begin{cases} 0 & \text{if } u(x) < 0 \\ u(x) & \text{if } 0 \leq u(x) \leq 1 \\ 1 & \text{if } 1 < u(x). \end{cases} \quad (27)$$

#### 4.1 Minimization with Soft Constraints

Soft constraints are implemented by adding additional terms to the cost function, as has been done in (13), (17) and (20). Minimization is then performed by computing the corresponding Euler-Lagrange equations with the additional terms. This leads to solutions that prefer shapes that fulfill the constraints, but do not necessarily exactly fulfill them.

For the area constraint the corresponding Euler-Lagrange equation is computed by derivation of (13). We obtain the equation system:

$$0 = \operatorname{div} \left( g \frac{\nabla u}{|\nabla u|} \right) - f - 2\lambda_0 \left( \int_{\Omega} u - \frac{c}{|\Omega|} dx \right). \quad (28)$$

Note that the area update term for  $u$  is equal for all points in  $\Omega$  since the last term is completely independent of  $x$ . A positive value is added in the case that the area of the current segmentation is smaller than the desired area constraint, and a negative value in the case that it is greater. The term is zero in the case that the constraint is exactly fulfilled. The constraint parameter  $c$  is divided by the size of the image domain  $\int_{\Omega} dx = |\Omega|$  in order to equally distribute the update value over all points.

The centroid soft constraint (17) yields an Euler-Lagrange equation where the update term for the centroid constraint is dependent on the location of  $x$ :

$$0 = \operatorname{div} \left( g \frac{\nabla u}{|\nabla u|} \right) - f - 2 \sum_{i=1}^d \lambda_{1i} \left( \int_{\Omega} (\mu_i - x_i) u dx (\mu_i - x_i) \right), \quad (29)$$

where  $\mu_i$  and  $x_i$  denote the  $i$ -th elements of vectors  $\mu$  and  $x$ . Note that it yields  $d$  additional terms to the equation system, as well as  $d$  additional optimization parameters  $\lambda_{11}, \dots, \lambda_{1d}$ .

Similar functions can be derived for the soft covariance and higher order moment constraints. The corresponding Euler-Lagrange term for an arbitrary moment of order  $k$  gives the following more general formulation:

$$0 = -2 \sum_{i_1=1}^d \dots \sum_{i_k=1}^d \lambda_{ki_1 i_k} \left[ \int_{\Omega} \left( a_{i_1 \dots i_k} - \prod_{l=1}^k (\mu_{i_l} - x_{i_l}) \right) u \, dx \left( a_{i_1 \dots i_k} - \prod_{l=1}^k (\mu_{i_l} - x_{i_l}) \right) \right]$$

In this formulation some terms appear multiple times. This is due to the fact that for all permutations  $\sigma$  of  $i_1 \dots i_k$  we have  $a_{i_1 \dots i_k} = a_{\sigma(i_1) \dots \sigma(i_k)}$ , because of the symmetry of the constraint tensors.

For all soft constraints the additional parameters  $\lambda_{k1}, \dots, \lambda_{kd}$  have to be optimized, with  $k$  being the order of the respective moment.

## 4.2 Computing Projections to Moment Constraint Sets

The moment hard constraints can be implemented by projecting onto the constraint sets during optimization. In our implementation we prefer the projection method over soft constraints, because firstly the constraints can be fulfilled exactly, and secondly there are no additional parameters that need to be optimized.

In the case of moment hard constraints we enforce all constraints during the optimization by back-projecting the current segmentation onto the constraint set after every iteration. Segmentations  $u$  are projected onto the intersection of two convex sets:

1. the convex set  $[0, 1]$  and
2. the intersection of the respective moment constraints.

Since all moment constraints presented in this chapter are linear in  $u$  an orthogonal projection can be directly computed for all combinations of moment constraints. Hence, iterative projections are needed between two sets only – independent of the order or the number of moment constraints. In the following we will show how these projections can be easily computed and implemented. A special case arises in the case of the area constraint, because the projection can be computed in a single step. In all other cases we use the algorithm of [2] to iteratively project onto the intersection of the convex sets.

### 4.2.1 Projection to Area Constraint

The projection onto the range  $[0, 1]$  and onto the area constraint in (11) can be combined in one step.

For any  $u_0 \in \mathcal{B}$ , the projection  $u$  onto those constraints has to solve the convex program

$$\begin{aligned}
u &= \arg \min_{v \in \mathcal{B}} \frac{1}{2} \|v - u_0\|_2^2 \\
s.t. \quad & v(x) - 1 \leq 0 \quad \forall x \\
& -v(x) \leq 0 \quad \forall x \\
& \|v\|_1 - c = 0
\end{aligned}$$

By means of the Karush-Kuhn-Tucker conditions for convex problems, if we can find the functions  $u$ ,  $\xi_1$ , and  $\xi_0$  and a scalar  $v$  that fulfil the following conditions

$$u(x) - u_0(x) + \xi_1(x) - \xi_0(x) + v = 0 \quad \forall x \quad (30)$$

$$u(x) \leq 1 \wedge \xi_1(x) \geq 0 \quad \forall x \quad (31)$$

$$u(x) \geq 0 \wedge \xi_0(x) \geq 0 \quad \forall x \quad (32)$$

$$\xi_1(x) = 0 \vee u(x) = 1 \quad \forall x \quad (33)$$

$$\xi_0(x) = 0 \vee u(x) = 0 \quad \forall x \quad (34)$$

$$\|u\|_1 = c, \quad (35)$$

then  $u$  is the correct projection.

With the following method we can find a solution for the conditions above:

1. Set  $u := u_0$ ,  $\xi_1 := 0$ ,  $\xi_0 := 0$ ,  $v := 0$
2. On all positions where  $u(x) > 1$ , set  $u(x) := 1$ ,  $\xi_1(x) := u_0(x) - 1$
3. On all positions where  $u(x) < 0$ , set  $u(x) := 0$ ,  $\xi_0(x) := -u_0(x)$

All constraints but the area constraint are now fulfilled.

