Learnable Swendsen-Wang Cuts for Image Segmentation

Alexander Vezhnevets, Vladimir Vezhnevets
Department of Computational Mathematics and Cybernetics, Graphics and Media Lab
Moscow State Lomonosov University, Moscow, Russia
{avezhnevets, dmoroz}@graphics.cs.msu.ru

Abstract

We propose a framework for Bayesian unsupervised image segmentation with descriptive, learnable models. Our approach is based on learning descriptive models for segmentation and applying Monte Carlo Markov chain to traverse the solution space. Swendsen-Wang cuts are adapted to make meaningful jumps in solution space.

Keywords: *Monte-Carlo Markov Chain, Swendsen-Wang Cut, Image Segmentation, Machine Learning,*

1. INTRODUCTION

We consider the task of unsupervised image segmentation, meaning we assume no user interaction during the segmentation process. Unsupervised segmentation algorithm can be a generic one and seek to generate reasonable segmentation results for any possible image observed. In practice and especially in computer vision one is often working with certain image class and is in need of task-oriented segmentation. For this case, algorithms that can focus on partitioning the image in the way that would provide better representation of image for the task at hand are needed. In this paper we consider inferring segmentation rules from human-labeled collection of images. We do not make any recognition or analysis of resulting region. Thus, our task is to partition an image into number of segments, inferring knowledge of what "good" segmentation is from a manually prepared training set.

Problem of unsupervised image segmentation can be formulated as an optimization problem in graph cuts framework. For this end image is represented as a graph and one seeks to find such a graph partition (cut) that minimizes some given criterion. There is a large body of research dedicated to the problem of unsupervised image segmentation in graph cuts framework. First, algorithms based on graph spectral analysis [7][8] are proposed to find a minimal normalized cut. Also, for supervised segmentation mostly, algorithms based on maximum flow algorithm are proposed [6]. The main criticism of these approaches [2] is narrowness of segmentation models these methods can work with and that they do not have any clear statistical interpretation. Other approach [1][2][3] is to formulate Bayesian model of segmentation, with generative models for image segments and perform a stochastic search in space of all possible segmentations by Monte Carlo Markov chain. Swendsen-Wang graph cuts algorithm can be successfully applied for this purpose. All cited methods are generic and try to obtain reasonable results for any image class. Learning a descriptive model of segments with help of Gestalt features [9] provides more flexible framework, but using naïve optimization method makes it very dependant on initial segmentation and is very computationally expensive.

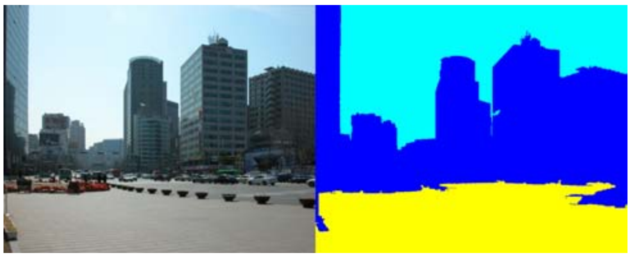


Figure 1: Example of proposed method performance on urban images. Model was trained to segment sky, road and vertical object into different segments.

One specialized approach was proposed for the task of obtaining geometric labeling for image segments [4]. Authors propose to generate multiple "hypothetical" segmentations and average predictions of their geometric labels. To generating those hypotheses a classifier is trained to distinguish image elements that should belong to the same regions.

In this work we use some of the recited methods or their parts as building blocks for our framework. We perform oversegmentation into small segments and use them as low-level image elements to reduce the task complexity as in [2][4]. We use descriptive models for image segments trained offline as in [9]. We also learn similarity of image elements (probability to belong to the same segment) and use it to sample the solution space as in [4]. We use Bayesian formulation for segmentation as in [1][2], but use learned descriptive models for segments. We employ Monte Carlo Markov chain to find maximum aposteriory probability solution as in [1]. We implement Swendsen-Wang cuts [2] to make chains moves in solution space significant, getting faster convergence. While we heavily rely on related research we significantly differ from described works. We use far more general and theoretically sound optimization procedure then in previous works on descriptive models for segmentation.

2. PREPROCESSING AND FEATURES

To get better spatial support and to be more computationaly efficient we perform oversegmentation by EDISON [10] mean-shift based algorithm. This algorithm makes a generic segmentation of given image into many small sized homogenous elements, often called superpixels. The choice of overegmentation algorithm is arbitrary.

In general, it is not relevant what low-level image representation we employ. It could be pixels, superpixels, edgelets and anything else. Bellow we will use "image element" when referring to them. Any union of image elements we will call a segment or a region.

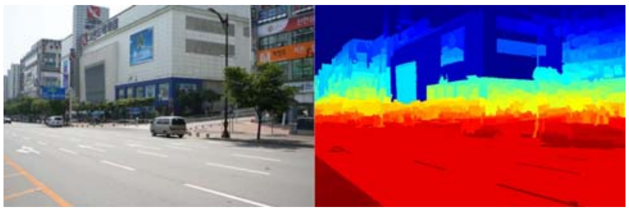


Figure 2 Example of oversegmentation by EDISON method.

