A Continuous Shape Prior for MRF-Based Segmentation

Dmitrij Schlesinger

Dresden University of Technology

Abstract. The competition between discrete (MRF based) and continuous (PDE based) formulations has a very long history, especially in context of segmentation. Obviously, both have their advantages and drawbacks. Therefore the choice of a discrete or continuous framework is often driven by a particular application or (even more often) by personal preferences of a particular researcher. In this work we present a model for binary segmentation, where discrete and continuous parts are combined in a well founded and simple way. We discuss the properties of the proposed model, give a basic inference algorithm and validate it on a benchmark database.

1 Introduction

The discussion about properties, advantages and drawbacks of discrete and continuous approaches is not new (see e.g. [1]). In most cases however the questions are posed in a quite ultimate way: "which framework is better" with respect to a particular property (precision, modeling capabilities, computational efficiency etc.). In this work we try to combine the advantages of continuous and discrete approaches for segmentation. To start with, we would like to discuss/recall some relevant properties. The first issue is the modeling. In a continuous framework it is much easier to express a-priory assumptions that relate to low-level features, like e.g. boundary length, curvature etc. The main drawback of discrete methods in this respect is the presence of well known metrication artifacts. Of course, it is possible to avoid these effects (to some extent) in the discrete framework too [2,3]. However, it leads to higher-order MRF-s or to a complicate graph structure, that makes the modeling less transparent and the related tasks hard to optimize. On the other hand, some semi-global properties (such as e.g. scene layout [4] or star-convexity [5]) is more convenient to express in the discrete framework. Another important related property here is the ability to deal with multi-label segmentation. Again, this is possible in continuous approaches as well [6]. However, it leads to the complex coupling constraints that is again hard to optimize.

An important topic is the ability to model shapes that is nowadays almost obligatory for segmentation (especially in the unsupervised case). Many different shape priors were studied in the continuous optimization community in the past. The most elaborated (in our opinion) techniques are based on the Level Set representation (see e.g. [7] and references therein). In the discrete domain this topic is not as elaborated so far. In [8] it is shown that it is in principle possible in part-shape based segmentation. However, again the resulting model (although of second order only) has a very complex neighborhood structure and a lot of free parameters, that restricts the applicability.

A. Heyden et al. (Eds.): EMMCVPR 2013, LNCS 8081, pp. 350–361, 2013.

Another branch of questions deals with inference. The continuous functionals used for inference are often convex and can be optimized globally. Even if not, convex approximations/relaxations can be used. In the discrete case most of inference tasks are NP-complete. In particular, if the Maximum A-posteriori decision is employed as the decision strategy, relaxations are often used for approximations, thus in fact, discrete optimizations are substituted by corresponding continuous ones.

The main drawback of continuous methods is the lack of sound statistic interpretation, which makes it impossible to learn unknown parameters. For discrete models in contrast, the learning can be posed in a well founded statistic way using e.g. the Maximum Likelihood principle or applying the ideas of discriminative learning, such as e.g. Structural SVM [9].

The main aim of the paper is to show that the advantages of discrete and continuous models can be combined in a well founded and simple way. This work is inspired by [10,11] at most. In these works however, a very simple shape prior is used, which is obviously not suitable for real segmentation tasks. We use a more general class of Level Set like functions to represent shapes. In short, the proposed model consists of two parts – a discrete and a continuous one. The former is a standard MRF consisting of the Ising model for the a-priori probability distribution for segmentations and a Gaussian Mixture Model for appearances (similar to e.g. [12,13]). Like in [10], the prior probability distribution is parametrized by a Level Set like function, which assigns additional unary potentials to each pixel. The proposed model has the following properties that are desired for segmentation:

- It is very simple and generic and has few free parameters.
- The modeling of the low-level features is expressed in a transparent way using continuous framework.
- The segmentation is posed in a sound statistic way using discrete formulation.
- Basic algorithms for inference and learning are based on standard techniques widely used in discrete and continuous optimization respectively.
- The method is able to work in fully unsupervised manner and gives promising results.