Without loss of generality, let the projection of  $u_0$  to  $[0, 1]$  have a smaller norm than  $c$ . If not, set  $u_2(x) := 1 - u(x)$  and  $c_2 := \|\Omega\| - c$ , switch  $\xi_0$  and  $\xi_1$ , perform the projection onto the volume  $c_2$  as described in the following, and afterwards again reflect  $u_2$  at 1 and switch  $\xi_0$  and  $\xi_1$ .

To fulfill the area constraint (35), we can now raise the sum  $u(x) + \xi_1(x) - \xi_0(x)$  by the same amount in all pixels, and adjust  $v$  such that equation (30) is fulfilled. If  $0 < u(x) < 1$ , raise  $u(x)$ , if  $u(x) = 1$ , raise  $\xi_1(x)$ , if  $\xi_0(x) > 0$ , lower  $\xi_0(x)$ .

Unfortunately, the difference  $v$  is non-trivial, because the area constraint (35) depends on  $u$ , not on  $u + \xi_1 - \xi_0$ . Therefore, we have to employ Algorithm 1: Afterwards, we have to adjust the last iteration, if  $\Delta_c < 0$ . This algorithm computes the desired projection and it does not require to explicitly store  $\xi_0$  and  $\xi_1$ , as the sum in (30) can be stored in one field  $u$  and the checks in the algorithm above can be performed on  $u$  as well. Unfortunately though, the algorithm requires several  $O(n)$  computations in the regression to find the minima and in the update steps.

However, if we sort the discrete pixel positions by their value  $u_0$ , and remap them after the projection, the projection can be performed very easily with Algorithm 2. Let us assume now, that  $u_0$  is stacked to a vector and monotonically decreasing in  $x$  from pixel  $x = 1$  to pixel  $x = |\Omega|$ .

Figure 4 demonstrates the input and output of Algorithm 2, assuming that  $u_0$  and  $u$  are sorted in decreasing order. The optimal value  $v$  that needs to be added to  $u_0$

**Algorithm 1** Simple Projection Algorithm

---

```

v ← 0
Δc ← c - ||u||
while Δc > 0 do
  Δu ← min { min{x∈Ω|ξ0(x)>0} {ξ0(x)}, min{x∈Ω|u(x)<1} {1 - u(x)} }
  Δk ← 0
  for all x ∈ Ω do
    if ξ0(x) > 0 then
      ξ0(x) ← ξ0(x) - Δu
    else if u(x) < 1 then
      u(x) ← u(x) + Δu
      Δk ← Δk + Δu
    else
      ξ1(x) ← ξ1(x) + Δu
    end if
  v ← v + Δu
end for
  Δc ← Δc - Δk
end while

```

---

**Algorithm 2** Projection Algorithm with Sorting

---

```

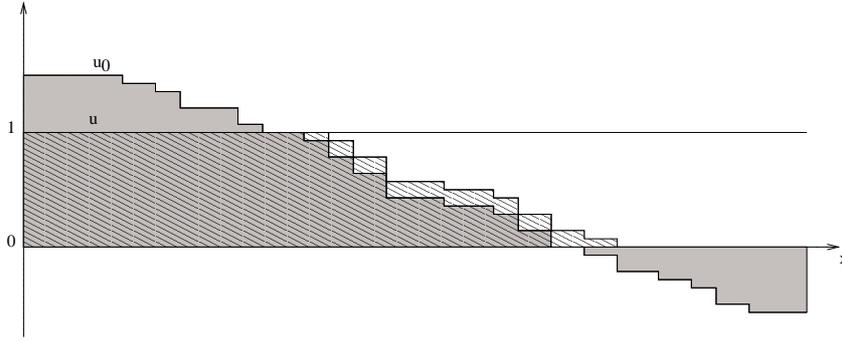
v ← 0
Δc ← c - ||u0||
x1 ← min {x | u0(x) < 1}
x2 ← min {x | u0(x) < 0}
while Δc > 0 do
  Δ1 ← 1 - u0(x1) + v
  Δ2 ← -u0(x2) + v
  if Δ1 > Δ2 then
    v ← v - min { Δ2,  $\frac{\Delta_c}{x_2 - x_1}$  }
    Δc ← Δc - min { (x2 - x1)Δ2, Δc }
    x2 ← min {x | u0(x) - v < 0}
  else
    v ← v - min { Δ1,  $\frac{\Delta_c}{x_2 - x_1}$  }
    Δc ← Δc - min { (x2 - x1)Δ1, Δc }
    x1 ← min {x | u0(x) - v < 1}
  end if
end while

```

---

in order to fulfill both the area constraint  $\int u dx = c$  and the constraint  $u \in [0, 1]$ , is computed in one step of Algorithm 2.

After Algorithm 2 terminates, we simply subtract  $v$  from  $u_0(x)$  in every pixel to get  $u$ ,  $x$ , clamp  $u$  to  $[0, 1]$  and put the difference into the (virtual)  $\xi_0$  and  $\xi_1$ . Now all KKT conditions are fulfilled and  $u$  is the desired projection. Note that every assignment in the algorithm is performed at most  $2n$  times and the algorithm only requires constant memory. Therefore, its complexity is dominated by the sorting algorithm.



**Fig. 4** Area of  $u$  before and after applying Algorithm 2. The gray area denotes the area before the projection: neither the area constraint  $\int u dx = c$  nor the constraint  $u \in [0, 1]$  is fulfilled. The shaded area shows the area of  $u$  after projection: both constraints are fulfilled after two steps of the algorithm.

#### 4.2.2 Projection to Higher Order Moment Constraints

For moments of an order higher than 0 we use the method of [2] to iteratively compute projections of  $u$  onto the intersection of the convex set  $[0, 1]$  and the linear moment constraints. The algorithm of [2] projects to the intersection of convex sets by alternatingly projecting to the respective sets. The correct solution is found by remembering the previous projection for each step. This leads to an algorithm that converges to the solution, although slow.