2.1 Segment features

Choice of features, describing image elements and segments, is arbitrary and should be done according to the task at hand. We choose features described in [4], since we were experimenting with urban scenes. Probably, features used in [9] would yield better results in general case.

Feature for segments and image elements are the same. They consist of color, location, shape and perspective cues estimated from an image. One more advantage of this feature set is that it can be efficiently recalculated for a given segment if all features for segments elements are known (most of features would be average of elements features, weighted by their area).

3. BAYESIAN SEGMENTATION

We formulate segmentation just the same as in [1].

3.1 Graph partition

Let $G =< V, E_0 >$ be an adjacency graph, where $V = \{v_1, v_2, ..., v_p\}$ is the set vertices for image elements such as pixels, superpixels, edgelets or anything else and E_0 is a set of links connecting adjacent elements. Our goal is to partition initial graph into a set of connected subgraphs $G_k =< V_k, E_k >, k = 1,2,..., n$, so that:

$$\bigcup_{k=1,n} V_k = V, V_k \neq 0, V_i \cap V_j = 0 \mid i \neq j$$

$$E_k = \{e = (u, v) \in E_0 \mid u, v \in V_k\}, k = 1,2,..., n$$

Generally, the number of desired sub graphs is unknown. We will denote segmentation as $\pi_n = \{G_1, G_2, ..., G_n\}$.

The space of all partitions (segmentations) is denoted by

$$\Omega_\pi = \bigcup_{n=1}^{|V|} \Omega_{\pi_n}$$

where Ω_{π_n} is a space of all π_n (partitions into n segments).

Each sub graph $G_k =< V_k, E_k >, k = 1,2,..., n$ represents a coherent visual pattern. Links between any two given sets of vertices V_i and V_j is a *cut*

$$C(V_i, V_j) = \{e =< s, t >: e \in E_0, s \in V_i, t \in V_j\}, i \neq j$$

3.2 Image model

We will denote observed image as *I*, image information for an image element *v* as I_v and information for elements set *V* as I_V. We use term “image information” to describe observed raw image data. Afterwards, special features can be computed from it. A segmentation of an image is

$$W = (n, \pi_n, C),$$

where C is a model for an image segment. We will describe construction of C in section 6, now let us just assume it is a descriptive model, from which we can estimate $p(I_V \mid C)$.

If we assume patterns independence, then a Bayesian posterior would be

$$W \sim p(W \mid I) \propto \prod_{i=1}^{n} p\left(I_{V_i} \mid C\right) \cdot p(W).$$

4. BASIC IDEA

We formulate unsupervised segmentation task as graph partitioning problem. We also defined a Bayesian posterior distribution in the space of all possible graph partitions, given a model for a segment. Since our goal is to find “the best” segmentation possible given a concrete image, we want to find MAP graph partition that should provide us with such segmentation. There are two things we need for solving a given task – model for segments and come up with algorithm to traverse the solution space.

To derive segments model we take learning approach as in [4][9] and use boosted decision trees to learn posteriors from human-labeled collection of images. The learning process is described in section 6.

To traverse solution space we use Monte-Carlo Markov chain (MCMC) much like as in [1]. MCMC sampling methods are known in statistics [11] as methods of sampling unknown distributions, for which only the probability of any given samples can be computed. In words, we construct a stochastic process (Markov chain) which has graph partitions as its states. Each state transition (jump) is first sampled from a special *proposal distribution* and is accepted (jump is made) with some probability. Such randomized process explores the solution space searching for a MAP solution. While it does not guarantee global maxima (if we run the process for a finite number of iterations), due to randomness it is resistant to local maxima problem (it is able to jump from it). We use discriminative model for image elements similarity to derive proposal distribution (section 5), from which we can sample. It is also learnt by boosted decision trees. We use Swendsen-Wang cuts [2] to propose strong moves in the solution space. Markov Chain is run for a given amount of iteration and then the solution with maximum posterior is chosen.

5. SAMPLING THE SOLUTION SPACE

To effectively traverse solution space we need a reasonable distribution in that space which we can sample. We use the same idea as in [4]. We estimate probability for two image elements *s,t* to be in the same region (probability of edge *e* to be present), which is learned from ground-truth image collection.

$$q_e = p\left(e \mid F(s), F(t)\right),$$

where $F(s), F(t)$ are local features computed from image. This probability gives us a distribution in the space of all possible segmentations. Any partition of adjacency graph (segmentation) is defined by set of graph links *E*, thus

$$q(E) = \prod_{e \in E} q_e \prod_{e \in E_0 - E} (1 - q_e)$$

is a distribution in segmentation space. We need this distribution, because it is easy to sample from. Indeed, if we turn edges in E_0 "on" with probability q_e, then resulting adjacency graph will be a sample form described distribution. We need this distribution to "guide" our search.

