A Real-Time MRF Based Approach for Binary Segmentation

Dmitrij Schlesinger

Dresden University of Technology

Abstract. We present an MRF based approach for binary segmentation that is able to work in real time. As we are interested in processing of live video streams, fully unsupervised learning schemes are necessary. Therefore, we use generative models. Unlike many existing methods that use Energy Minimization techniques, we employ max-marginal decision. It leads to sampling algorithms that can be implemented for the proposed model in a very efficient manner.

1 Introduction

In this work we deal with segmentation – perhaps one of the most popular tasks in computer vision. In our opinion, the actually most successful methods are based on Energy Minimization techniques. Both modelling aspects (especially for segmentation) and efficient algorithms were extensively studied and elaborated in the past. However, there are still many open questions in this context. The first one is that simple pairwise energies are rarely able to produce satisfactory results in practice without further enhancement and/or learning. Simple and more or less general models (like e.g. [3]) give satisfactory results only in quite easy situations, for example if the colour distributions for segments are known and do not essentially overlap. Usually, either user interaction [8] or very elaborated energy functions (e.g. [2,4,6]) are necessary to improve the results. Unfortunately, very often the Energy Minimization tasks to be solved occur to be quite hard. It leads to inference algorithms, which are typically not very efficient.

In this paper we follow another strategy which is becoming increasingly popular for MRF-based approaches, namely the Maximum Marginal decision. It was already shown, that the marginal based decision strategies give at least competitive results for example for image denoising and deblurring [11,12] and stereo reconstruction [9]. For segmentation however, this option seems to remain unexplored so far. In our opinion, the main challenge for a comprehensive comparison is the lack of algorithms for marginal based inference, which are both accurate and computationally efficient enough for typical computer vision tasks. Therefore, for complex models the marginal based inference usually does not perform well in practice. In this work we address this problem in a straightforward way – we show, how to reach good (in our case real-time) performance using reasonable assumptions and a careful implementation.

Another important question is learning. Modern models usually use Conditional Random Fields (CRF) in order to incorporate as many additional aspects

A. Pinz et al. (Eds.): DAGM/OAGM 2012, LNCS 7476, pp. 387–396, 2012.

as possible. The CRF framework is very convenient in this respect, because it allows to incorporate almost everything in a sound theoretical way. Elaborated general methods can be used for learning, for example the framework of Structural SVM-s (see e.g. [7] and references therein). CRF-s have, however, their price. The most crucial question is the generalization ability. The more complex the model, the more effort is needed to provide statistically sound guarantees for the results. The next problem is that the semi-supervised learning becomes extremely difficult. Furthermore, fully unsupervised learning is not possible at all. Especially this aspect becomes crucial in situations where there is no chance to have completely labelled data for training. One such situation is the segmentation of live video streams, which we address in this work. In dealing with learning we heavily exploit the fact, that the same quantities are necessary for both inference and learning. Therefore the latter can be easily included with almost no additional computational cost.

2 Model

The presented approach is based mainly on [5,10] except for the appearance model and the shape parametrization. In this section we recall the main ideas and discuss different model parts.

Let R be a set of image pixels. They are considered as vertices of a graph $G = (R, E)$, where the edges $\{r, r'\} \in E$ connect neighbouring pixels (in particular we use 4-neighbourhood). The image $x : R \to C$ is a mapping that assigns a colour value $c \in C$ to each pixel $r \in R$. The colour value in a pixel r is denoted by x_r. In binary segmentation a label $k \in \{0, 1\}$ (background/foreground) should be assigned to each position r forming a labelling $y : R \to \{0, 1\}$. A label chosen by the segmentation in a pixel r is denoted by y_r. As we are interested in segmentation of a live video stream, we cannot expect a reasonable user interaction, i.e. the learning of unknown parameters should be performed in a fully unsupervised manner. Therefore, we advocate a generative model that consists of the prior probability distribution of labellings $p(y)$ and a conditionally independent probability distribution $p(x|y) = \prod_r p(x_r|y_r)$ for observations. For the prior we use the Ising model enhanced by a shape prior. To summarize, the probability distribution for pairs (x, y) reads

$$p(x, y; \phi, \theta) = \frac{1}{Z(\phi, \theta)} \exp\big[E(x, y; \phi, \theta)\big], \tag{1}$$

with the energy

$$E(x, y; \phi, \theta) = \alpha \sum_{rr'} \mathbb{1}(y_r{=}y_{r'}) + \lambda \sum_r y_r \phi(r) + \sum_r q(x_r, y_r; \theta) \tag{2}$$

and the normalizing constant $Z(\phi, \theta)$[1]. The coefficients α and λ weight the importance of the Potts prior and the shape prior respectively. Unfortunately,

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

their learning is very time consuming (especially in an unsupervised manner) and actually can not be performed in real time. Hence, we consider them to be known. Other parameters of the probability distribution are the shape function $\phi : R \to \mathbb{R}$ and the unknown parameters θ of the appearance model.