We can summarize all projections to an arbitrary number of moment constraints in one projection step because all moment constraints are linear sets. Hence, we project between the two sets  $[0, 1]$  and the intersection of the moment constraints.

Centroid Constraint:

We will in the following show the projection formula for the centroid constraint.

An equivalent formulation of (14) is the constraint set

$$\mathcal{C}_1 = \left\{ u \in \mathcal{B} \mid \int_{\Omega} (\mu_i - x_i) u dx = 0, \forall 1 \leq i \leq d \right\}. \quad (36)$$

We are looking for the orthogonal projection  $\hat{m}$  that projects  $u$  to the nearest  $\hat{u}$  in  $\mathcal{C}_1$  s.t.  $\hat{u} = u + \hat{m}$ . Since  $\mathcal{C}_1$  is linear in  $u$  this projection is unique. We have

$$\hat{m} = \arg \min_{m \in \mathcal{M}} \|m\| \quad (37)$$

$$\text{with } \mathcal{M} = \left\{ m \in \mathcal{B} \mid \int_{\Omega} (\mu_i - x_i) m dx = - \int_{\Omega} (\mu_i - x_i) u dx, \forall 1 \leq i \leq d \right\}. \quad (38)$$

The left-hand sides of the constraint equations in  $\mathcal{M}$  are linear in  $m$ , and the right-hand sides are constants (independent of  $m$ ).

Then the following projection theorem [18] holds:

**Theorem 2.** *The orthogonal projection  $\hat{m}$  (projection of minimum norm) to a set of constraints  $(m, y_i) = c_i, \forall 1 \leq i \leq d$  is given by*

$$\hat{m} = \sum_{i=1}^d \beta_i y_i \quad (39)$$

where the coefficients  $\beta_i \in \mathbb{R}$  are the solution to the linear equation system

$$(y_1, y_i)\beta_1 + \dots + (y_d, y_i)\beta_d = c_i \quad (40)$$

*Proof.* A proof can be found in [18].

In our case the  $y_i$  and  $c_i$  are given by

$$y_i = \mu_i - x_i, c_i = - \int_{\Omega} (\mu_i - x_i) u \, dx, \quad \forall 1 \leq i \leq d, \quad (41)$$

reducing the problem to solving a linear system of equations of the size  $d \times d$  to obtain the coefficients  $\beta_1, \dots, \beta_d$  and the resulting projection is

$$\hat{m} = \sum_{i=1}^d \beta_i (\mu_i - x_i). \quad (42)$$

This means in particular, that points that have a high distance to the constraint centroid  $\mu$ , get assigned a higher value to change. The reason is the orthogonal projection that finds a projection where the smallest change in  $u$  is achieved that fulfills the constraint. Points with a high distance to  $\mu$  contribute higher values to the sum (42).

Covariance Constraint:

Similarly, we can compute the projection to the covariance constraint which yields

$$\hat{m} = \sum_{i=1}^d \sum_{j=1}^d \beta_{ij} (a_{ij} - (\mu_i - x_i)(\mu_j - x_j)), \quad (43)$$

where  $a_{ij}$  are the entries of constraint matrix  $A$ . Here we solve a  $d^2 \times d^2$  system of linear equations to obtain the coefficients  $\beta_{ij}$ . If we exploit the symmetry of  $A$  and merge terms that appear duplicate times in the sum, the number of coefficients reduces to

$$d' = \sum_{i=1}^d \sum_{j=i}^d 1 = \frac{1}{2}d(d+1), \quad (44)$$

which is still in the same complexity class  $O(d^2)$ , however reduces the number of coefficients that need to be computed by a factor of almost 2.

Moment Constraints of Arbitrary Order:

Higher order moment constraints and combinations of different order moment constraints can be computed in an analogous way.

The general projection for a moment of order  $k$  can be computed by

$$\hat{m} = \sum_{i_1=1}^d \dots \sum_{i_k=1}^d \beta_{i_1 \dots i_k} \left( a_{i_1 \dots i_k} - \prod_{l=1}^k (\mu_{il} - x_{il}) \right). \quad (45)$$

The corresponding linear equation system has  $d^k$  unknowns  $\beta_{i_1 \dots i_k}$ .

### 4.3 Optimality Bound

Unfortunately, the threshold theorem [4] guaranteeing optimality for the unconstrained binary labeling problem does not generalize to the constrained optimization problems considered here. Nevertheless, we can prove the following optimality bound:

**Proposition 6.** *Let  $u^* = \arg \min_{u \in \mathcal{C}} E(u)$  be a minimizer of the relaxed problem and  $E_{opt}$  the (unknown) minimum of the corresponding binary problem. Then any thresholded version  $\hat{u}$  of the relaxed solution  $u^*$  is within a computable bound of the optimum  $E_{opt}$ .*

*Proof.* Since  $E_{opt}$  lies energetically in between the minimum of the relaxed problem and the energy of the thresholded version, we have:

$$E(\hat{u}) - E_{opt} \leq E(\hat{u}) - E(u^*). \quad (46)$$

For all experiments in this paper this bound is on average around 5%. How to assure that the binarized version still exactly fulfills the moment constraints remains an open challenge.

## 5 Moment Constraints for Object Tracking

Object tracking over a sequence of images is the problem of finding the shape of an object that was given in the first frame in the subsequent frames of the sequence. This section shows how segmentation with moment constraints can be generalized with a few modifications to a method for object tracking.

Constraining the moments of a shape during a sequence of images leads – in combination with the presented method for image segmentation – to a method for object tracking. Given the moments of the shape in the first frame, we can constrain these moments for all subsequent frames as well.  $u$  is now a function defined on the image plane  $\Omega$  evolving over time  $T \subseteq \mathbb{R}^+$ , i.e.  $u : \Omega \times T \rightarrow [0, 1]$ .

