Geodesic pixel neighborhoods for multi-class image segmentation

Vladimir Haltakov[1]
http://campar.in.tum.de/Main/VladimirHaltakov
Christian Unger[1]
http://campar.in.tum.de/Main/ChristianUnger
Slobodan Ilic[2]
http://campar.in.tum.de/Main/SlobodanIlic

[1] BMW Group
Munich, Germany
[2] Siemens AG
Corporate Technology
Munich, Germany

Abstract

The problem of multi-class image segmentation is traditionally addressed either with a Conditional Random Field model or by directly classifying each pixel. In this paper we introduce a new classification framework built around the concept of pixel neighborhoods. A standard unary classifier is used to recognize each pixel based on features computed from the image and then a second classifier is trained on the predictions from the first one summarized over the neighborhood of each pixel by a new histogram feature. We define a local and a global pixel neighborhoods which adapt to the image structure by making use of the geodesic distance defined over image intensities.

We evaluate our model on three challenging datasets and show that our model is able to capture both local and global context relations. We compare our method to two strongly related, well known methods and show increased performance.

1 Introduction

Multi-class image segmentation has been a very active research field in recent years because of its importance for many applications like scene understanding and object detection. It is a complex problem that poses several challenges: developing better classifiers, designing more discriminative features, finding more efficient optimization techniques and modeling the relations between the image pixels in different parts of the image. In this paper we focus on the last one. A common way to address the problem of structured prediction is to model it as a Conditional Random Field (CRF) [13], but in this paper we take a different approach by using classification and integrating structure constraints directly in the features.

We introduce a classification framework based on the concept of pixel neighborhoods as visualized in Fig. 1. We first classify each image pixel individually based on features computed from the image. Then, for each pixel we use the geodesic distance in the intensity space to find a set of related pixels called a neighborhood. We introduce two types of neighborhoods: an adaptive local neighborhood and a rays based global neighborhood that are able to express local or global relations respectively. We then summarize the responses of the classifier over each neighborhood by computing a new histogram based feature. These features are then used to train a second classifier which is again used to classify each pixel,

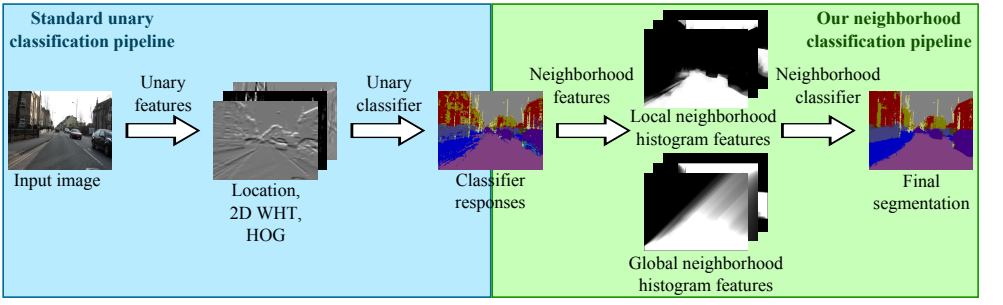

Figure 1: Pipeline of the method. The blue part shows the standard unary classification process, while the green part shows the neighborhood classification pipeline.

but in contrast to the first one, it integrates local and global constraints from the neighborhood features and is therefore able to improve the results of the first classifier significantly.

Our contribution is threefold. Firstly, we introduce a classification framework based on the concept of pixel neighborhoods, which captures structure constrains with a new histogram based neighborhood feature. Secondly, we propose a novel way to use the geodesic distance to compute the local pixel neighborhood. Thirdly, we introduce a new global rays based neighborhood, again using the geodesic distance, that can also capture global context.

We evaluate our method on three widely used and very challenging datasets: CamVid [1, 2], MSRC-21 [19] and the Stanford background dataset [6]. We analyze the performance of the different parts of our model and show how they contribute to increase the segmentation performance. Furthermore, we compare to two well known and strongly related methods: auto-context [25] and the robust P^n model of [8] and show an increase in performance.