2 Approach

2.1 Model

In this work we consider binary segmentation for simplicity although all consideration can be easily generalized to the multi-label case. The model architecture is illustrated in Fig. 1. Let $G = (R, E)$ be a graph over the pixel grid, i.e. each node $r \in R$ corresponds to a pixel, the set of edges E corresponds to the 4-neighborhood structure. At the same time the nodes are embedded in a continuous space $\Omega \subset \mathbb{R}^2$, i.e. each pixel has its coordinates, which we denote by $r \in \Omega$ as well to omit notational clutter. The segmentation is a mapping $y : R \to \{0, 1\}$ that assigns a label (0 for the background

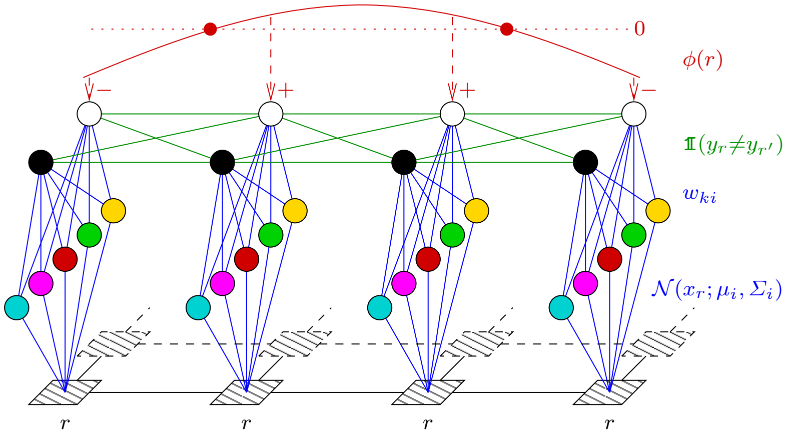


Fig. 1. Model overview (best viewed in color). The shape function $\phi(r)$ (red) assigns additional unary potentials to the foreground labels (white) at each position of the underlying MRF (green). The conditional probability distributions $p(x_r|y_r)$ for observations are Gaussian Mixtures (blue) with a common set of Gaussians (colored circles).

and 1 for the foreground) to each node r. An image is also a mapping $x : R \to C$ that assigns a color value $c \in C$ to each pixel. We denote by y_r and x_r the label and the color value in pixel r correspondingly. The probability distribution for pairs (x, y) is defined by[1]

$$\begin{aligned} p(x,y;\phi) &= p(y;\phi)\cdot p(x|y) = \\ &= \frac{1}{Z(\phi)} \exp\Big[-\alpha \sum_{rr'\in E} 1\!\!1(y_r \neq y_{r'}) + \sum_r y_r \cdot \phi(r)\Big] \cdot \prod_{r\in R} p(x_r|y_r). \end{aligned} \qquad (1)$$

The prior probability distribution consists of two parts (energy terms). The first one is the Ising model that penalizes different labels in neighboring pixels by a penalty $\alpha > 0$. The second one is a shape function $\phi : \Omega \to \mathbb{R}$, which supports/suppresses the foreground label in each pixel by assignment an additional unary energy term to it.

As usual for MRF-s, the conditional probability distribution $p(x|y)$ is assumed to be conditionally independent. We use Gaussian Mixture models with common Gaussians for $p(x_r|y_r)$ like in [11]. In contrast however, we use general multivariate Gaussians instead of the isotropic ones as we are not interested in real-time performance. To summarize, the probability to observe a color $c \in C$ for a label $k \in \{0, 1\}$ is

$$p(c|k) = \sum_i w_{ki} \cdot \mathcal{N}(c;\mu_i,\Sigma_i) \qquad (2)$$

with the label-specific weights w_{ki} and Gaussians $\mathcal{N}(\cdot)$ with mean values μ_i and covariance matrices Σ_i.

[1] The parameters are separated from the random variables by semicolon.

2.2 Inference and Learning

Assuming for a moment that the shape function is known, the segmentation is posed as a Bayesian decision task with the Hamming distance as the loss. It leads to the maximum marginal decision strategy

$$y_r^* = \arg\max_{k\in\{0,1\}} p(y_r{=}k|x;\phi) \quad \forall r \in R. \tag{3}$$

Although most successful segmentation methods are based on the maximum a-posteriori decisions, marginal based inference is becoming increasingly popular for MRF-based approaches due to several reasons (see e.g. [14]). First of all, it follows from a more reasonable loss function. Besides, as we will see later, we need marginals for other tasks as well, namely for the shape estimation and unsupervised learning of the appearance characteristics.