Firstly, we will show how the area constraint can be used to track an object over time: We assume that the area of the shape should not change over time, and therefore constrain the area of a shape at time  $t$  to be equal to the area of the shape in the previous frame  $t - 1$ :

$$\min_{u \in \mathcal{B}} E(u) \quad \text{s.t.} \quad \left| \int_{\Omega} u(x,t) \, dx - c_{t-1} \right| = 0, \quad \forall t \in T, \quad (47)$$

where  $c_{t-1}$  is the area of the shape in time frame  $t - 1$  and we define  $c_{-1}$  as the area of the ellipse drawn by the user in the first frame.

Alternatively, we can allow for a small change of area and thereby model motion of the object towards or away from the camera (or the camera towards/away from the object) by replacing the  $= 0$  in equation (47) by  $\leq v$  with a constant value  $v \in \mathbb{R}$ . Here,  $v$  corresponds to the amount that the area of the shape is allowed to change from one frame to the next, i.e. depends on the velocity of the motion.

Equally, we can impose a constraint that the object should not move too far from one frame to the other: The centroid of the shape should not change more than a given value  $v$  which corresponds to the velocity of the object:

$$\min_{u \in \mathcal{B}} E(u) \quad \text{s.t.} \quad \left| \frac{\int_{\Omega} xu(x,t) \, dx}{\int_{\Omega} u(x,t) \, dx} - \mu_{t-1} \right| \leq v, \quad \forall t \in T, \quad (48)$$

where  $v$  corresponds to the maximum length of the vector between the centroid in one frame and the centroid in the next frame, and  $\mu_{t-1}$  is the centroid of  $u$  in the previous frame. Again, we define  $\mu_{-1}$  as the centroid of the ellipse drawn by the user in the first frame in order to obtain an initialization for  $t = 0$ .

Similar constraints can be imposed on the covariance and higher order moment constraints. For the covariance constraint, for example, rotation of the object can be constrained by assuming that the covariance matrix should not change more than a given value.

Optimization can be performed with the same method as explained in Section 4 while the respective moment constraints are updated in each frame.

## 6 Experimental Results

In this section we present a qualitative and quantitative evaluation of the proposed method on medical imagery and other real-world images and videos. For all experiments we use  $g(x) = 1$  and  $f(x) = \log(p_{bg}(I(x))/p_{obj}(I(x)))$  with input image  $I : \Omega \rightarrow \mathbb{R}$ . We compute the likelihoods  $p_{obj}$  and  $p_{bg}$  using color or gray-scale

histograms from inside and outside regions defined by the user input. Respective moment constraints on centroid, area or covariance structure are easily imposed by simple mouse interactions. Solutions to the constrained convex optimization problems are computed on the fly. In all experiments shown moment constraints are enforced by iteratively projecting solutions to the respective constraint sets after each iteration of optimization. Typical run-times on the GPU are around 1 second for an image of the size  $300 \times 400$ .

## ***6.1 Quantitative Evaluation on Medical Images***

We evaluate segmentations with and without moment constraints on a set of medical images and quantitatively compare the results to a manually labelled ground truth.

### **6.1.1 Area and Centroid Constraints**

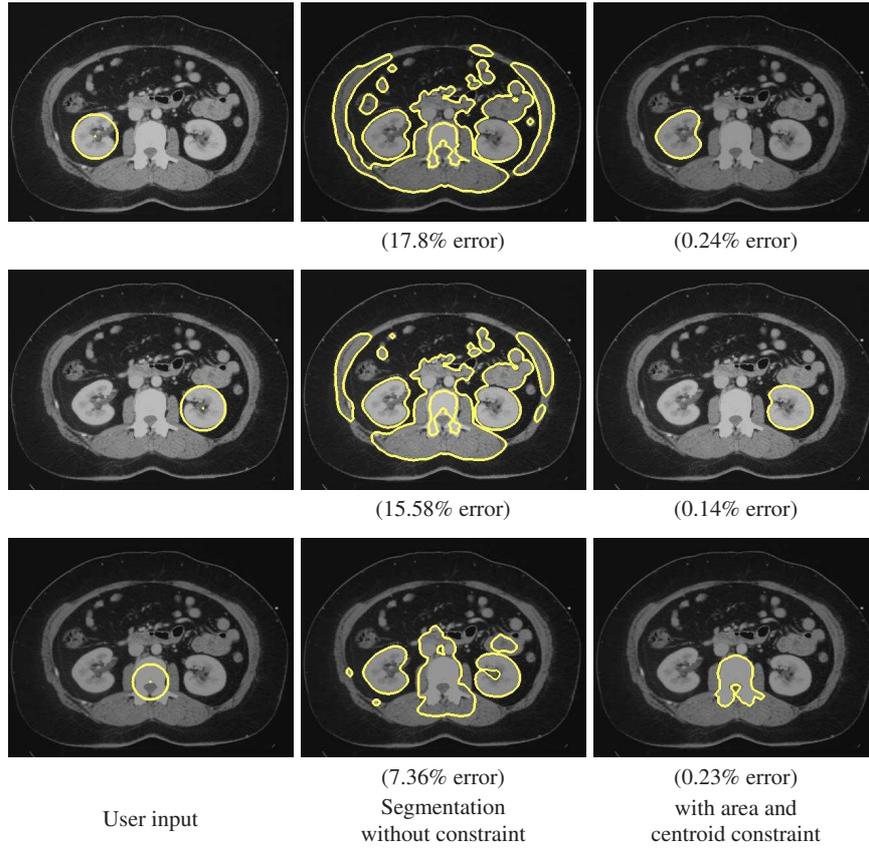
Figure 5 shows a comparison of segmentation with and without a constraint on the area and centroid for a CT image of kidneys and spine: without constraints no shape information is taken into account for the segmentation, resulting in a segmentation that includes many different regions. Enabling the area and centroid constraints leads to segmentations that prefer the center and the size of the circle that was clicked by the user. This leads to substantial improvements of the segmentations without affecting the fine-scale boundary estimation.

### **6.1.2 Higher Order Constraints**

More sophisticated structures can be specified with higher order moments. Since covariance matrices can be represented by ellipsoids, an intuitive user input is achieved by clicking an ellipse with the mouse. The axes of the ellipse define the entries of the corresponding covariance matrix, while the center and area of the ellipse define the centroid and area constraints. Figures 6 and 7 show segmentations with and without constraints resulting from user defined ellipses describing the approximated size, location and shape of the desired object.