To control coarseness of the segmentation we can introduce "temperature" parameter *T*. Making probabilities $q_e{}^T$ we can achieve different, but sensible results.

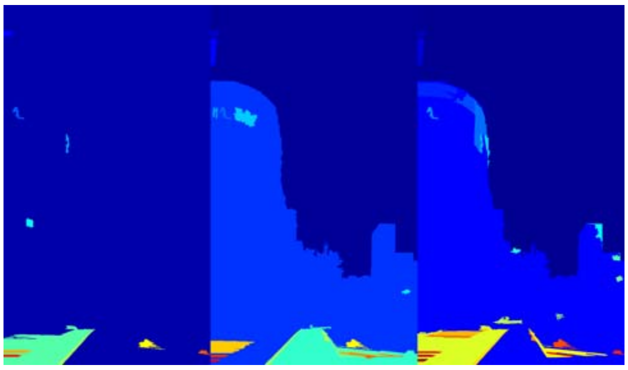


Figure 3 Sampling with different temperature parameters. From left to right *T=1,2,3*.

6. MODEL LEARNING

To obtain segments posteriors and probability of image elements to correspond to the same segment we use machine learning. We use a collection of human-labeled images to learn from. Each image is manually labeled into homogenous regions. So far, we have two models to learn – elements similarity and segments posteriors. To start learning we need to construct training set, consisting of positive and negative examples, for a learner to distinguish from one-another. Segments and superpixels are represented by features described in section 2.1.

6.1 Learning elements similarity

For learning image elements similarity $q_e = p\left(e \mid F(s), F(t)\right)$ we need to learn the probability for image elements to be in the same segment. Features of image elements that belong to the same segment in human-labeled examples are taken as positive examples, and those labeled as belonging to different as negative. We use boosted decision trees [12] to learn the probabilities. Trained classifier has error 12% measured by cross-validation.

6.2 Learning posteriors $p(I_V \mid C)$

Constructing positive examples for segments is easy, since we can explicitly use segments from ground truth labeling. To construct negative examples [9] proposed to apply labeling from one image to another (providing wrong, but sensible segmentation). Such approach is not quite adequate from machine learning perspective, because those segmentation examples are drawn from a different distribution then the segmentation algorithm will observe during simulation. Fortunately, we have ability to sample from the distribution we will observe, since it is the sampling function learnt on previous step. We sample segmentations from the proposal distribution and construct negative examples from those of them that either have heterogeneous ground truth labeling or are significantly smaller then ground truth segment. Trained classifier has error 6% measured by cross-validation.

It is important to use different parts of training set for learning segments posteriors and proposal function to avoid overfitting.

6.3 Obtaining calibrated probabilities

Boosted decision trees, being logistic regression model, can be made to produce the probabilistic output. Authors in [7][4] used normalized margins of prediction as probabilities. Recent research [5] suggest that such transform yields poorly calibrated probabilities. We use Platt scaling [13] to calibrate outputs of constructed classifiers. This step is very important since we search for MAP solution. Wrong probabilities can yield visually inadequate solutions. We calibrate models for both similarity and posterior learning.

6.4 Segmentation prior $p(W)$

Introduced models for image segments are local in their nature (since we assume their independence) and a reasonable segmentation prior could contribute much to adequateness of a final segmentation. We didn't use any segmentation prior in this work, but any model for image segmentation prior, such as Markov random field, could be employed, since method does not have any limitations for it.

7. MARKOV CHAIN DESIGN

To design a Markov Chain to traverse solution space we need to define several *dynamics* – ways for chain to change its state. For Markov chain to sample from correct distribution it should be ergodic and observe detailed equation. Markov chain is ergodic if it is irreducible and its states are a periodic. In words, that means that any state can be reached from any other with non zero probability by a finite number of jumps and that there is no such state that we will periodically visit with probability one. Detailed equation means that for any move probability of making it equals the probability of going backward.

Since we are using descriptive model trained offline, we need only to devise jumps in partitions space – split, merge, death, birth. Jumps are made according to Metropolis-Hastings algorithm [11]. According to it, we must accept the proposed jump with probability

$$\alpha(A \to B) = \min(1, \frac{q(B \to A)}{q(A \to B)} \frac{P(B \mid I)}{P(A \mid I)})$$

where *A* and *B* are states of chain (graph partitions in our case), and *q* is the proposal distribution from which we sample. In words, we need a probability of a chain to make a jump to be equal to the probability to jump back. This is a sufficient criterion for a Markov chain to converge to desired distribution.

8. SWENDSEN-WANG CUT

In this section we will briefly describe Swendsen-Wang cuts algorithm. Method does not depend on type of posterior, so it is generally the same as original method [1] for descriptive models.