2 Related work

We divide the related work on multi-class segmentation in two broad categories: energy minimization and classifier based methods. While there is a vast amount of literature on this topic, here we focus on the methods that are mostly related to our.

2.1 Energy minimization based methods

Modeling the problem of multi-class segmentation in a probabilistic framework based on a MRF or a CRF [13] is one of the commonly used approaches in the literature. Although relatively simple pairwise CRF models like TextonBoost [20] can already provide good results, they are limited to modeling only simple pixel relations. Higher-order CRFs overcome this limitation by considering larger groups of variables simultaneously, but parameter estimation and inference become computationally expensive. The Robust P^n model of [8] introduces a specific new type of higher-order potentials, based on presegmented image regions, for which inference is tractable. However, learning the CRF parameters of the Robust P^n model requires exhaustive search on a validation dataset. Several other methods build on the Robust P^n model and extend it to include 3D information [22], object detectors [11] or build region hierarchies [10, 12]. While we use a similar approach as the Robust P^n model to capture

the local context, our global neighborhood allows us to also learn long range scene context, which leads to improvement in many cases.

One of the general problems with CRF models is that the resulting energy usually has several parameters (for example a weighting between the unary, pairwise and higher-order potentials) that need to be learned in a parameter estimation step. This is not only a very difficult problem, requiring significant approximations to become tractable, but also has the disadvantage that one set of parameters is learned and fixed for the whole dataset and does not depend on the current image to be segmented. Decision tree fields [16] and the regression tree fields [7] address this problem by training a tree containing different sets of parameters that depend on the image appearance. While those models are powerful, they are more complex and computationally expensive.

Our method, in contrast, implicitly learns the balance between the different input terms during training and does not require separate algorithms for parameter estimation and inference. Furthermore, energy minimization methods can only learn pixel relations that fit the form of specific potential functions, which are usually simple in order to be tractable. Our classifier does not have this limitation and can learn more complex relations.

2.2 Classifier based methods

Another approach to multi-class image segmentation is to directly model the problem as the prediction of a classifier. Since a single classifier trained to predict the label of individual pixels usually produces very noisy results, a commonly used idea is to train a sequence of classifiers in which each classifier uses the predictions of the previous one and integrates more information into the features.

In the semantic texton forests model of [21], the authors first train a random forest based on simple image features and then train a second forest that takes as input local rectangular features computed on the output of the first one.

The stacked hierarchical learning of [15] builds on the idea of stacking [3] and employs a hierarchy of regions where the classifier at each level uses the predictions from the previous level. The regions hierarchy ranges from big regions (the whole image) to small regions (superpixels) and each classifier predicts the proportion of labels of each region.

The concept of entangled forest [14] is also related to our work. For several levels of the decision trees, predictions from the parent node are added to the feature pool together with image features. The GeoF model of [9] builds on top of this model by adding geodesically smoothed versions of the parent node's predictions in order to make better use of the scene structure. Note that we use the geodesic distance in a very different way than GeoF, where the geodesic distance transform is defined over big probabilistic regions and is incorporated in the features of the unary classifier. In our model, in contrast, we use the geodesic distance to find a set of related pixels for each pixel in the image to build a new feature and train a separate classifier.

Another strongly related method is auto-context [25]. The authors propose to train a series of classifiers such that each classifier uses both image features and the predictions from the previous one. The main focus of auto-context is to automatically learn context relationships, by sampling features and predictions from fixed locations around each pixel. Our model is also able to capture global context, but instead of using fixed positions, our geodesic neighborhoods are able to adapt to the image structure and align to the object boundaries, which makes the input to the classifier more reliable.

2.3 Combined methods

The inference machines method of [17] is a combination of both approaches - a classifier is used to learn the messages passed during belief propagation inference, which is typically used in graphical models. The work of [18] extend the inference machines by allowing the model to additionally learn spatial semantic context. The authors propose to define several source regions for each point in a 3D point cloud that cover larger portions of the scene. These source regions are somewhat similar to our rays based neighborhood, but are integrated in a very different way in the model. Furthermore, their shape is fixed and they do not align to the scene structure, like our geodesic neighborhoods.