### **6.1.3 Quantitative Performance Evaluation**

Clearly, the user-specified moment constraints allow to visibly improve the segmentation. To quantify this improvement, Table 1 shows average relative errors (i.e. the percentage of incorrectly labeled pixels per image) with standard deviations for an evaluation of the segmentation without constraint, with area constraint only, and with area and centroid and covariance constraint, respectively. Some of the images

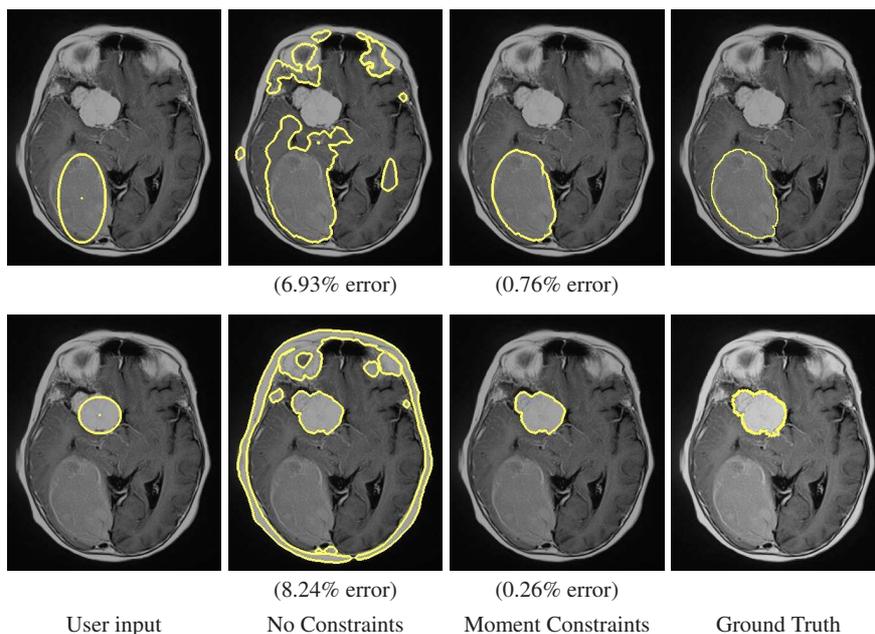


**Fig. 5** Segmentation of a CT image with kidneys and spine. The centroid and area constraints enable the user to specify the approximate location and size of the desired object that should be segmented. Imposing these moment constraints during optimization leads to drastic improvements in the segmentation.

that were used for the tests and their segmentations are shown in Figures 5, 6 and 7. The table shows that the use of these rather simple and easy to transmit constraints yield a reduction of incorrectly classified pixels by a factor of about 10.

## 6.2 Computation Times for Moment Constraint Projections

Figure 9 shows run times in seconds for projections onto the presented moment hard constraints. We computed the number of seconds that were needed on a 3.4 GHz Intel Core i7-2600 CPU to compute one projection onto all moment constraints of order 0 to  $k$  for  $k \in \{0, \dots, 30\}$  using the projection formula (45). The respective



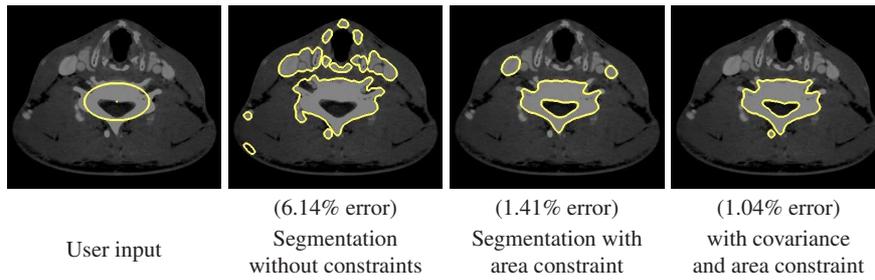
**Fig. 6** Tumor extraction in brain MR images using segmentation with and without constraints on covariance and area. While the algorithm does not require any local boundary information, constraining its second order moments by a simple user interaction suffices to generate the desired segmentation.

	Average relative error
Segmentation without constraint	12.02 % $\pm$ 0.89%
with area constraint	2.36 % $\pm$ 0.11%
with area and centroid constraint	0.41 % $\pm$ 0.05%
with area, centroid and covariance	0.35 % $\pm$ 0.09%

**Table 1** Average relative errors with standard deviations for segmentation without and with moment constraints.

constraints were obtained by computing the moments of the manually segmented reference shape shown in Fig. 3 (a). While projections onto the lower order moment constraints are computed in just a few milliseconds, the figure implies exponentially growing run time for an increasing number of constraints.

We conclude that segmentation with moment constraints is applicable to real-time tasks for the lower order moments, whereas higher order moments that can theoretically constrain arbitrary shapes, are applicable only to off-line tasks.



**Fig. 7** Segmentation without and with constraints for a CT image of the neck. Constraining the area yields a segmentation which prefers the size of the ellipse that was clicked by the user, resulting in less incorrectly labeled pixels, compared to the segmentation without constraints. The covariance constraint additionally considers the dimensions of the ellipse yielding an even more accurate segmentation. Again, the convex constraints merely constrain respective moments of the solution leading to drastic improvements of the segmentation results.