The most interesting part is the estimation of the shape function ϕ. We consider it primarily as a continuous function in order to be able to use elaborated techniques from continuous optimization. Consequently, we *do not* consider ϕ as a random variable, mainly because it is not possible to introduce probability measures for function spaces. Therefore, in our model the shape function is a parameter of the probability distribution. Remember that we are mainly interested in fully unsupervised segmentation. Taking all this into account, a reasonable choice is to estimate the shape function e.g. according to the Maximum Likelihood principle, i.e. $\ln p(x;\phi) \to \max_\phi$. However, it is easy to see that such a formulation for (1) has a trivial solution. For those pixels that would be assigned to the foreground according to the appearance model only (i.e. $p(x_r|1) > p(x_r|0)$) the optimum with respect to ϕ is reached at $\phi(r) = \infty$, for other pixels $\phi(r) = -\infty$ holds. It is indeed expected because we did not introduce any requirements (prior assumptions etc.) for shape so far.

It is a common technique in Machine Learning to enhance an objective function (e.g. the Likelihood in our case) by a *regularizer*. In Machine Learning this trick is used mainly in order to resolve ambiguities and/or increase robustness of the learning. Here we use the same ideas in order to be able to express our prior assumptions about the shape. Hence, we pose the shape estimation as the following optimization task:

$$F(\phi) = \ln\Big[\sum_y p(x,y;\phi)\Big] + \mathcal{R}(\phi) \to \max_\phi, \tag{4}$$

where the first addend is the log-likelihood and the second one is a regularizer. Numerous choices are possible for the latter. We use a simple one to keep the whole approach as general as possible:

$$\mathcal{R}(\phi) = -\int_\Omega \big(\lambda_1 \cdot \|\nabla\phi\|^2 + \lambda_2 \cdot \triangle\phi^2\big)d\omega, \tag{5}$$

First of all we would like to require that the shape function is smooth. At the same time penalizing gradients only is not appropriate in our context because in this case the shape function tends to be “flat” inside the segments. As the consequence, it does not influence the underlying MRF good enough. Therefore we penalize both gradients and Laplacians to facilitate the shape function to be “as linear as possible”.

The derivative of the subject (4) with respect to the value of the shape function at a given position r yields[2]

$$\frac{\partial F}{\partial \phi(r)} = p(y_r{=}1|x;\phi) - p(y_r{=}1;\phi) + \frac{\partial \mathcal{R}(\phi)}{\partial \phi(r)}. \tag{6}$$

The first term is the posterior marginal probability for the pixel r to be the foreground. These probabilities are estimated using Gibbs Sampling. For the second one (prior marginal probabilities) we use the following approximation. In [11] it is proposed just to binarize the shape function at zero level and substitute real prior marginals by $p(y_r{=}1;\phi) \approx \mathbb{1}(\phi(r) > 0)$. In our case it does not work well, because the shape is much "weaker" as in [11] and non-parametric. However, similar observations can be used. We performed a couple of experiments and found out that the prior marginals can be well approximated (for reasonable values of α, λ_1 and λ_2) by a sigmoid function

$$p(y_r{=}1;\phi) \approx \frac{\exp\bigl(\beta\phi(r)\bigr)}{\exp\bigl(\beta\phi(r)\bigr) + 1}, \tag{7}$$

with a constant $\beta > 1$, which makes the sigmoid function stronger compared to the independent case $\alpha = 0$. The last term in (6) is the Gâteaux derivative of (5) and is obtained (after discretization) by convolution with the corresponding mask.

For unsupervised learning of the Gaussian mixtures (2) we follow standard techniques and use the Expectation-Maximization Algorithm. Note that in the Expectation step again the marginal posterior label probabilities should be computed.

2.3 Discussion

At this place we would like to discuss some properties of our model and relations to other ones. The discrete part is a standard MRF. Without the shape prior it is even simpler as e.g. GrabCut [13] due to the constant edge strengths α. Obviously, the discrete part can be extended to admit more elaborated features. At this stage however we do not follow this way because we would like to keep the model as simple as possible and therefore as generic as possible. In particular, we would like to show that even such simple MRF is able to produce reasonable results if it is "weakly supported" by other parts (e.g. by a continuous shape prior in our case). Nonetheless, we are mainly interested in an unsupervised segmentation that requires generative models.