To efficiently explore solution space we use Swendsen-Wang [2] graph cuts algorithm. At each step, given a graph partition $G_k = < V_k, E_k >, k = 1,2,...,n$ algorithm turns "on" edges $e \in E_k$ with probability q_e and obtains a set of connected components CP. Each component belongs to a certain sub graph. One of the components, chosen at random, is proposed, according to some distribution, to be reassigned another subgraph or to form new subgraph.

Swendsen-Wang Cuts:

Input $G_0 = < V_0, E_0 >$, discriminative probabilities $q_e, \forall e \in E_0$, and descriptive posterior probability $P(W \mid I)$

Output: Samples $W \sim P(W \mid I)$

```
1. Initialize a graph partition: $\pi : G = \bigcup_{i=1}^{n} G_i$
2. Repeat, for current state A
3.   Repeat for each subgraph $G_l = < V_l, E_l >, l = 1,2,...n$,
4.     For $e \in E_l$, turn edge "on" with probability $q_e$.
5.     $V_l$ is divided into $n_l$ connected components:
         $\{g_{li} = < V_{li}, E_{li} >, i = 1,...,n_l\}$.
6.   Collect connected components from all subgraphs
         $CP = \{V_{li} : l = 1,...,n, i = 1,...,n_l\}$.
7. Select a component $V_0 \in CP$ at random with probability
   $q(V_0 \mid CP)$ (usually $q(V_0 \mid CP) = 1/|CP|$ ).
8. Propose to assign $V_0$ to a subgraph $G_{l'}$. $l'$ follows a
   probability $q(l' \mid V_0, A, G_0)$
9. Accept the move with probability $\alpha(A \to B)$.
```

Reassigning a connected component from one subgraph to another provides split and merge dynamics, assigning a connected component to a new subgraph yield birth operation and assigning a connected component containing the hole subgraph make death operation.

Now we have partition space traversing algorithm, we need to define jump probabilities to make this random walk effective.

Theorem 1. [2] *Considering the above notation, if a candidate* $V_0 \in CP$ *, proposed to reassign from* G_l *to* $G_{l'}$ *with probability*

$$\alpha(A \to B) = \min(1, \frac{\prod_{e \in C(V_0, V_{l'} - V_0)} (1 - q_e)}{\prod_{e \in C(V_0, V_l - V_0)} (1 - q_e)} \frac{q(l \mid V_0, B, G_0)}{q(l' \mid V_0, A, G_0)} \frac{P(B \mid I)}{P(A \mid I)}$$

then the Markov chain is ergodic and observes detailed equation.

The proof of the theorem can be found in [2]. Since theorem does not put any limitations on posteriors there is no difference if we use generative models as in original method, or employ learned models.

To be most effective we must design proposal probability $q(l' \mid V_0, A, G_0)$, so that $\alpha(A \to B) = 1$. In this case, each move is always accepted and our Markov chain becomes a generalized Gibbs sampler.

Consider partition state $A = (V_1, V_2, ..., V_n)$ (each set of nodes V_i correspond to respective sugraph $V_i \in G_i$) and a component $V_0 \in V_l$ chosen by SWC for reassignment. Then we have $n+1$ possible assignments

$$\{S_1 = V_1, S_1 = V_1, ..., S_l = V_l - V_0, ..., S_n = V_n, S_{n+1} = \varnothing\}.$$

We denote corresponding resulting states $\{B_1, B_2, ..., B_{n+1}\}$ respectively. The proposition is to choose the assignment $S_{l'}$ with following probability

$$q(l' \mid V_0, A, G_0) \propto \prod_{e \in C_{l'}} (1 - q_e) \cdot p(B_{l'} \mid I).$$

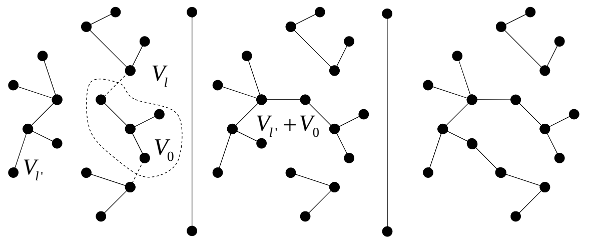
It is proved [2] that such choice of probability for assignment really makes $\alpha(A \to B) = 1$. Again, there is no difference if we use learned discriminative models instead of generative.

This choice of assignment proposition can be interpreted as steering to assign component to the best possible segment, modulating the decision by cut weight.

8.1 Reducing calculations

Clearly, computational issues arise. Calculating cuts is fast, but evaluating posteriors for each possible assignment is exhaustive. We will address this issue below.

Basically, we want to limit moves to reassignment between adjacent components. The main issue is preservation of ergodicity and detailed balance. If we just limit possible subgraph candidates to adjacent ones, then we can possible get a state not directly reachable from a previous one. Consider jumps illustrated on scheme 1. First jump $A \to B$ reassigns component V_0 from V_l to $V_{l'}$, producing new partition with sugraphs formed by nodes $V_l - V_0$ and $V_{l'} + V_0$. Notice, that subgraph formed by nodes $V_l - V_0$ is not connected. If during next step $B \to C$ one of connected components from $V_l - V_0$ is reassigned, then there is no direct backward jump, because reassigned component and its former subgraph are not adjacent any more.