3 Method

In this section we describe in detail our neighborhood classification framework and how to define local and global neighborhoods based on the geodesic distance. We also provide details about the classifiers and the features used in our experiments.

3.1 Multi-class image labeling

The multi-class image labeling problem is typically formulated in a probabilistic framework by assigning two random variables x_i and y_i to each pixel. The variable x_i encodes the appearance of the pixel in the image. The variable y_i takes values from a set of classes $\mathcal{L}$ and denotes the label of the pixel (e.g. car, street or building). The optimal labeling is found by maximizing the conditional probability $P(\mathbf{y}|\mathbf{x})$ of the labeling $\mathbf{y}$ given the image $\mathbf{x}$.

There are several ways to model this probability distribution. The simplest way is to assume that the label of each pixel is independent on the labels of all other pixels. In this case the joint probability distribution becomes the product of the distributions of each pixel taken individually: $P(\mathbf{y}|\mathbf{x}) = \prod_i P(y_i|x_i)$. While the probability distribution of each pixel can be directly estimated by a classifier, this independence assumption is very strong and usually does not hold in practice and therefore the results are typically very noisy.

A more sophisticated way to model the probability distribution $P(\mathbf{y}|\mathbf{x})$ is to provide additional constraints on groups of two or more pixels. Such constraints are typically modeled as a CRF in which inference can be done using energy minimization algorithms.

3.2 Neighborhood classification framework

In this paper we propose a classification framework that models the dependence of each pixel on a set of related pixels called a neighborhood (see Fig. 1). First, we estimate the probability distribution $P_U(y_i|f_U(x_i))$ of each pixel individually based on features $f_U(x_i)$ computed from its appearance in the image. The distribution is estimated by a unary classifier trained under the assumption that all pixels are independent on each other. This process is very similar to the standard way to define the unary potentials in CRF based methods [19].

After we have the responses of the unary classifier, we define the neighborhood N_i of pixel i to be a set of pixels related to it in a certain way. We then train a second classifier, the neighborhood classifier, which operates on features $f_N(N_i, P_U(y_{N_i}|f_U(x_{N_i})))$ that summarize the responses of the unary classifier for the corresponding neighborhood with a

Local $\gamma = 0$ | Local $\gamma = 50$ | Local $\gamma = 1000$ | Global $\gamma = 1000$

Figure 2: Visualization of the shapes of the presented neighborhoods for selected pixels (marked in black). Note that every ray generates its own separate neighborhood, but on the last image we visualize all of them together.

histogram based neighborhood feature. The neighborhood classifier estimates the probability distribution $P_N(y_i|f_N(N_i, P_U(y_{N_i}|f_U(x_{N_i}))))$ under the assumption that the label of pixel i is independent of the labels of the other pixels given its neighborhood N_i.

3.3 Pixel neighborhoods

As described above, the neighborhood N_i of an image pixel i is a subset of pixels that are related in certain way to this pixel. Below we introduce two ways to define the neighborhood that are able to capture local and global context respectively.

3.3.1 Local neighborhood

We define the *local neighborhood* N_i as the set of the n pixels closest to pixel i according to the geodesic distance. The geodesic distance extends the euclidean distance with a term that considers the gradients in the image intensities. If two point pairs have the same euclidean distance, but there is an edge in the image between the points in the first pair then they will have a higher geodesic distance. For the experiments in this paper we use $n = 200$. We define the geodesic distance between two pixels i and j similarly to [9]:

$$d(i,j) = \inf_{\mathbf{G} \in \mathcal{P}_{i,j}} \int_0^l \sqrt{1 + \gamma^2 (\nabla I \cdot \mathbf{G}'(\mathbf{s}))^2} ds, \tag{1}$$

where $\mathcal{P}_{i,j}$ is the set of all possible paths between pixels i and j, $\mathbf{G}$ is a path from this set with length l and $\mathbf{G}'$ is its spatial derivative. The parameter γ indicates the weight between the image gradient and the spatial distance between the two pixels. For $\gamma = 0$ the geodesic distance becomes equivalent to the euclidean distance, while for $\gamma = 1000$ it is dominated by the image gradients (see Fig. 2).