### 6.3 Segmentation of Real-World Images

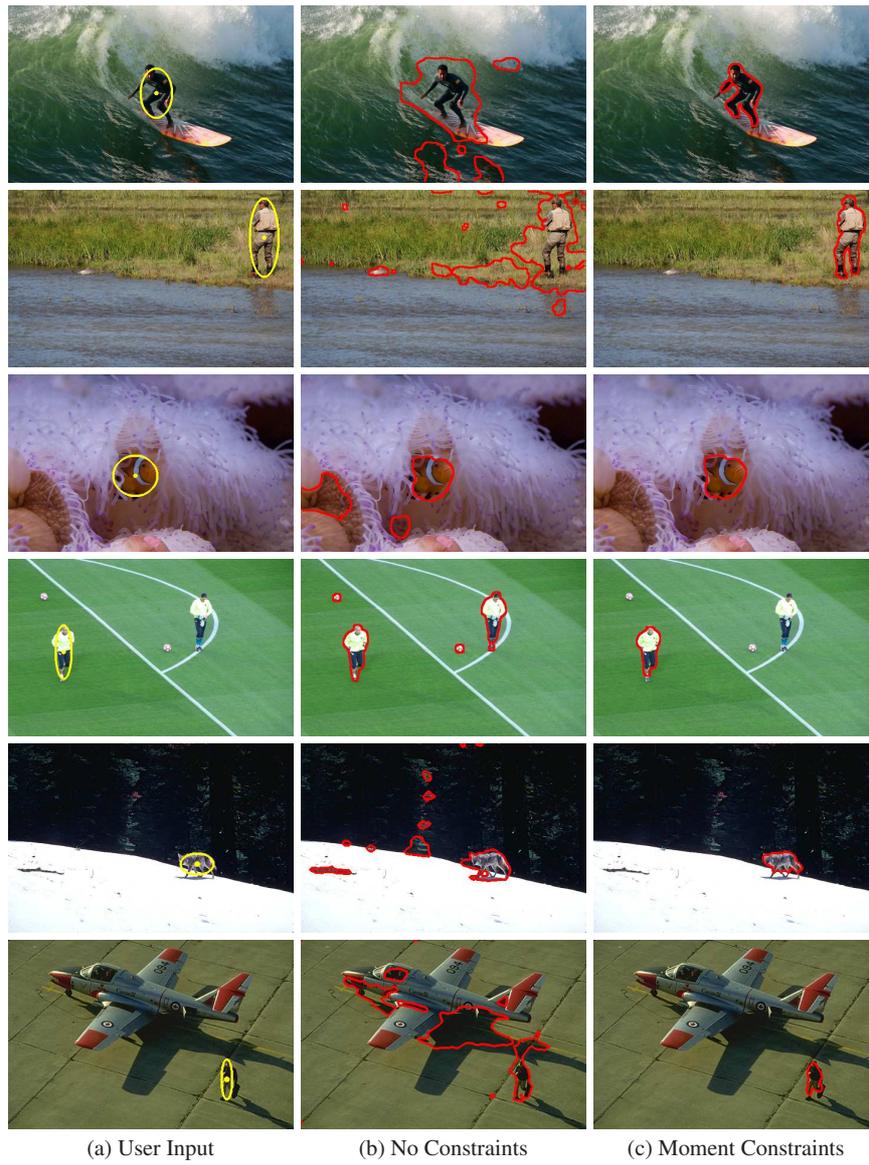
Figure 8 shows how moment constraints can improve segmentation of non-medical images. Here the data term is based on RGB histograms. The purely color-based segmentations without moment constraints shown in the second column demonstrate that the color distributions of respective objects are not sufficiently different to discriminate the objects of interest. The third column of Figure 8 shows the segmentation results with constraints on area, centroid and covariance. All moment constraints are extracted from the user-specified ellipse, allowing a deviation of 10% for each constraint to handle imprecise user input. The moment constraints allow to quickly disambiguate the color information leading to substantial improvements of the segmentation.

### 6.4 Optimality Bounds

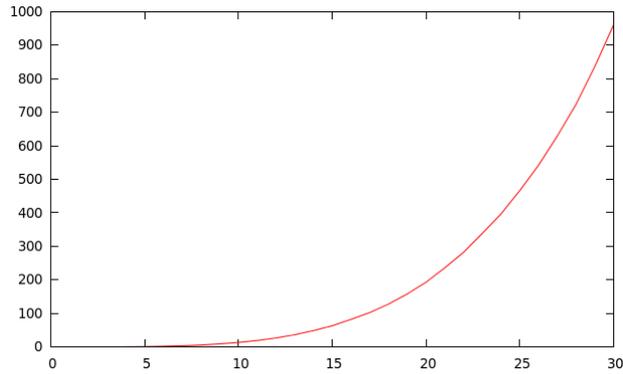
The threshold theorem [4] states that in the unconstrained case the thresholded version is a globally optimal solution of the binary energy (4). Since this is not the case when we impose additional moment constraints on the segmentation, we analysed the distance of the continuous result to the thresholded version. To this end, we compute the relative energy bound

$$\frac{E(\hat{u}) - E(u^*)}{E(u^*)}, \quad (49)$$

where  $u^*$  is the continuous solution before thresholding and  $\hat{u}$  is the solution after thresholding. Figure 10 shows segmentation results for  $u$  before thresholding. The



**Fig. 8** Segmentation of real world images without and with moment constraints. (a) The user marks an ellipse at the approximate size and location of the object with two mouse clicks. Color histograms for foreground and background are determined from the inside and outside of the ellipse, respectively. (b) Segmentation results without constraints using only the histograms. (c) Segmentation results with constraints on area, centroid and covariance. Parameters of the constraints are derived from the ellipse marked by the user.