State A State B State C

Scheme 1. Illustration of possible inconsistent state. There is no jump from state C to state B, since for smallest component its former subgraph is not adjacent.

To avoid such situations we can simply assume the posterior of unconnected subgraphs to be zero.

$$\tilde{p}(I_V \mid C) = p(I_v \mid C) \cdot I_{con}(V)$$

Where $I_{con}(V)$ is indicator of nodes in V to be adjacent (clearly, we should have also included adjacency matrix E_0 into condition, because it determines connectivity of *V*, but we didn't do it to simplify notation).

Indeed, in this case we have a zero probability to make an assignment that would lead to undesired situation. This does not affect any conditions for detailed balance or ergodicity, since it only modifies posterior.

9. RESULTS

To give a try to our framework we have conducted experiments on a set of urban scene images. Areas of image containing only sky, road or buildings were labeled as homogenous. Cars, pedestrians and other obstacles where treated as part of "building" segments if they were obscuring them, or labeled as individual segment if otherwise. We used about a hundred images. First thirty images where used for learning proposal distribution, sixty for learning segments posteriors and others were used for testing. All images were obtained by handheld digital camera. Figure 9 gives segmentation results and can be found on the last page of the paper.

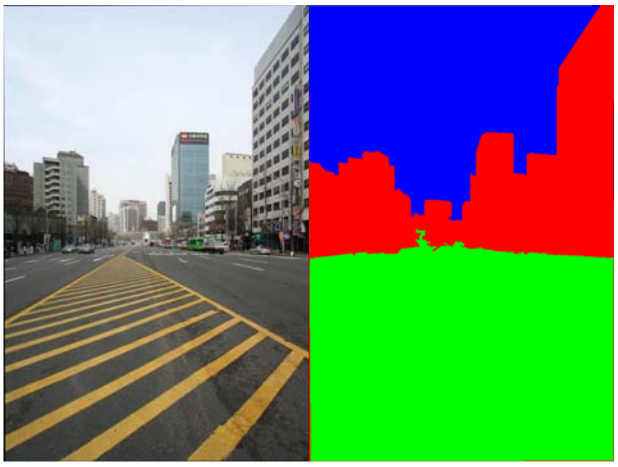


Figure 4 Image from the training set and it manual grounf-truth segmentation.

9.1 Posterior optimization dynamics

Figure 4 shows the dynamic of posterior probability for image (logarithmic scale). We used linear cooling schedule, starting with *T=4* and linearly cooling to reach *T=1* at the final iteration. For initialization we used a random segmentation sampled with help of learnt similarity function for image elements.

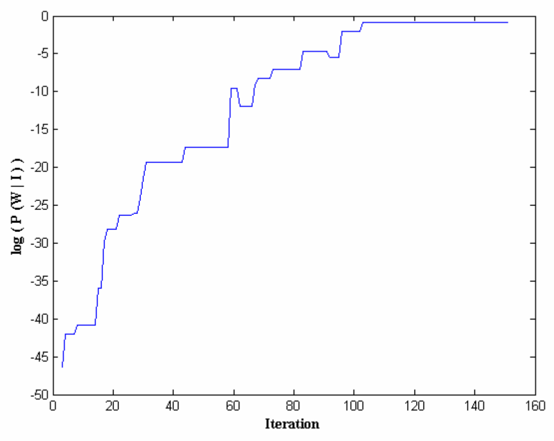


Figure 5 Logarithm of posterior probability of image 2.

It takes about 500 seconds for process implemented in Matlab with no optimization on Athlon 2 GHz processor to converge for ~0.7 megapixel image. It is much faster then method with naive optimization [7] that took from 15 to 40 minutes to converge for ~0.1 megapixel image. Computational bottleneck is feature calculation as it takes about 1-3 seconds to calculate them for all image segments. In our current implementation features for all segments are recalculated each time the posterior for possible state is estimated, while actually only 2 segments features change. Also, feature set could be reduced, since boosted decision trees use only about 30% of them (although many of them are calculated simultaneously in one operation, so removing 70% features will not provide respective boost in calculations). For further optimization hierarchical Swendsen-Wang cut algorithm [3] could be used.

9.2 Other approaches

It is reasonable to evaluate whether there can be more simple way to solve the problem. We tried simplifications – greedy optimization and normalized cut for adjacency graph, with links weighted by leant similarity function. First experiment should set clear if stochastic search for global minima is significant for segmentation quality. Second should prove that Bayesian approach is relevant and just learning pair wise affinity for image elements and applying classical graph cuts is not enough.