If we simplify our model in another way doing the discrete part independent (i.e. $\alpha = 0$), we observe the following analogy. As the log-likelihood in (4) can be computed explicitly, the model becomes a continuous one, where the data terms are obtained by marginalization over labels in each pixel. The optimization (4) reads then

$$\sum_r \ln \left[\frac{\exp(\phi(r))}{\exp(\phi(r)) + 1} \cdot p(x_r|1) + \frac{1}{\exp(\phi(r)) + 1} \cdot p(x_r|0) \right] + \mathcal{R}(\phi) \to \max_{\phi}. \tag{8}$$

Suppose, the shape function is very "strong" (which is mainly the case in continuous optimization, as Level Set functions are supposed to be close to the distance transform),

[2] We omit detailed derivation here, because they are standard and quite straightforward.

i.e. the sigmoid function in (8) is very close to the Heaviside function $H(\phi(r)) = \mathbb{1}(\phi(r)>0)$. In other words, we substitute the marginalization by binarization. Then the model becomes a standard Level Set based continuous model for segmentation

$$\sum_r \Big[H(\phi(r)) \cdot \ln p(x_r|1) + \big(1 - H(\phi(r))\big) \cdot \ln p(x_r|0)\Big] + \mathcal{R}(\phi) \to \max_\phi . \tag{9}$$

We would like to stress that the optimization (4)-(6) is in fact a standard task extensively studied in the continuous optimization in the past. The difference is only that the data terms are not given explicitly but their derivatives (first two summands in (6)). Therefore the whole spectrum of methods (like e.g. successive over-relaxation, scale-space methods etc.) can be used for efficient optimization. Moreover, elaborated regularizers can be of course used instead of the simple one (5), such as e.g. TV-norm.

2.4 Implementation Details

We implemented the method in a multi-threaded manner like in [11][3]. The overall system architecture is given in Fig. 2. Basically, it consists of three blocks that work in parallel. The core of the algorithm is the “Gibbs Sampling” block that permanently computes marginal posterior label probabilities for the current appearances and shape function. The “Appearances” block observes these probabilities, performs maximization steps of the EM-Algorithm and computes the unary potentials $\ln p(x_r|k)$ according to (2). The “Shape” block also takes the marginal posterior probabilities as the input and performs gradient steps according to (6). The output of this block is the shape function $\phi(r)$ (i.e. the shape dependent unary potentials in (1)).

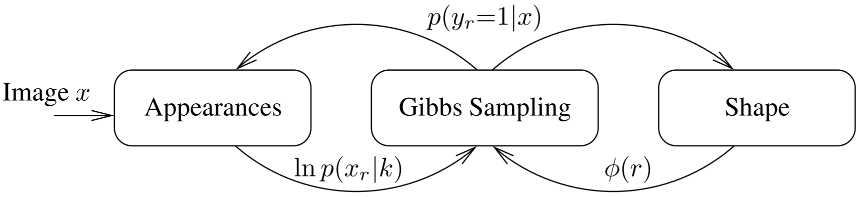


Fig. 2. The system architecture

We do not use any stopping criteria. In our experiments we just limit the time for the system to work. After the time is elapsed, the final segmentation is obtained by thresholding the current marginal label probabilities at 0.5 level.

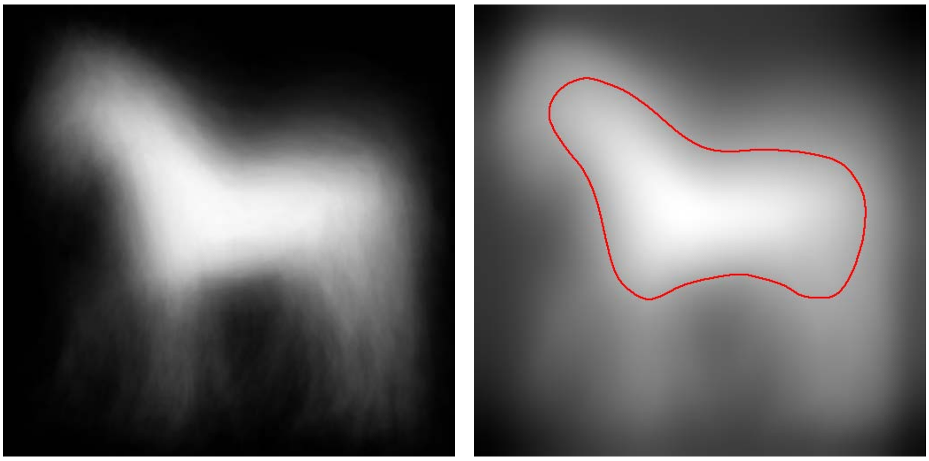
The most slow procedure is the learning of appearances. We did not use acceleration techniques as in [11] in order to avoid approximations. Hence, for each pixel the marginalization over the Gaussians is necessary that involves exponentiation and taking the logarithm many times. The fastest procedure is Gibbs Sampling that can be easily implemented in a very efficient manner. The speed of the shape estimation lies in between. To summarize, for a typical run of 5 minutes the system was able to perform

[3] The source code will be available soon at [23].

about 33 thousands iterations (scans over the whole image) of Gibbs Sampling, about 7 thousands gradient steps for the shape learning and only about 250 iterations (M-steps) for appearances. Obviously the computational efficiency can be essentially improved. We plan to address this issue in the nearest future.