The second energy term is the shape prior, that assigns additional unary terms $\phi(r)$ for the foreground label in each pixel r. We use simple quadratic shape, parametrized in a Gaussian like manner as

$$\phi(r) = \phi_0 - \frac{(r_x - z_x)^2}{2\sigma_x^2} - \frac{(r_y - z_y)^2}{2\sigma_y^2}. \tag{3}$$

(subscripts $_x$ and $_y$ correspond to the horizontal and vertical directions, respectively). The parameters of the shape function are the centre z, variances σ and a bias ϕ_0. The function has the following influence on the prior probability distribution. Foreground labels $y_r{=}1$ at positions close to the centre z are supported by an additional positive energy $\phi(r)$. The centre z has thereby the maximal possible support of ϕ_0. Positions far from the centre are suppressed accordingly. Zero level set is an orthogonal ellipse with the half-axes $\sigma_x\sqrt{2\phi_0}$ and $\sigma_y\sqrt{2\phi_0}$.

The data-terms $q(x_r, y_r; \theta)$ in (2) are logarithms of conditional colour probabilities $p(x_r|y_r; \theta)$. A common choice for the appearance model is a multivariate Gaussian mixture for each segment (see e.g. [8]). Taking into account that the model should be as simple as possible for computation and can be learnt quickly, we modify the standard Gaussian mixture model in the following way. First of all we use orthogonal isotropic Gaussians instead of the multivariate ones, i.e. the i-th Gaussian is given by[2]

$$\mathcal{N}(c; \mu_i, \sigma) \sim \frac{1}{\sigma^{3/2}} \exp\left[-\frac{\|c - \mu_i\|^2}{2\sigma^2}\right]. \tag{4}$$

The variance σ is thereby common for all Gaussians. This simplification is obviously computationally more efficient because it is not necessary to compute matrix products. However, it has its price, namely more Gaussians are needed in order to adequately represent the target probability distributions of colours. In the next section we give some hints how to cope with this problem using appropriate computational schemes.

The second modification is that we use a common set of Gaussians for both segmentation labels, i.e. the probability of a colour c for a label k is

$$p(c|k) = \sum_{i=1}^{n} w_{ki} \mathcal{N}(c; \mu_i, \sigma). \tag{5}$$

Hence, the appearance models for labels differ only by the mixture coefficients w_{ki}. This modification has numerous advantages compared to the standard case, where distinct Gaussian sets are used for different segmentation labels. The main one is with respect to learning. In video processing it is often the case that the

[2] Colours c are three-dimensional vectors e.g. in the RGB colour space.

appearance model should be re-learnt very quickly. Let us consider the situation, when a new object appears in the video stream. If distinct sets of Gaussians are used for different labels, this object is labelled as foreground or background mainly based on its colouring. In the proposed modification instead, both labels "have a chance" to occupy the object. Hence, other model aspects can influence the final decision about it. Besides, it is easy to see that the case of separate Gaussian sets for each segment is a special case of the proposed "common pool", if particular weights of the latter are set to zero. Therefore we do not see the necessity to additionally restrict the appearance model.

3 Inference and Learning

The segmentation is formulated as a Bayesian decision task with the Hamming distance between two labellings $H(y,y')=\sum_r \mathbb{1}(y_r{\neq}y'_r)$ as the cost function. It leads to the maximum marginal decision

$$y_r^* = \arg\max_k p(y_r{=}k|x;\phi,\theta). \tag{6}$$

The posterior marginal probabilities of states are computed approximately using Gibbs Sampling. In the next section we give some additional technical details for its efficient implementation.