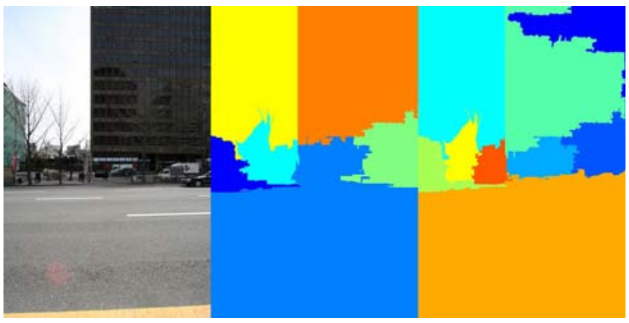


Figure 6 Results for normalized cut segmentation using learnt super pixel's similarity function. From left to right: original image, segmentation into 6 segments, segmentation into 9 segments.

9.2.1 Normalized cuts

Figure 5 shows segmentation corresponding to a cut with minimal normalized cost for a graph $G = < V, E_0 >$ with links having weights $1 - q_e$. It is clear, that segmentation is not visually adequate. This is due to two factors. First, image elements affinity is not perfectly learned (and can hardly be learned significantly better). Second, minimum normalized cost has no statistical interpretation and may not be quite accurate model for visually coherent segmentation.

9.2.2 Greedy optimization

We have implemented a following greedy optimization procedure – at each step, a segment with minimum posterior is chosen for reassignment and a move that will guarantee the maximum increase in overall posterior of segmentation is chosen and accepted with probability one. Such method guarantees an increase of posterior probability at each step (exception may be when a chosen component cannot be reassigned with any gain in posterior). It is clear, that once segmentation converges to a local maximum (no component can be reassigned with increase in probability) segmentation will not change. Experiments have shown that such algorithm, as anticipated, is very dependant on initialization. Original stochastic algorithm also benefits from good initialization, but as theory states and experiments justify will sooner or later converge to desired distribution independent of initial state.

Nevertheless, greedy algorithm converges quite fast and in case of good initial state provides good results. Probably, in tasks where good initial segmentation can be provided by heuristic means can be effectively used.

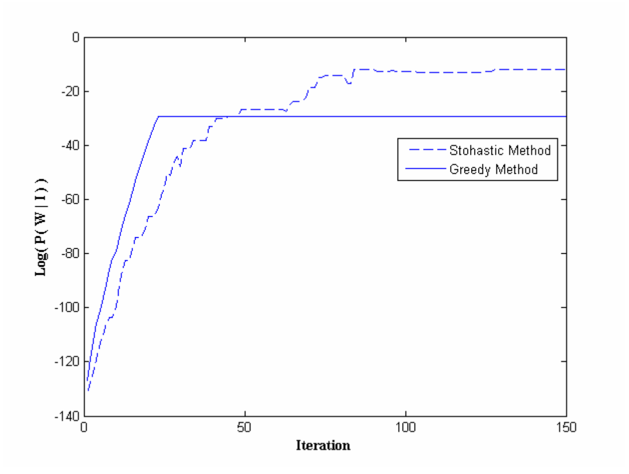


Figure 7 Posterior dynamics for greedy and stochastic methods. Dashed line and solid lines represent stochastic and greedy methods respectively.

9.3 Failure study

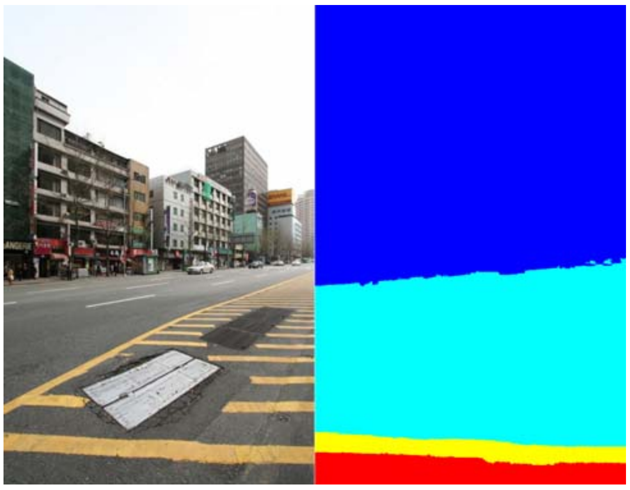


Figure 8 Example of visually bad segmentation results.

To no surprise, method fails in some cases. Figure 6 presents the example of failure. The problem in this case, as in others observed during experiments, was always problem of posterior estimation. We use learning that yields about 10% error, thus a global minimum of such posterior distribution is not necessary visually adequate. There is a little chance that features and learning methods can reach 100% accuracy any time soon. Ideas that may help avoid "overfitting" of learned posterior could be, for example, an adequate image prior. Also, as shown in geometric labeling [4], marinating multiple hypotheses and producing an average result should also provide more robustness. We will discuss more ideas for improvement in section 11 dedicated to future work.

10. CONCLUSION

This paper proposed a new framework for unsupervised image segmentation with learnable descriptive models for segments. Segmentation process is formulated as search for maximum posteriory probability of in the space of all possible segmentation and Swendsen-Wang cuts algorithm to traverse the solution space is employed. Learning pair wise image elements similarity provides means for effectively sample solution space and also get a well posed machine learning task for estimating segmentation posterior. A modification to Swendsen-Wang cuts that improves algorithms speed also proposed.