In order to find the closest n neighbors with respect to the geodesic distance, we propose an algorithm based on the algorithm of Dijkstra for finding the shortest paths between a fixed point and all other points in a graph, that in our case is the 4-connected grid of the image pixels. In practice, we sample the pixels on a 3 by 3 grid in order to cover bigger portions of the image with a smaller number of points. For a formal description of the algorithm please see the supplementary material.

While this algorithm computes an approximation of the geodesic distance, it delivers good results given the discretization of the image. Note that although there are some methods for fast computation of the geodesic distance transform like [23], [27], they focus on computing the distance from a point to a whole region and are therefore unpractical for computing the distance to all points individually.

3.3.2 Global neighborhood

The local neighborhood only covers a relatively small area around the pixel of interest and thus can only provide local context. In order to capture long range relations, we introduce a global neighborhood. From the pixel of interest we shoot 8 rays in directions equally spaced at 45° to the borders of the image (see Fig. 3). Here, we again make use of the geodesic distance, by taking the union of the geodesic neighborhoods of all points on the ray. As illustrated on Fig. 2, the neighborhood is able to adapt to the image structure along the rays and therefore to deliver more reliable information. We define one separate neighborhood for each ray, resulting in 8 neighborhood sets for each pixel.

3.4 Features and classifiers

To train both the unary and the neighborhood classifier we employ the JointBoost method [24]. However, our method is not dependent on a particular classifier and every other multi-class classifier could be used. We use the same parameters for all datasets and all experiments in the paper. In fact, the parameters of the unary and the neighborhood classifiers differ only in the iterations count, which has an effect on the training time, but not on the accuracy.

In order to speed up the training, we use some standard techniques similar to [12, 20]. We subsample the training images in a regular grid of size 5. At each boosting iteration we randomly choose 30% of the features and for each feature we perform 100 comparisons with thresholds sampled uniformly between the minimum and the maximum value of the feature across all training samples. We perform 5000 boosting iterations for the unary classifier, but only 1000 iterations for the neighborhood classifier, because doing more iterations leads to a marginal improvement, while being 5 times slower both at training and at evaluation time.

In order to compensate for the big difference in the number of samples in the different classes, we adjust the weight of each sample as if the number of the samples for each class is the same. If we do not do this the classifier ignores classes with a small number of samples like people or signs and focuses on the big classes like the sky, buildings and roads. Note that this measure leads to improved average per class accuracy, but decreases the global accuracy.

3.4.1 Unary features

The unary features are computed directly from the image. We use histograms of oriented gradients [4] and the 2D Walsh-Hadamard transform [6], which are both fast to compute.

The HOG features [4] are computed in a window of size 5 around every pixel and divided into 16 directional bins. This gives us a 16 dimensional feature vector for every pixel, containing the gradients for each direction. The 2D Walsh-Hadamard transform [6] is a discrete approximation of the cosine transform and is very fast to compute. Similarly to [26], we first convert the image in the Lab color space. Then the first 16 coefficients of the Walsh-Hadamard transform are computed in windows of 2, 4, 8, 16 and 32 pixels around every pixel separately for each color channel. In this way we have 48 features for each scale, except for the window of size 2, because in this case the Walsh-Hadamard transform has only 4 coefficients, resulting in a feature vector with 204 elements. Additionally we add the normalized 2D coordinates of each pixel in the image as feature in order to encode location context. For the final feature vector we just stack the 2 feature vectors described above and the 2 coordinates of the point into one vector of size 222.