The most interesting part is the estimation of the shape function. We consider it as an unknown parameter of the prior probability distribution of labellings and follow the Maximum Likelihood principle. The goal is to maximize

$$F = \ln \sum_y p(x,y;\phi) = \ln \sum_y \exp\bigl[E(x,y;\phi)\bigr] - \ln Z(\phi) \to \max_\phi . \tag{7}$$

(in doing so we assume that the appearance parameters θ are known and omit them here for readability). We use the Expectation Maximization algorithm. In the n-th E-step the marginal label probabilities – this time both posterior $p(y_r{=}1|x;\phi^{(n)})$ and prior $p(y_r{=}1;\phi^{(n)})$ ones – should be estimated for the current shape $\phi^{(n)}$. The gradient of the function to be maximized in the M-step is then

$$\frac{\partial F}{\partial\cdot} = \sum_r p(y_r{=}1|x;\phi^{(n)})\frac{\partial\phi(r;\cdot)}{\partial\cdot} - \sum_r p(y_r{=}1;\phi^{(n)})\frac{\partial\phi(r;\cdot)}{\partial\cdot}, \tag{8}$$

where $(\cdot)$ stands for the parameter to be estimated (e.g. ϕ_0, z or σ).

In practice we often observe (for reasonable values of the energy weights α and λ), that the prior label probabilities for the foreground are almost 1 inside and almost 0 outside the zero level set of ϕ. Therefore a good approximation for the second term in (8) can be computed explicitly. In particular it is zero for the differentiation with respect to the shape centre z, i.e. under the above assumption the normalizing constant in (1) does not depend on z at all. To summarize, the above assumption leads to the simple system of equations

$$
\begin{aligned}
&\frac{\partial F}{\partial z} \sim \sum_r p(y_r{=}1|x;\phi^{(n)}) \cdot (r-z) = 0\\
&\frac{\partial F}{\partial \sigma_x} \sim \sum_r p(y_r{=}1|x;\phi^{(n)}) \cdot (r_x-z_x)^2 - \pi\phi_0^2\sigma_x^3\sigma_y = 0\\
&\text{(likewise for } \sigma_y)\\
&\frac{\partial F}{\partial \phi_0} = \sum_r p(y_r{=}1|x;\phi^{(n)}) - 2\pi\phi_0^2\sigma_x\sigma_y = 0,
\end{aligned}
\tag{9}
$$

that can be easily solved for z, σ and ϕ_0.

Let us remember that the marginals are calculated approximately by sampling. An important question is, how many samples are necessary in order to reliably approximate marginals. The parameters of interest here are "of global nature" – these are just five real numbers that influence (and are influenced by) the whole pixel domain. Hence, it is reasonable to assume that statistics accumulated over the whole set of labellings do not deviate essentially from the statistics that are accumulated just over one labelling, sampled according to the given probability distribution. This leads to the following updating schema. First, a labelling $\bar{y}$ is sampled according to the posterior probability distribution with current parameters. The posterior label probabilities in (9) are replaced by 0/1 depending on the sampled label $\bar{y}_r$ in each pixel r. Then the system (9) is solved for the unknown parameters. Finally, the actual shape parameters are moved towards the found "optimal" ones by a step $\eta < 1$.

We omit the detailed considerations for learning of the appearance model $p(c|k)$. In short, we follow a similar scheme. According to the Maximum Likelihood principle the corresponding Expectation Maximization schema is derived. Then the necessary statistics are replaced by the ones accumulated for one generated labelling.

4 Implementation Details

To start with, we consider Gibbs Sampling for labellings y. In each pixel r a label is sampled according to the posterior label probabilities conditioned on the current labels in the neighbouring pixels (we denote them by $N(r)$ and the corresponding restriction of y by $y_{N(r)}$):

$$
p\big(k|x, y_{N(r)}\big) \sim \exp\Big[q(x_r,k) + \phi(r)\cdot k + \alpha \sum_{r'\in N(r)} \mathbb{I}(k{=}y_{r'})\Big]. \tag{10}
$$

As these probabilities are normalized to sum to 1, it is not necessary to compute the above expression for both background and foreground. Only a difference of energies (expression in the square brackets) should be computed:

$$
\triangle e = q(x_r,k{=}1) - q(x_r,k{=}0) + \phi(r) + \alpha \sum_{r'\in N(r)} (2y_{r'}-1) \tag{11}
$$

(here for "foreground−background"). Let ξ be a random number sampled in the range $[0\ldots1]$. Then the foreground label should be chosen if

$$\xi > \frac{1}{1+\exp(\triangle e)} \tag{12}$$

holds. The expression on the right-hand side is a simple function of the energy difference $\triangle e$, which can be precomputed in advance and stored in a look-up table. In summary, it is only necessary to compute (11) in order to sample a new label in a pixel. It is indeed a very simple expression and can be computed very fast, provided that q and ϕ are known.