This work was part by three other works: learning classification model for segmentation [9], Swendsen-Wang cuts for image segmentation [2] and geometric context from single image [4]. We learn "goodness" of segmentation as in the first work, use optimization method as in second, and use the idea of sampling segmentations by learning pair wise similarity for image elements from third. However, we significantly differ from all of them. We use far more general and theoretically sound optimization procedure then the first work. We use descriptive models rather than generative in contrast to second. Also, method for evaluating less segmentations by working with only adjacent subgraphs, but maintaining detailed equation and ergodicity is proposed. This makes computations faster and makes a step towards practical applicability of our framework. Comparing to work on geometric labeling of single image it is important to note, that we build a general descriptive model of probability of any given image segment to be homogenous (in words, to be a *good segment*) and do not perform any recognition of segments classes. Extending our framework in such way is discussed in future work section below.

11. FUTURE WORK

Proposed framework is quite general and has lots of possible extensions and improvements. We will discuss some of them.

11.1 Improvements

Currently, main problem is the imperfectness of learned segments model. Some times, a global minimum corresponds to visually inadequate segmentation. Clearly, learning requires more attention. ROC curves should be studied and optimal metrics for optimization should be experimented with. Construction of training sets should also be looked at more closely, probably active learning should be used to obtain edge cutting results for classifiers accuracy. Feature set should also be improved as in terms of descriptiveness and also in computational efficiency.

Although, classifiers can definitely be improved, there is no way to make them perfect. To fight this problem, first of all, segmentation prior should be definitely introduced. Using Markov random field for it seems to be the most natural decision. Also, there is an interesting idea [4] that rather obtaining one solution that seems to be the best, one could obtain many and average along them. In some sense, it suggests to look for expectation of segmentation rather then MAP segmentation. Since our framework is based on procedure that samples segmentation space according to a given distribution it is quite adequate for experiments in that direction.

11.2 Extension

Our framework is quite general and due to a large freedom in defining posterior to be sampled from it can be easily extended to include segments recognition in the process. Really, if we introduce a family of models, rather then just one, we can perform recognition of segments class in the process of segmentation. We should only introduce model switching jumps [1] to Markov chain dynamics.

11.3 Practical applications

Image segmentation is a task wieldy used in computer vision, image processing and adjacent fields. Proposed framework could be successfully applied in medical imaging to automatically segment and recognize different tissues and areas of interest. Learnable segmentation can also be applied as a first step for automatic 3D reconstruction [4] and general scene analysis can significantly benefit from it. Image processing could also use this framework as first step for smart image enhancement and editing.

12. AKNOWLEGMENTS

This work was partially supported by RFBR grant 06-01-39006-GFEN.

13. REFERENCES

[1] Z. W. Tu and S. C. Zhu, “Image Segmentation by Data-Driven Markov Chain Monte Carlo”, *IEEE Trans. on PAMI*, vol.24, no.5, 2002.

[2] A. Barbu, S.C. Zhu, “Graph partition by Swendsen-Wang Cuts”, *ICCV, Nice, France,* 2003.

[3] A. Barbu, SC. Zhu, “Multigrid and Multi-level Swendsen-Wang Cuts for Hierarchic Graph Partition”, *CVPR 2004*

[4] D. Hoiem, A.A. Efros, and M. Hebert “Geometric Context from a Single Image,” *International Conference of Computer Vision (ICCV), IEEE,* Vol. 1, October, 2005, pp. 654 - 661.

[5] A. Mizil, R. Caruana, “Predicting Good Probabilities with Supervised Learning” *Proceedings of the 22nd international conference on Machine learning (ICML)* 2005, pp. 625 - 632

[6] V. Kolmogorov, R. Zabih, “What energy functions can be minimized via graph cuts?”,*ECCV*, 2002

[7] J. Shi, J. Malik, “Normalized cuts and image segmentation”, *PAMI*, 22, no 8, pp. 888-905, 2000.

[8] Z. Wu and R. Leahy, “An optimal graph theoretic approach to data clustering *PAMI*, vol.15, 1101-1113, 1993.

[9] X. Ren and J. Malik, “Learning a Classification Model for Segmentation”. *in ICCV '03,* volume 1, pages 10-17, Nice 2003.

[10] D. Comanicu, P. Meer: "Mean shift: A robust approach toward feature space analysis." *IEEE Trans. Pattern Anal. Machine Intell.*, 24, 603-619, May 2002.

[11] W.K. Hastings. "Monte Carlo Sampling Methods Using Markov Chains and Their Applications", *Biometrika*, 57:97-109, 1970.

[12] Friedman, J., Hastie, T., & Tibshirani, R. (1998). “Additive logistic regression: A statistical view of boosting” *Technical Report.*

[13] J. Platt. “Probabilistic outputs for support vector machines and comparison to regularized likelihood methods”. In *Adv. in Large Margin Classifiers*, 1999.