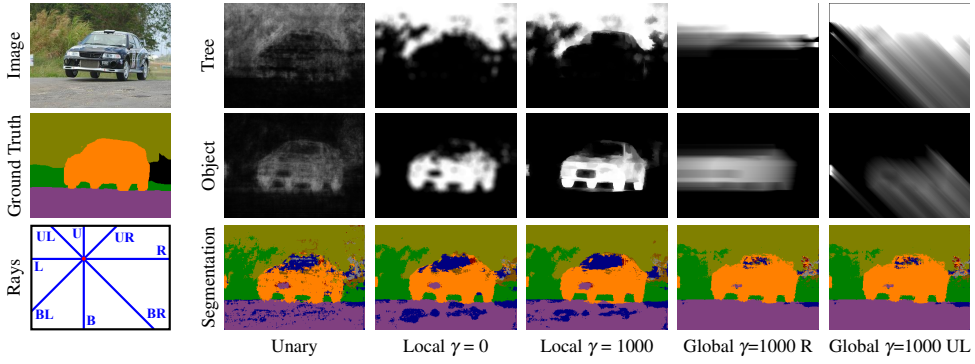

Figure 3: Histogram based neighborhood features. We show the bins corresponding to the classes TREE and OBJECT as a probability map, the segmentation from the neighborhood classifier for the local and global neighborhoods and the raw unary responses. The last two columns show two of the rays of the same global neighborhood. A high value for a pixel in the global neighborhood of a ray means that there is a region of this class in this direction.

3.4.2 Neighborhood features

The goal of the neighborhood features is to summarize the information from the unary classifier over a given neighborhood. We employ a voting approach, in which every pixel in the neighborhood votes for the most likely label given the unary classifier responses. Then we compute a normalized histogram over $\mathcal{L}$ from the votes of all pixels in the neighborhood and use this histogram directly as a feature vector. The values of the histogram for a given pixel can actually be interpreted as a probability distribution since they are normalized and this can be seen in Fig. 3. However, the final segmentation is not determined according to this distribution, but by the neighborhood classifier. In the case that we have multiple neighborhoods we just stack the histograms of all pixels together into one big feature vector. We also add the probability distribution of the pixel itself computed by the unary classifier.

This is a very compact representation of the responses of the unary classifier, because we have only $|\mathcal{L}|$ feature values per neighborhood. Therefore, the size of the feature depends linearly on the number of classes and is constant with respect to the size of the neighborhood. This makes the training extremely fast, while we are still able to achieve good results.

4 Evaluation

We evaluate our method on 3 widely used semantic segmentation datasets: CamVid [1, 2], MSRC-21 [19] and the Stanford background dataset [8].

The **CamVid** dataset [1, 2] consists of image sequences taken with a camera mounted behind the windshield of a moving car. We adopt the protocol from [2, 11, 22] by using the same dataset split consisting of 367 images for training 233 for testing, we scale all images to a resolution of 320×240 and use only the 11 classes with most instances.

The **MSRC-21** dataset [19] contains images of nature and indoor scenes labeled pixel-wise in 21 classes (the classes MOUNTAIN and HORSE are commonly ignored). We use the dataset split of [19] consisting of 276 training and 256 test images.

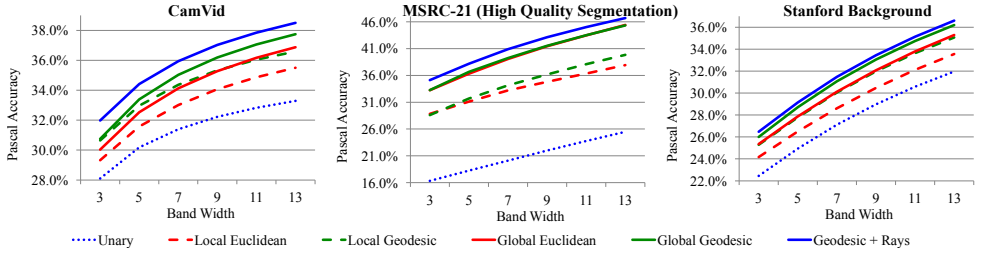

Figure 4: Pascal accuracy evaluated in a band around the object boundaries as in [8].

The **Stanford Background** dataset [5] consists of 715 images of outdoor scenes with at least one foreground object. The images are labeled in 8 classes. As in [5] we divide the dataset randomly in 572 training and 143 testing images and perform 5 fold cross validation.

We analyze the performance of the neighborhoods and compare our method to two related approaches. For the quantitative evaluation we adopt the 3 standard evaluation measures: the global pixel-wise accuracy, the average per class accuracy and the average per class intersection over union measure (Pascal accuracy). We provide quantitative results in Table 1 and result images for qualitative evaluation in Fig. 5 and in the supplementary material. Furthermore, we evaluate the segmentation around the object boundaries, by defining a band of variable width around them and evaluating only on this region as in [8] (see Fig. 4).