3 Experiments

We validate our model on the Weizmann Horse Database [15]. First of all we would like to note that our simple generic shape prior does not fit "horses" good enough. The obvious problems are legs, pigtails and manes. The next topic is the following. As we perform fully unsurprised segmentation, the problem is highly ill-posed, i.e. there can be many objects in the scene (despite of horses) that are "compact" and fit well to our shape prior. At the same time the optimization in (4)-(6) is non-convex due to the likelihood. Therefore we need a reasonable initialization for the shape function. We obtained it by just considering the position dependent probability distributions of labels, estimated from ground truths over the whole database. For each pixel we count how many times it is labeled as foreground according to the ground truth (to do this, images were rescaled to a predefined size). The probability map obtained in such a manner is given in Fig. 3(a). After that we fit the shape function into these probabilities, i.e. we perform optimization (4)-(6), where the posterior marginals are substituted by the probability map obtained as described above. The resulting initial shape function is shown in Fig. 3(b). The initialization for each image is obtained by rescaling this initial shape function to the original image size.



(a) Probability map. **(b)** Shape function (zero-level is shown in red).

Fig. 3. Initialization for the shape function (scaled to fit into the gray-value range)

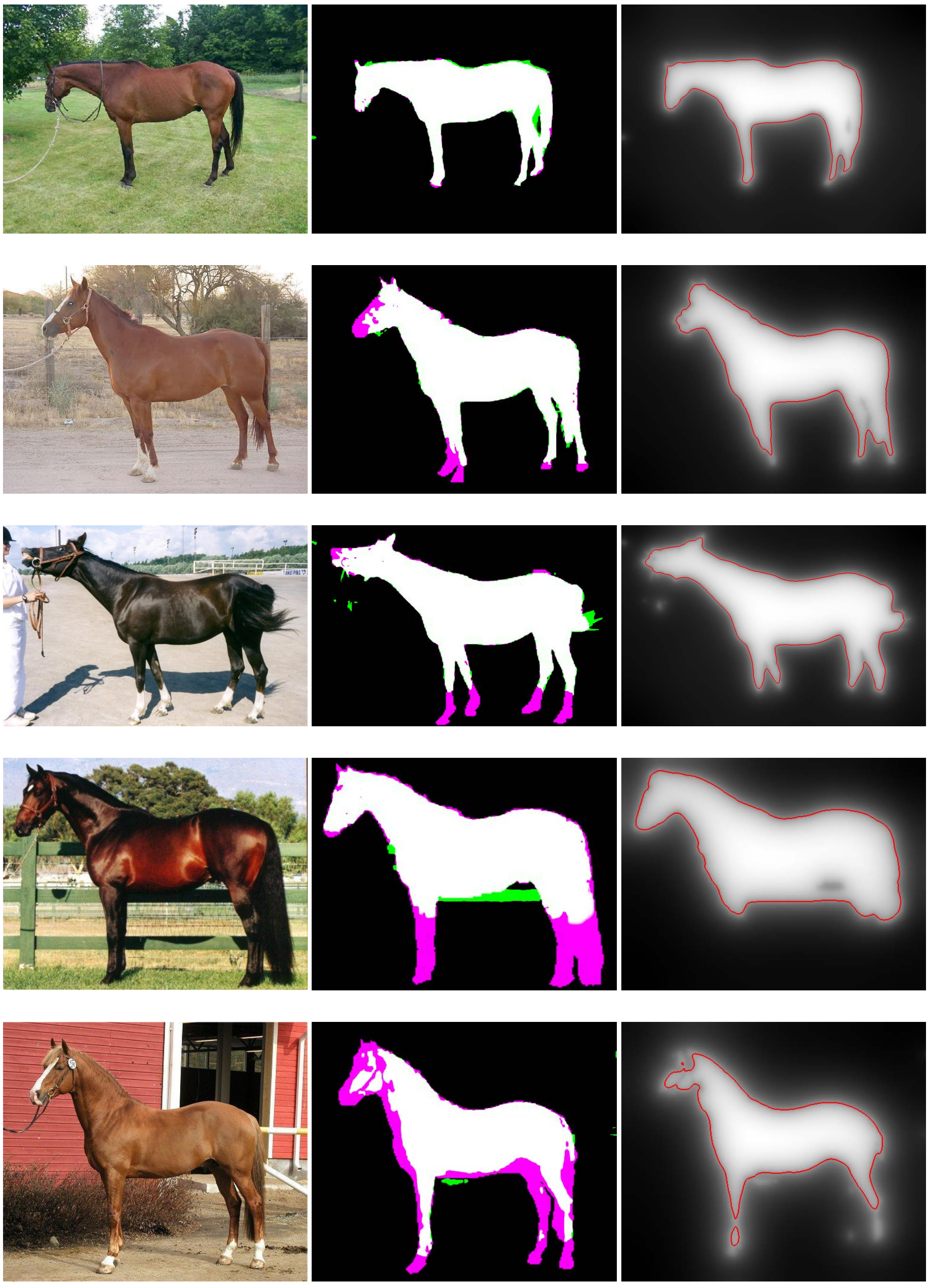


Fig. 4. Some selected results. Left: original images; middle: obtained segmentation overlaid with ground truth (green – false positive, magenta – false negative), right: obtained shape function, zero level is shown in red. Best viewed in color.