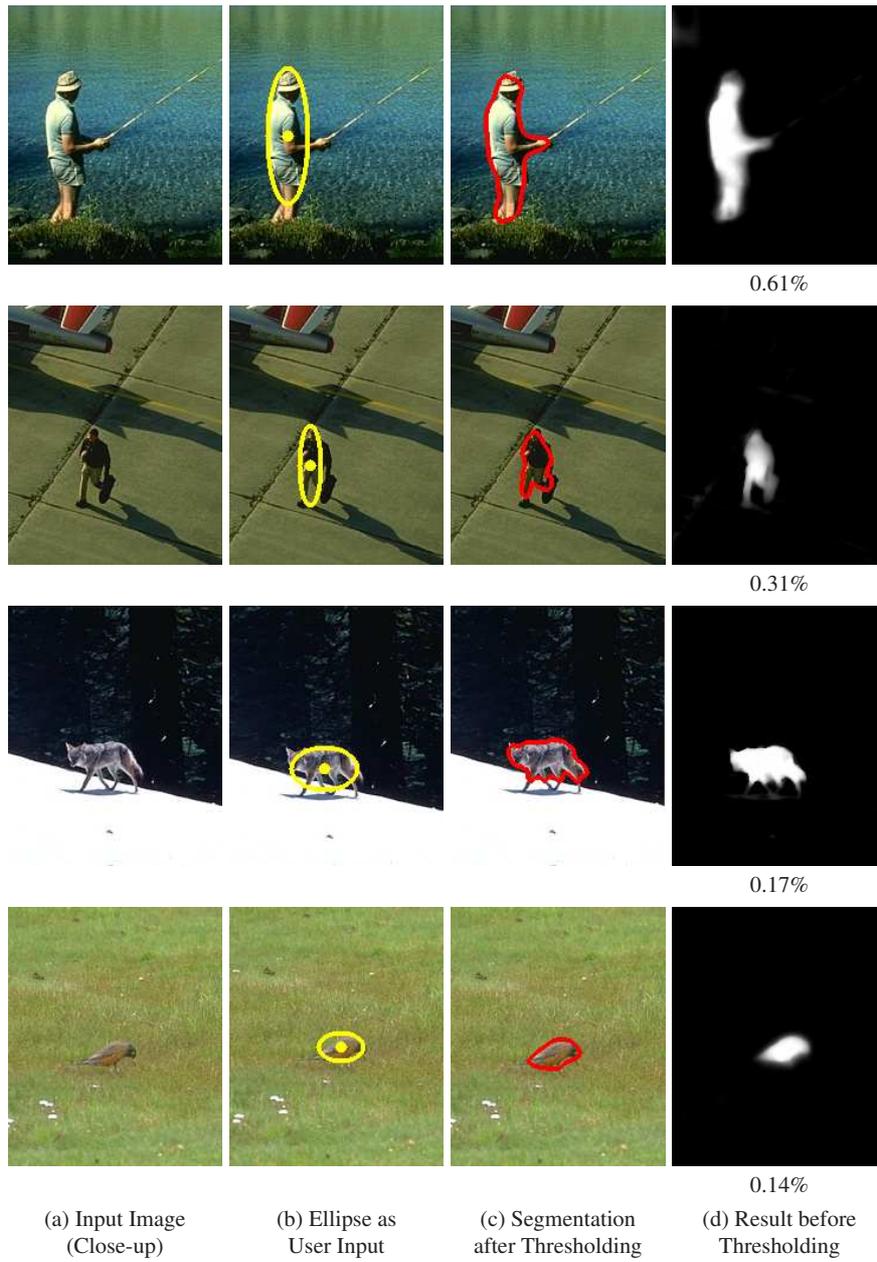
**Fig. 9** Run time (in seconds) vs. number of moment constraints  $k$  in projection. The plot shows the number of seconds needed to compute a projection onto the moments of order 0 to  $k$ . While lower order moment constraints are computed in a few milliseconds, computation times appear to grow exponentially for increasing numbers of constraints. Fortunately, for interactive segmentation applications one is mostly interested in constraining the low-order moments only, leaving the fine-scale details to be determined by the image data.

percent values below the images refers to the energy bound (49). In all four experiments that are shown in the figure the distance to the thresholded version is below 1%. In our experiments, we further observed that the more constraints are imposed on a segmentation, the larger the distance to the thresholded version increases.

### 6.5 Comparison to Segmentation with User Scribbles

Many interactive segmentation methods that utilize manual input from the user implement scribbles to obtain initial values for the histograms that generate the data term  $f$ . Scribbles in this context are arbitrary pixels marked by the user with the mouse while different labels are used for foreground and background pixels. The methods are implemented in discrete, for example [22], or continuous, for example [24], settings.

Figure 11 shows a comparison of two segmentation methods with priors on color and location from user input: first the presented method with moment constraints and second a segmentation method based on user scribbles. The purpose of this experiment was to compare the effort that needs to be brought in by the user in order to sufficiently segment the respective object of interest from the background. For the scribble segmentation the user draws mouse strokes in two different colors: blue for foreground and green for background. Histograms for the data term  $f$  are computed from these marked pixels. Segmentation results with user scribbles were computed by minimizing functional (4), using the same method and same parameters as in the case of the moment constraints, except that the moment constraints were replaced



**Fig. 10** Segmentation with the convex moment constraints converges to nearly binary solutions, making the method robust to the chosen threshold. The percentage below the images in (d) refers to the distance between the continuous result and its corresponding thresholded version.



**Fig. 11** Comparison of segmentation with moment constraints and user scribbles: Segmentation with user scribbles needs more mouse interaction to obtain similar results, since the implementation with moment constraints needs just two mouse clicks for drawing the respective ellipse.

by a constraint that all pixels that were marked as foreground by the user belong to the segmented region, and that the background pixels are not contained by the segmented region.

The figure shows that segmentations with user scribbles need much more mouse interaction compared to segmentation with moment constraints and therefore are able to substantially simplify the task of image segmentation for the user.

## 6.6 Tracking with Moment Constraints

Figures 12 and 13 show how the proposed method can be applied to tracking objects in videos. As can be seen in Figure 13, the purely color-based segmentation does not suffice to correctly segment object from background in the case of non-unique color distributions.

We impose shape information by constraining the low order moments (area, centroid and covariance) throughout the entire image sequence. As can be seen in the first image of each sequence, the user initializes the method with two mouse clicks: an ellipse of the approximate size and location of the object is drawn on the first frame of the sequence. This is sufficient user input, since histograms and moment constraint parameters are derived from the ellipse: again, histograms for foreground and background are computed from the inside and outside of the ellipse, respectively, and the constraint parameters for area, centroid and covariance are derived from the ellipse's area, center point and principal axes. The subsequent frames of the video use the histograms and moment constraints from the first frame, allowing a small deviation of the centroid from each frame to the next, which corresponds to a constraint on the maximum velocity. Since no previous learning of shapes is necessary, the approach naturally applies to arbitrary object shapes.