4.1 Local neighborhoods

In this section we analyze the performance of the local neighborhood for two different values of the weight between the euclidean and the geodesic term γ. For $\gamma = 0$ (*Local Euclidean*) the neighborhood does not adapt to the image structure and just takes all points in a certain radius, while for $\gamma = 1000$ (*Local Geodesic*) the neighborhood aligns to the image gradients. The results show that using the local neighborhoods with our method gives a significant improvement over the unary classifier on all datasets. This is due to the fact that in the neighborhood features each pixel receives information from a bigger patch of pixels related to it and is able to correct many of the errors of the unary classifier.

We can observe that the *Local Geodesic* neighborhood performs better then *Local Euclidean*. While the difference in the numbers may not seem big, the latter adapts much better to the object edges where a strong image gradient is present (see Fig. 4). This effect is also illustrated in Fig. 3 - for the *Local Euclidean* neighborhood a pixel close to an object boundary will receive support not only from the object it belongs to, but also from nearby objects, which leads to blurred borders of the neighborhood features and therefore worse segmentation around borders. The *Local Geodesic* neighborhood, on the other hand, adapts to the image structure and provides more consistent information around the object boundaries.

4.2 Global neighborhoods

The local neighborhoods consider only the information over a local patch which is problematic if a bigger region of the object is labeled wrongly by the unary classifier. Our global approach, on the other hand, integrates information from other parts of the image in order to resolve this problem. Here, we again evaluate the global neighborhoods for two values of γ

Method	CamVid			MSRC-21			Stanford Background		
	Global	Average	Pascal	Global	Average	Pascal	Global	Average	Pascal
Unary	70.6	56.4	35.7	61.4	50.4	33.4	66.6	64.2	46.0
Local Euclidean	74.7	61.2	40.0	70.5	64.3	45.2	70.4	68.1	49.7
Local Geodesic	74.7	62.1	40.4	71.8	65.8	47.1	71.3	68.3	50.5
Global Euclidean	75.3	62.3	41.1	76.1	**70.8**	53.6	72.3	69.8	51.7
Global Geodesic	76.2	63.0	41.8	76.1	**70.8**	53.7	72.7	69.9	52.0
Local+Global Geodesic	76.8	**63.5**	**42.4**	**76.3**	**70.8**	**53.9**	**72.8**	**70.4**	**52.2**
Auto-context	74.5	61.7	40.5	72.5	67.3	49.4	72.0	69.4	51.5
Robust P^n	**77.9**	56.0	40.5	73.4	65.0	47.7	71.9	67.9	50.8

Table 1: Quantitative evaluation on CamVid, MSRC-21 and Stanford Background.

(0 and 1000) and again observe that using the geodesic distance to adapt the neighborhood to the image structure is beneficial both in terms of overall performance and in accuracy around the object edges. Combining the local and the global geodesic neighborhoods (*Local+Global Geodesic*) further improves the results, especially around the object boundaries.

It is interesting to see that the improvement from the global neighborhood is especially strong on the MSRC-21 dataset. Most of the images in it contain only 2 or 3 of the 21 classes which are usually strongly related. For example cows and trees appear surrounded by grass and sky and almost never by water. Using the global neighborhood, our method is able to learn those context relations and resolve many of the errors of the unary classifier (see Fig. 5). In the other two datasets the benefit is smaller (but still significant), because all classes appear in almost all of the images together and the global context relations are weaker.

4.3 Comparison to related methods

We compare the best version of our method (*Local+Global Geodesic*) to two closely related methods: auto-context [25] and the robust P^n model [8]. Since the performance of those methods is strongly dependent on the unary classifier, we use the same image features and the same classifier as in our method to allow for a fair comparison.

For **Auto-context** we implemented the same fixed sampling scheme as described in [25] and used the same classifiers as in our method, but giving them access to both the probability distributions of the previous classifier and the image features. Since we train two classifiers we also perform two iterations of auto-context. As the authors of [25] point out in their analysis, after the second iteration the performance of the method saturates quickly.