In Fig. 4 some selected results are presented. If the coloring properties are distinctive enough, the produced segmentation is almost perfect (the first row, less that 1% misclassified pixels). There are also many "good" segmentations (like in the second and the third rows, about 2-3% misclassified pixels), where the errors have a local nature. Note, that isolated segmentation errors (see the third row) are often not presented in the zero-level set of the found shape function. Obviously, if a particular task consists of the retrieval of compact connected segments that need not necessarily be precise, the zero-level can serve as the segmentation result.

In most cases the method found segmentations that we would consider rather as "satisfactory" ones (fourth and fifth rows) – characteristic horse shapes are clearly visible at most, the obtained segmentations are however not precise enough (about 8-9% misclassification). Finally, there are cases, for which the method does not work at all (see Fig. 5), i.e. it produces segmentations that have nothing in common with what is desired.

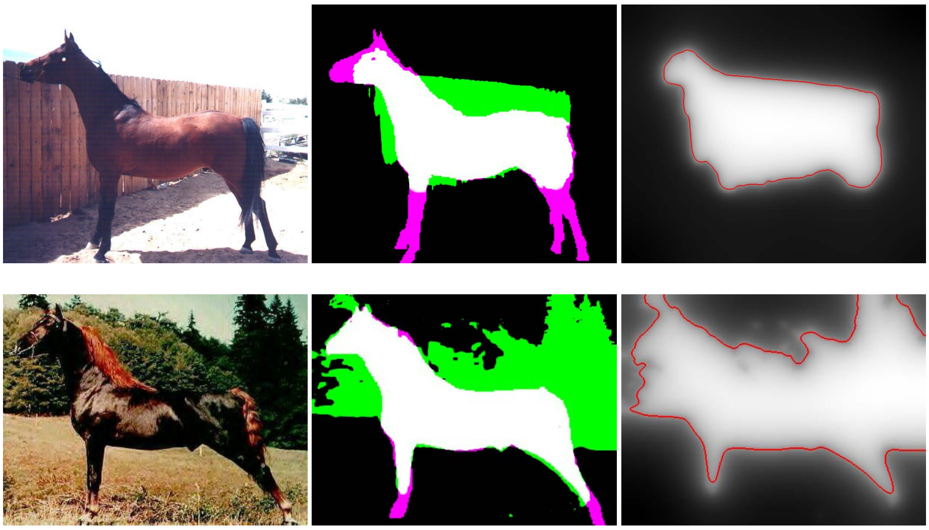
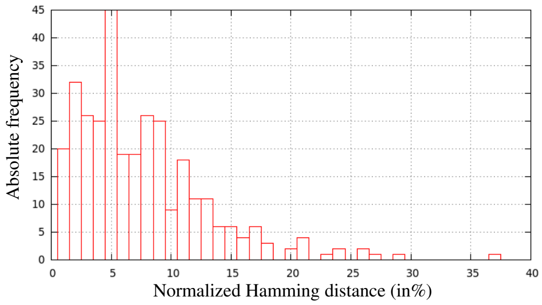


Fig. 5. Fail cases (see description for Fig. 4)

The quantitative results are presented in Fig. 6(a). For each image we compute the Hamming distance normalized to the image size. We prefer to give a complete histogram of the Hamming distances to give a better feeling, how many images in the database were segmented correctly. As it is seen from the figure there were only 12 images (out of 328 in total), where more than 20% of pixels were misclassified. 237 images were segmented with less than 10% of inaccuracy. The most frequent error value (for 45 images) is 5%. The average normalized Hamming distance is 7.8%.