About the author

Alexander Vezhnevets is a Ph.D. student at Moscow State University, Department of Computational Mathematics and Cybernetics at Graphics and Media Lab. He graduated with honors Faculty of Computational Mathematics and Cybernetics of Moscow State University in 2006. His research interests include, but are not limited to, machine learning, statistics, computer vision, image processing and adjacent field. avezhnevets@graphics.cs.msu.ru.

Vladimir Veznevets is a vice head of Graphics and Media Laboratory of Moscow State Lomonosov University, Department of Computational Mathematics and Cybernetics. He graduated with honors Faculty of Computational Mathematics and Cybernetics of Moscow State University in 1999. He received his PhD in 2002 in Moscow State University also. His research interests include image processing, pattern recognition, computer vision and adjacent fields. His email address is dmoroz@graphics.cs.msu.ru

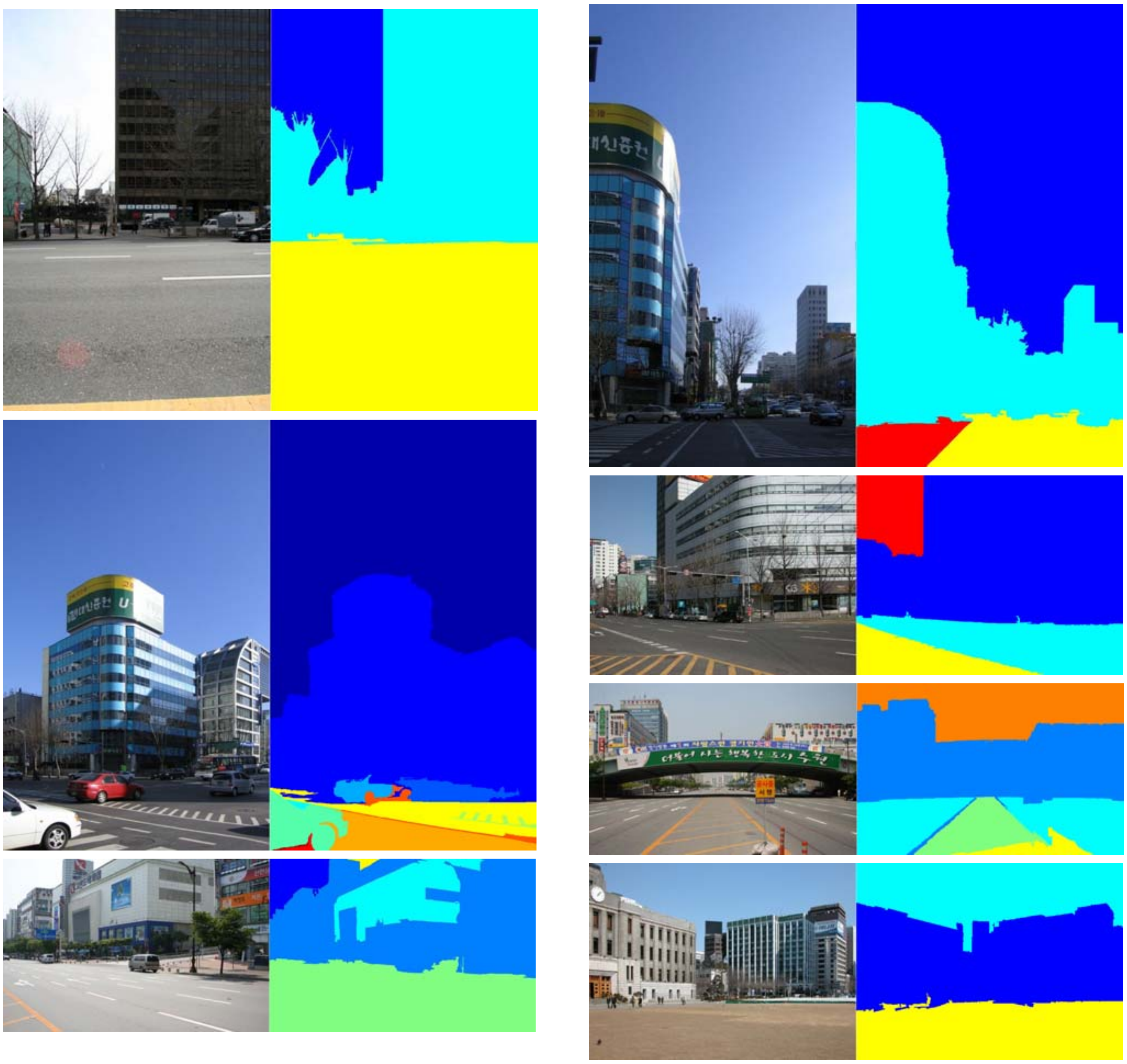


Figure 9. Segmentation results on test set.