On the MSRC-21 dataset sampling points at large distances gives auto-context an advantage over our local models, but our combined local and global model is able to capture the long range context better. On the other datasets, however, the global context is a weaker cue and because auto-context has a fixed sampling scheme, it cannot adapt to the local image structure which leads to worse performance than our model.

The **Robust P^n model** [8] is also strongly related to our method, because it also integrates the information over bigger segments in order to provide consistent segmentation. We use the code provided online by the authors, but plug in our own unary potentials. We use the same parameters for the unsupervised mean-shift segmentation for MSRC-21 as specified in [8] and the suggestions provided by the authors for the other two datasets. We also use the heuristic provided in the original paper to train the parameters of the CRF model.

From the quantitative results we see that the Robust P^n model performs well according to the global accuracy measure on the CamVid dataset, but our method is still able to out-

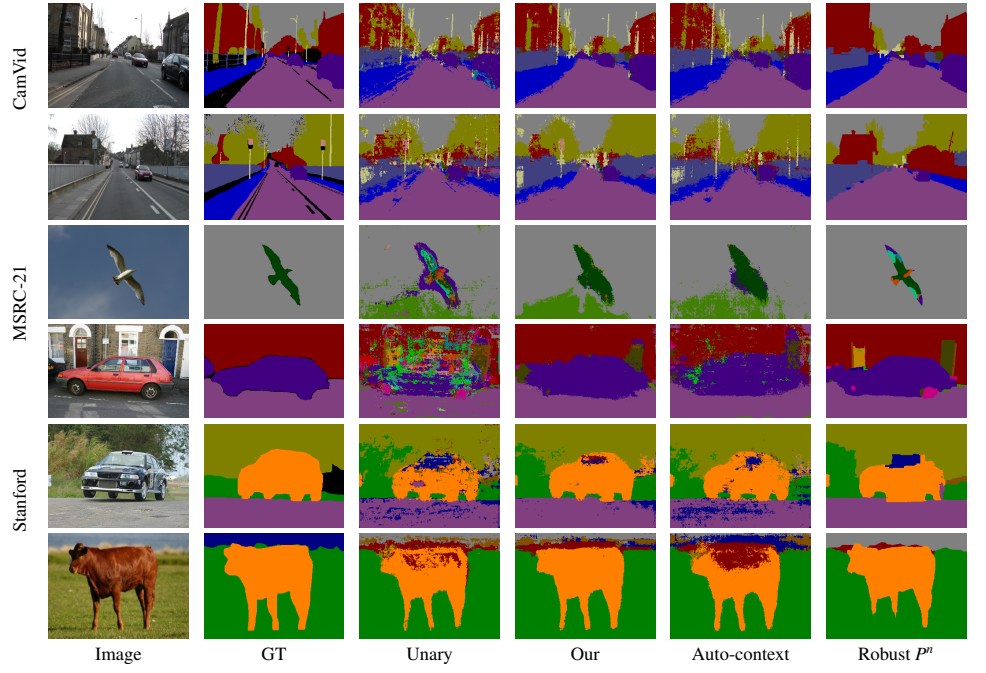

Figure 5: Result images from our method (Local+Global Geodesic) and the two related methods we compare to: auto-context and the Robust P^n model.

perform it on the average accuracy and the Pascal measure. The reason for this is that, while some regions are well segmented, many of the smaller objects tend to be ignored (see Fig. 5). On the other two datasets, our method is actually able to outperform the Robust P^n model on all measures, while we again observe big differences in the average per class accuracies.

5 Conclusion

We formulate the problem of multi-class image segmentation in a new classification framework. We show how pixel neighborhoods can be used to integrate local and global context relations as a feature in a standard classification process and we also use the geodesic distance in image intensity space in a novel way in order to allow the neighborhoods to adapt to the image structure, which improves the segmentation especially around object boundaries.

References

[1] Gabriel J. Brostow, Julien Fauqueur, and Roberto Cipolla. Semantic object classes in video: A high-definition ground truth database. *Pattern Recognition Letters*, 2008.