Unfortunately, it is not easy to compare the proposed model with other ones in a straightforward manner. The obvious reason is that there are many algorithms in the literature that are very complex, too different and therefore not comparable. Elaborated segmentation techniques usually use high-level knowledge about the object to be segmented. For example in [16,17,18] template based techniques are used, where the templates are learned in advance and match characteristic shape fragments during inference. In [19] a complex hierarchic model is presented. In [20] a kernelized structural SVM learning framework is employed. We would like also to mention [14], where relatively simple generic features (although a lot of them) are used in a CRF framework and marginal based inference and learning are exploited. On this background our model looks like an extreme oversimplification. On the other hand, really simple generic algorithms seem to be not able to cope with real segmentation tasks. So for example GrabCut initialized in a reasonable manner has only 85.5% accuracy[4] (data from [20]). To conclude, our 92.2% accuracy were really surprising. We summarize the results found in the literature in Fig. 6(b).

We should admit that at the moment the used optimization techniques (as well as our implementation) are very far from being computationally efficient. Processing of one image takes about 5 minutes on a standard quad-core computer thus the processing of the whole database takes about a day. As a consequence we were not able to tune free parameters of the method carefully. We performed a couple of experiments for only 3-4 images taken randomly from the database and decided for values that seemed reasonable. We also would like to note that for many images 5 minutes were obviously not enough, i.e. the method had not enough time to converge. Consequently, we hope that the segmentation accuracy can be further improved by using efficient implementation.



(a) Histogram of the normalized Hamming distances.

Approach	Accuracy
ObjCut [16]	96.0
Levin [17]	95.5
Borenstein [18]	93.6
Zhang [19]	95.4
Bertelli [20]	94.3
Domke [14]	92.0
Our	92.2
GrabCut [13]	85.5
Co-segmentation [21]	80.1
MNcut [22]	51.0

(b) Comparison to other approaches.

Fig. 6. Quantitative evaluation

Looking at the initialization (see Fig. 3) it might be well asked, whether our results are caused by the proposed model or by a suitable chosen initialization. In order to verify that the model really does its job, we performed the following check. We binarized the initial shape function at the zero-level and used this binary mask as a segmentation result. For each image we compute the normalized Hamming distance between the corresponding ground truth and the binary mask and average over the database. The obtained accuracy was 85.1%. Of course, a reasonable initialization is necessary for the

[4] To be in line with other sources we summarize the results in term of segmentation accuracy, i.e. the percentage of correctly classified pixels.

method. However, it is able to significantly change the initialization and considerably improve the segmentation accuracy.

It is also interesting to see, what are influences of the different model parts to the results. To investigate this we performed two additional tests – one without the shape function at all (i.e. $\phi \equiv 0$) and another one with independent MRF (i.e. $\alpha = 0$). Without the shape the model almost always produces a reasonable result – more or less compact segments, which have however nothing in common with horses. The accuracy in this case was about 83%. The situation without the Ising prior is slightly better – about 85%. In this case horses can be seen in obtained segmentations, which are however very noisy.

4 Conclusion

In this work we presented a model for segmentation, which consists of two parts: a discrete MRF and a continuous shape prior. We show that a proper combination of seemingly very different frameworks leads to promising results remaining at the same time quite simple. We gave a basic algorithm for inference that again consists of two parts, each one being standard in the corresponding framework. To conclude: discrete and continuous methods need not compete, they should rather work together.

There are numerous ways for the further research. The main motivation of the work was the intention to combine advantages of different methods. In particular we use a continuous shape prior in order to be able to use elaborated techniques from continuous optimization. In this work however we applied only a very basic method that is extremely inefficient in practice. Therefore this paper is rather a “work in progress”. Other choices for the continuous part are to be evaluated.

Our treatment of “inference” (see sec. 2.2) differs from a commonly used one. Usually, *parameters* of a probability distribution are not dependent on a particular observation, i.e. they represent some intrinsic properties. They should be learned on a training dataset and used (remaining thereby unchanged) during inference. If something is image-dependent, then it is rather a *random variable*. In our model however, it is hardly possible to interpret the shape function as a random variable because of the infinite dimensionality. Therefore it is also not possible to consider the regularizer (5) as the prior probability distribution of shapes. Hence, we are obligated to consider it as a parameter. We plan to investigate the related questions more carefully in the future.