The most time consuming part is the computation of the data energies q, which are logarithms of the observation probabilities (5). To accelerate it we make use of the observation that the Gaussian number i in (5) can be seen as an additional random variable i_r for each pixel – i.e. the probabilities $p(x_r|y_r)$ are obtained by marginalization over i_r. Therefore the summation over i_r can be replaced by its sampling. Note, that in this case the values of q in (11) are just Gaussian weights, i.e. $q(x_r,k) = \ln w_{ki_r}$, which makes the computation of (11) even faster. We tested different sampling techniques for generation of i_r and finally decided for Metropolis Sampling that gave the best results (taking into account both quality and efficiency). In one sampling step only two "proposals" are considered – the current Gaussian and a new one chosen randomly. The sampling is performed based on difference of their energies. Since we use isotropic Gaussians of the same variance, this difference is a linear function of colour values. Its coefficients can be pre-computed in advance (after each learning step) that makes computations in each pixel extremely fast. For the exponent the corresponding look-up table can be used in a similar way as for the Gibbs Sampling considered above.

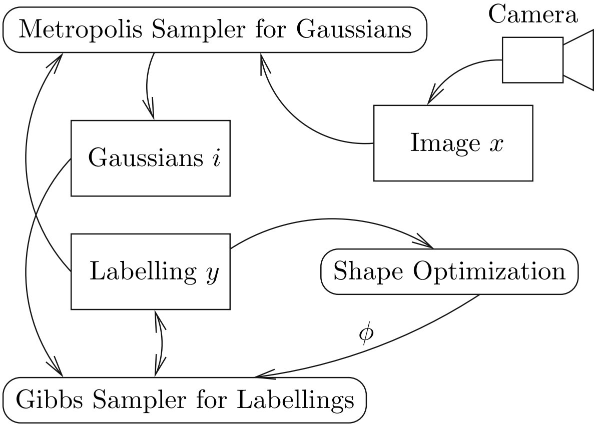


Fig. 1. System architecture

The last question we would like to discuss is the possibility to run different parts of the model in parallel. It is often useful to provide sampling procedures with an initial labelling of high probability in the target probability distribution in order to reduce the so-called "burn-in" phase. In the case of video processing it is reasonable to take as an initialization the last generated labelling, sampled for the previous frame. Consequently, nothing additional should be done in the sampling procedure during the transition from frame to frame, i.e. the procedure should just continue to generate. Moreover it needs not "to know" that the frame was changed. To summarize, the overall system structure is presented in Fig. 1. Common data blocks are shown by rectangles. Rounded rectangles represent procedures, which work asynchronously. Arrows depict data flows. In addition, a fourth procedure is necessary for frame capturing and visualisation. Hence, the system fits very well into modern quad-core architectures.

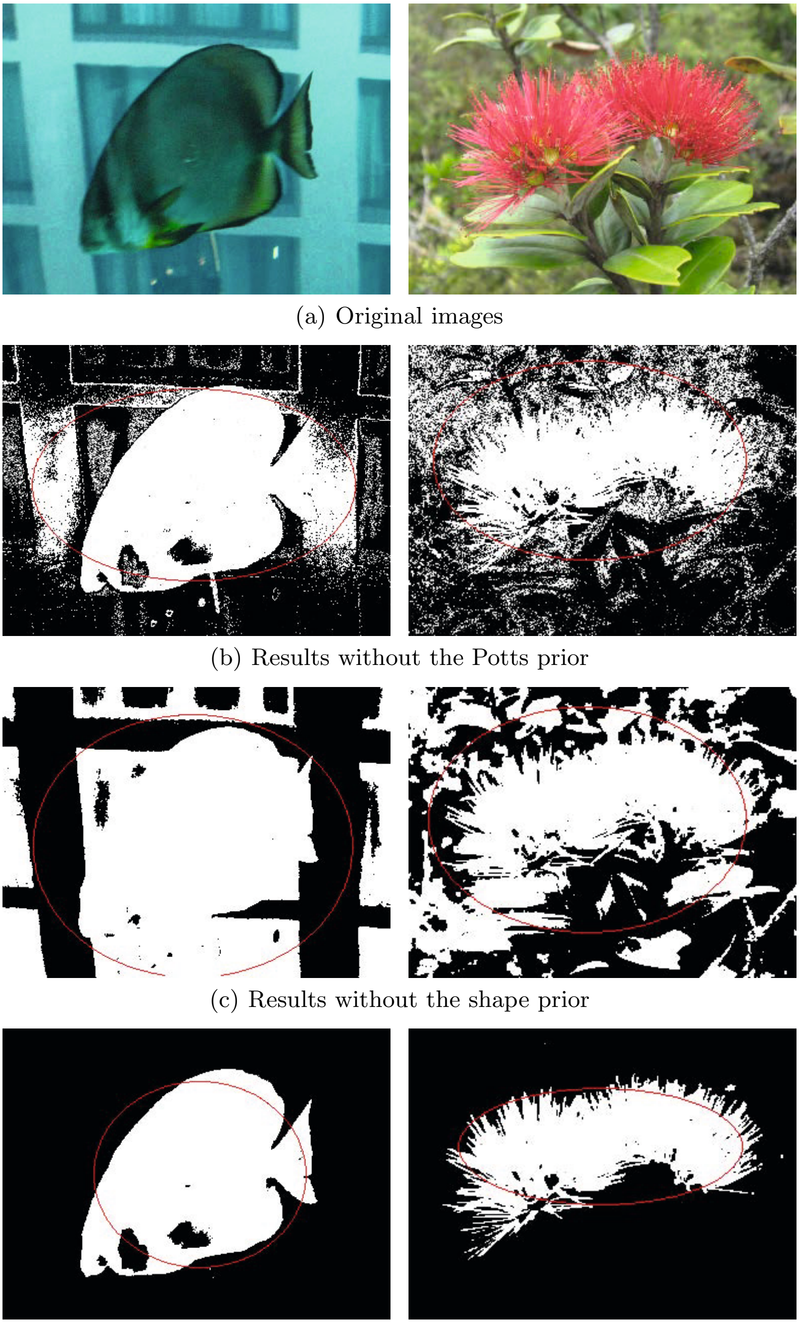
In this section we described only the most important aspects that allow to implement the needed inference and learning procedures in a very efficient manner. More technical details will be given in a technical report in the near future. The complete source code can be found in [1].