[2] Gabriel J. Brostow, Jamie Shotton, Julien Fauqueur, and Roberto Cipolla. Segmentation and recognition using structure from motion point clouds. In *ECCV*, 2008.

[3] William W. Cohen and Vitor R. Carvalho. Stacked sequential learning. In *IJCAI*, 2005.

[4] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[5] S. Gould, R. Fulton, and D. Koller. Decomposing a scene into geometric and semantically consistent regions. In *ICCV*, 2009.

[6] Y. Hel-Or and H. Hel-Or. Real time pattern matching using projection kernels. *ICCV*, 2003.

[7] Jeremy Jancsary, Sebastian Nowozin, Toby Sharp, and Carsten Rother. Regression tree fields - an efficient, non-parametric approach to image labeling problems. In *CVPR*, 2012.

[8] P. Kohli, L. Ladicky, and P. H. S. Torr. Robust higher order potentials for enforcing label consistency. In *IJCV*, 2009.

[9] Peter Kontschieder, Pushmeet Kohli, Jamie Shotton, and Antonio Criminisi. GeoF: Geodesic forests for learning coupled predictors. In *CVPR*, 2013.

[10] L. Ladický, C. Russell, P. Kohli, and P. H. S. Torr. Associative hierarchical crfs for object class image segmentation. In *ICCV*, 2009.

[11] L. Ladický, P. Sturgess, K. Alahari, C. Russell, and P. H. S. Torr. What, where and how many? Combining object detectors and CRFs. In *ECCV*, 2010.

[12] Lubor Ladický, Chris Russell, Pushmeet Kohli, and P. H. S. Torr. Associative hierarchical random fields. In *PAMI*, 2013.

[13] J. D. Lafferty, A. McCallum, and F. C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*, 2001.

[14] Albert Montillo, Jamie Shotton, John Winn, Juan Eugenio Iglesias, Dimitri Metaxas, and Antonio Criminisi. Entangled decision forests and their application for semantic segmentation of CT images. In *Information Processing in Medical Imaging*, 2011.

[15] Daniel Munoz, J. Andrew Bagnell, and Martial Hebert. Stacked hierarchical labeling. In *ECCV*, 2010.

[16] Sebastian Nowozin, Carsten Rother, Shai Bagon, Toby Sharp, Bangpeng Yao, and Pushmeet Kohli. Decision tree fields. In *ICCV*, 2011.

[17] S. Ross, D. Munoz, M. Hebert, and J. A. Bagnell. Learning message-passing inference machines for structured prediction. In *CVPR*, 2011.

[18] R. Shapovalov, D. Vetrov, and P. Kohli. Spatial inference machines. In *CVPR*, 2013.

[19] J. Shotton, J. Winn, C. Rother, and A. Criminisi. TextonBoost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *ECCV*, 2006.

[20] Jamie Shotton, John Winn, Carsten Rother, and Antonio Criminisi. TextonBoost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context. IJCV, 2007.

[21] Jamie Shotton, Matthew Johnson, and Roberto Cipolla. Semantic texton forests for image categorization and segmentation. In *CVPR*, 2008.

[22] Paul Sturgess, Karteek Alahari, Lubor Ladicky, and Philip H. S. Torr. Combining appearance and structure from motion features for road scene understanding. In *BMVC*, 2009.

[23] P. J. Toivanen. New geodesic distance transforms for gray-scale images. In *Pattern Recognition Letters 17*, 1996.

[24] A. Torralba, K. P. Murphy, and W. T. Freeman. Sharing visual features for multiclass and multiview object detection. In *PAMI*, 2007.

[25] Zhuowen Tu and Xiang Bai. Auto-context and its application to high-level vision tasks and 3D brain image segmentation. In *PAMI*, 2010.

[26] C. Wojek and B. Schiele. A dynamic conditional random field model for joint labeling of object and scene classes. In *ECCV*, 2008.

[27] Liron Yatziv, Alberto Bartesaghi, and Guillermo Sapiro. O(N) implementation of the fast marching algorithm. In *Journal of Computational Physics 212*, 2006.