**Fig. 12** Moment constraints for object tracking. The user initializes the tracking by clicking an ellipse in the first frame, the moments of which constrain the segmentation in subsequent images. A small deviation of the centroid is allowed to track the moving object. Note that this approach is generic, as no shapes have to be previously learned.

## 7 Conclusion

In this chapter, we proposed the use of moment constraints in a convex shape optimization framework. In particular, we showed that for an entire family of constraints on the area, the centroid, the covariance structure of the shape and respective higher-order moments, the feasible constraint sets are all convex. While we cannot guarantee global optimality of the arising segmentations, all computed solutions are independent of initialization and within a known bound of the optimum. In both qualitative and quantitative experiments on interactive image segmentation, we demonstrated that respective moment constraints are easily imposed by the user and lead to drastic improvements of the segmentation results, reducing the average seg-



**Fig. 13** Moment constraints for object tracking. The user initializes the tracking in form of an ellipse in the first frame (left), from which histograms and constraint parameters are derived. The first row shows results with moment constraints, where a deviation of the centroid is allowed from each frame to the next one to account for the object’s motion. The second row shows results of histogram based tracking without constraints. This comparison shows that moment constraints can realize acceptable real-world object tracking with no previous learning of shapes.

mentation error from 12% to 0.35%. In contrast to existing works on shape priors in segmentation the use of low-order moment constraints does not require shape learning and is easily applied to arbitrary shapes since the recovery of fine scale shape details is not affected through the moment constraints. Efficient GPU-accelerated PDE solvers allow for computation times of about one second for images of size  $300 \times 400$ , making this a practical tool for interactive image segmentation.

## References

1. A. Blake and A. Zisserman. *Visual Reconstruction*. MIT Press, 1987.
2. J. P. Boyle and R. L. Dykstra. An method for finding projections onto the intersection of convex sets in Hilbert spaces. *Lecture Notes in Statistics*, 37:28–47, 1986.
3. V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. In *Proc. IEEE Intl. Conf. on Comp. Vis.*, pages 694–699, Boston, USA, 1995.
4. T. Chan, S. Esedoğlu, and M. Nikolova. Algorithms for finding global minimizers of image segmentation and denoising models. *SIAM Journal on Applied Mathematics*, 66(5):1632–1648, 2006.
5. D. Cremers, S. J. Osher, and S. Soatto. Kernel density estimation and intrinsic alignment for shape priors in level set segmentation. *Int. J. of Computer Vision*, 69(3):335–351, 2006.
6. P. Das, O. Veksler, V. Zavadsky, and Y. Boykov. Semiautomatic segmentation with compact shape prior. *Image and Vision Computing*, 27(1-2):206–219, 2008.
7. P. Etyngier, F. Segonne, and R. Keriven. Shape priors using manifold learning techniques. In *IEEE Int. Conf. on Computer Vision*, Rio de Janeiro, Oct 2007.
8. A. Foulonneau, P. Charbonnier, and F. Heitz. Affine-invariant geometric shape priors for region-based active contours. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 28(8):1352–1357, 2006.
9. D. M. Greig, B. T. Porteous, and A. H. Seheult. Exact maximum *a posteriori* estimation for binary images. *J. Roy. Statist. Soc., Ser. B.*, 51(2):271–279, 1989.
10. U. Grenander, Y. Chow, and D. M. Keenan. *Hands: A Pattern Theoretic Study of Biological Shapes*. Springer, New York, 1991.

11. E. Ising. Beitrag zur Theorie des Ferromagnetismus. *Zeitschrift für Physik*, 23:253–258, 1925.
12. M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *Int. J. of Computer Vision*, 1(4):321–331, 1988.
13. S. Kichenassamy, A. Kumar, P. J. Olver, A. Tannenbaum, and A. J. Yezzi. Gradient flows and geometric active contour models. In *IEEE Int. Conf. on Computer Vision*, pages 810–815, 1995.
14. M. Klodt and D. Cremers. A convex framework for image segmentation with moment constraints. In *IEEE Int. Conf. on Computer Vision*, Barcelona, Spain, 2011.
15. K. Kolev and D. Cremers. Integration of multiview stereo and silhouettes via convex functionals on convex domains. In *European Conference on Computer Vision (ECCV)*, Marseille, France, October 2008.
16. K. Kolev, M. Klodt, T. Brox, and D. Cremers. Continuous global optimization in multiview 3d reconstruction. *International Journal of Computer Vision*, 2009.
17. V. Lempitsky, P. Kohli, C. Rother, and T. Sharp. Image segmentation with a bounding box prior. In *IEEE Int. Conf. on Computer Vision*, Kyoto, Japan, 2009.
18. David G. Luenberger. *Optimization by Vector Space Methods*. John Wiley & Sons, Inc., New York, NY, USA, 1st edition, 1997.
19. D. Mumford and J. Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Comm. Pure Appl. Math.*, 42:577–685, 1989.
20. S. J. Osher and J. A. Sethian. Fronts propagation with curvature dependent speed: Algorithms based on Hamilton–Jacobi formulations. *J. of Comp. Phys.*, 79:12–49, 1988.
21. A. Papoulis and S. U. Pillai. *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill, New York, 4th edition edition, 2002.
22. C. Rother, V. Kolmogorov, and A. Blake. GrabCut: interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph.*, 23(3):309–314, 2004.
23. T. Schoenemann and D. Cremers. A combinatorial solution for model-based image segmentation and real-time tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2009.
24. M. Unger, T. Pock, D. Cremers, and H. Bischof. Tvseg - interactive total variation based image segmentation. In *British Machine Vision Conference (BMVC)*, Leeds, UK, September 2008.
25. O. Veksler. Star shape prior for graph-cut image segmentation. In *Europ. Conf. on Computer Vision*, pages 454–467, 2008.