5 Experiments

The first question we would like to discuss in this section is the quality of obtained segmentations[3]. We would like to note from the very beginning that our system of course does not outperform state-of-the-art methods, mainly because of its simplicity. In our opinion, the main advantage of the proposed approach is a closed and compact form that gives satisfactory results, it includes unsupervised learning and works in real time. Therefore we prefer just to give qualitative results together with discussions about the system capabilities and limitations.

The model consists of three main parts: the Potts prior, the shape prior and the appearance model. Hence, an important question is, what is, for example, the influence of the shape prior and would the results be considerably worse without it. The impact of different model parts is illustrated in Fig. 2. In the top row two examples are presented. In the first one (fish) the colours are highly ambiguous which makes the segmentation quite difficult. In the second example (flower) the colours are discriminative enough. However, the shape differs considerably from an ellipse. In the second row results are presented that were obtained by the model without the Potts prior (i.e. $\alpha = 0$). The zero level set of the found shape is shown in red. A higher density of foreground pixels can be clearly observed close to its centre. However, the results are very noisy and obviously far from satisfactory. In the third row results are shown that were obtained without the shape prior (i.e. $\lambda = 0$). This clearly illustrates, that in this case the unsupervised learning of the appearance model often goes into a completely wrong direction – although the results are not noisy as before, the foreground segment does not represent a compact "object".

[3] Here we present results for still images only. Examples of the live video segmentation are given in [1].



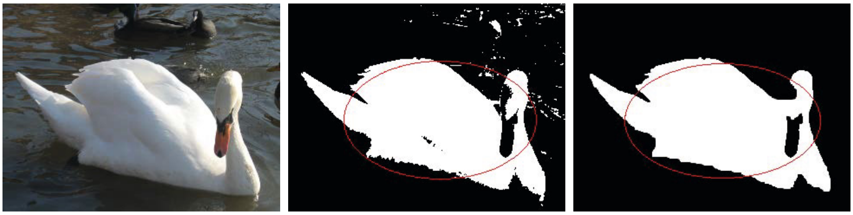
(a) Original images

(b) Results without the Potts prior

(c) Results without the shape prior

(d) Full model

Fig. 2. Influence of different model parts



(a) Original image (b) $\alpha = 1$, $\lambda = 1$ (c) $\alpha = 4$, $\lambda = 3$

Fig. 3. Influence of the model parameters

Table 1. Performance indicator for different environments

Intel Core i7-2620M	4×2.7 GHz	Ubuntu	2062
Intel Xeon 54XX v7.0	8×2.8 GHz	MacOS	2020
Intel Core2 Quad Q6600	4×2.4 GHz	WindowsXP	1970
Intel Core2 Duo T5750	2×2.0 GHz	Ubuntu	735
AMD Athlon X2	2×2.2 GHz	Windows7	473