References

1. Cremers, D., Pock, T., Kolev, K., Chambolle, A.: Convex relaxation techniques for segmentation, stereo and multiview reconstruction. In: Markov Random Fields for Vision and Image Processing. MIT Press (2011)
2. Shekhovtsov, A., Kohli, P., Rother, C.: Curvature prior for MRF-based segmentation and shape inpainting. In: Pinz, A., Pock, T., Bischof, H., Leberl, F. (eds.) DAGM and OAGM 2012. LNCS, vol. 7476, pp. 41–51. Springer, Heidelberg (2012)
3. Schoenemann, T., Kahl, F., Masnou, S., Cremers, D.: A linear framework for region-based image segmentation and inpainting involving curvature penalization. International Journal of Computer Vision 99(1), 53–68 (2012)

4. Hoiem, D., Efros, A.A., Hebert, M.: Recovering surface layout from an image. IJCV 75(1), 151–172 (2007)
5. Veksler, O.: Star shape prior for graph-cut image segmentation. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part III. LNCS, vol. 5304, pp. 454–467. Springer, Heidelberg (2008)
6. Cremers, D., Sochen, N., Schnörr, C.: A multiphase dynamic labeling model for variational recognition-driven image segmentation. IJCV 66(1), 67–81 (2006)
7. Cremers, D., Osher, S.J., Soatto, S.: Kernel density estimation and intrinsic alignment for shape priors in level set segmentation. International Journal of Computer Vision 69, 335–351 (2006)
8. Flach, B., Schlesinger, D.: Modelling composite shapes by gibbs random fields. In: CVPR, pp. 2177–2182 (2011)
9. Nowozin, S., Lampert, C.: Structured Learning and Prediction in Computer Vision. Foundations and Trends in Computer Graphics and Vision, vol. 6 (2010)
10. Flach, B., Schlesinger, D.: Combining shape priors and MRF-segmentation. In: da Vitoria Lobo, N., Kasparis, T., Roli, F., Kwok, J.T., Georgiopoulos, M., Anagnostopoulos, G.C., Loog, M. (eds.) S+SSPR 2008. LNCS, vol. 5342, pp. 177–186. Springer, Heidelberg (2008)
11. Schlesinger, D.: A real-time MRF based approach for binary segmentation. In: Pinz, A., Pock, T., Bischof, H., Leberl, F. (eds.) DAGM and OAGM 2012. LNCS, vol. 7476, pp. 387–396. Springer, Heidelberg (2012)
12. Boykov, Y., Jolly, M.P.: Interactive graph cuts for optimal boundary & region segmentation of objects in n-d images. In: ICCV, vol. 1, pp. 105–112 (2001)
13. Rother, C., Kolmogorov, V., Blake, A.: “GrabCut": interactive foreground extraction using iterated graph cuts. ACM Trans. Graph. 23(3), 309–314 (2004)
14. Domke, J.: Learning graphical model parameters with approximate marginal inference. CoRR abs/1301.3193 (2013)
15. `http://www.msri.org/people/members/eranb`
16. Kumar, M.P., Torr, P.H.S., Zisserman, A.: Objcut: Efficient segmentation using top-down and bottom-up cues. IEEE Trans. Pattern Anal. Mach. Intell. 32(3), 530–545 (2010)
17. Levin, A., Weiss, Y.: Learning to combine bottom-up and top-down segmentation. Int. J. Comput. Vision 81(1), 105–118 (2009)
18. Borenstein, E., Malik, J.: Shape guided object segmentation. In: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 969–976 (June 2006)
19. Zhang, L., Ji, Q.: Image segmentation with a unified graphical model. IEEE Trans. Pattern Anal. Mach. Intell. 32(8), 1406–1425 (2010)
20. Bertelli, L., Yu, T., Vu, D., Gokturk, B.: Kernelized structural svm learning for supervised object segmentation. In: CVPR, pp. 2153–2160 (2011)
21. Joulin, A., Bach, F., Ponce, J.: Discriminative clustering for image co-segmentation. In: 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1943–1950 (June 2010)
22. Cour, T., Benezit, F., Shi, J.: Spectral segmentation with multiscale graph decomposition. In: Computer Vision and Pattern Recognition (CVPR), vol. 2, pp. 1124–1131. IEEE (2005)
23. `http://wwwpub.zih.tu-dresden.de/~ds24/rtsegm2/rtsegm2.html`