In the next experiment we would like to illustrate the influence of the model parameters, which are not learned, i.e. the Potts strength α and the weight λ for the shape. This is shown in Fig. 3. In Fig. 3(b) the results are given for values, which we consider as most appropriate for the real-time video processing. It is easy to see, that they lead to non-satisfactory results due to both colour ambiguities and deviation from the elliptical shape. In order to obtain better results (see Fig. 3(c)) it is necessary to make the prior model stronger (in particular to use higher Potts strength). Unfortunately, a high Potts parameter leads to the long burn-in phase of Gibbs Sampling – i.e. more iterations are necessary to sample a good segmentation for the current frame starting with the last generated labelling for the previous one. Therefore a strong prior model can be used for real-time set-up only with video streams of low resolution or for a relatively slow motion.

Finally, we discuss the computational speed of the method. Unfortunately, the notation "real-time" is not well defined as such, because it highly depends on the particular environment. Therefore we prefer to give some concrete data, obtained for different architectures. Most of our experiments were performed on an Intel Core i7-2620M, 2.70GHz (64 bit, quad-core) for frame resolution 320×240 at 30 frames per second under Linux. The Metropolis Sampling for Gaussians was able to perform about 11 sampling iterations[4] per frame, the Gibbs Sampling for labellings – about 13 iterations per frame and the shape optimization was done about 45 times per frame. We measure the overall performance of a particular environment just by counting all activities (sampling iterations or optimization steps) per second. For example, in the above configuration there were about 2060 activities per

[4] One iteration is a scan over the whole image.

second. This performance indicator for different environments for 320×240 frame resolution is summarized in Table 1. In higher resolution (e.g. 640×480) the system gives satisfactory results as well. Of course, in comparison with low resolution the performance drops accordingly (to about 530 for the first environment), that influences the quality of the results, especially for fast motion.

6 Conclusions

In this work we presented an approach for binary segmentation that (i) is simple and therefore more or less general, (ii) includes fully unsupervised learning, (iii) is able to work in real-time and (iv) gives satisfactory results.

As our model is very simple, there are numerous ways for improvements. The main one is to use more elaborated shape priors. In our current implementation it is oversimplified and does not represent a "shape" in a common sense, but rather a "region of interest" that regularizes the learning. We hope however that more complex shapes can also be implemented in a similar manner, because our assumption about prior marginal probabilities in (9) seems to hold for other shape models as well.

References

1. http://www1.inf.tu-dresden.de/~ds24/rtsegm/rtsegm.html
2. Bleyer, M., Rother, C., Kohli, P., Scharstein, D., Sinha, S.: Object stereo – joint stereo matching and object segmentation. In: CVPR, pp. 3081–3088 (2011)
3. 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)
4. Delong, A., Gorelick, L., Schmidt, F., Veksler, O., Boykov, Y.: Interactive Segmentation with Super-Labels. In: Boykov, Y., Kahl, F., Lempitsky, V., Schmidt, F.R. (eds.) EMMCVPR 2011. LNCS, vol. 6819, pp. 147–162. Springer, Heidelberg (2011)
5. 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)
6. Gulshan, V., Rother, C., Criminisi, A., Blake, A., Zisserman, A.: Geodesic star convexity for interactive image segmentation. In: CVPR (2010)
7. Nowozin, S., Lampert, C.: Structured Learning and Prediction in Computer Vision. Foundations and Trends in Computer Graphics and Vision 6 (2010)
8. Rother, C., Kolmogorov, V., Blake, A.: "GrabCut": interactive foreground extraction using iterated graph cuts. ACM Trans. Graph. 23(3), 309–314 (2004)
9. Schlesinger, D.: Gibbs Probability Distributions for Stereo Reconstruction. In: Michaelis, B., Krell, G. (eds.) DAGM 2003. LNCS, vol. 2781, pp. 394–401. Springer, Heidelberg (2003)
10. Schlesinger, D., Flach, B.: A Probabilistic Segmentation Scheme. In: Rigoll, G. (ed.) DAGM 2008. LNCS, vol. 5096, pp. 183–192. Springer, Heidelberg (2008)
11. Schmidt, U., Gao, Q., Roth, S.: A generative perspective on MRFs in low-level vision. In: CVPR (June 2010)
12. Schmidt, U., Schelten, K., Roth, S.: Bayesian deblurring with integrated noise estimation. In: CVPR (June 2011)