Learning Where to Classify in Multi-view Semantic Segmentation

Hayko Riemenschneider[1], András Bódis-Szomorú[1], Julien Weissenberg[1], and Luc Van Gool[1,2]

[1] Computer Vision Laboratory, ETH Zurich, Switzerland
[2] K.U. Leuven, Belgium
{hayko,bodis,julienw,vangool}@vision.ee.ethz.ch

Abstract. There is an increasing interest in semantically annotated 3D models, e.g. of cities. The typical approaches start with the semantic labelling of all the images used for the 3D model. Such labelling tends to be very time consuming though. The inherent redundancy among the overlapping images calls for more efficient solutions. This paper proposes an alternative approach that exploits the geometry of a 3D mesh model obtained from multi-view reconstruction. Instead of clustering similar views, we predict the best view before the actual labelling. For this we find the single image part that bests supports the correct semantic labelling of each face of the underlying 3D mesh. Moreover, our single-image approach may surprise because it tends to increase the accuracy of the model labelling when compared to approaches that fuse the labels from multiple images. As a matter of fact, we even go a step further, and only explicitly label a subset of faces (e.g. 10%), to subsequently fill in the labels of the remaining faces. This leads to a further reduction of computation time, again combined with a gain in accuracy. Compared to a process that starts from the semantic labelling of the images, our method to semantically label 3D models yields accelerations of about 2 orders of magnitude. We tested our multi-view semantic labelling on a variety of street scenes.

Keywords: semantic segmentation, multi-view, efficiency, view selection, redundancy, ranking, importance, labeling.

1 Introduction

Multi-view 3D reconstructions are common these days. Not only have tourist data become ubiquitous [1, 2] but the images also often result from deliberate mobile mapping campaigns [3–6]. The images have to exhibit sufficient redundancy – overlap – in order to be suited for Structure-from-Motion (SfM) and Multi-View Stereo (MVS) reconstruction. In the meantime, solutions have been worked out to keep the number of images within bounds, primarily for making the reconstruction pipelines applicable to larger scenes. For instance, the redundancy can be captured by measuring visual similarity between images, and the scene can be summarized, e.g. by constructing a graph of iconic views [2].

D. Fleet et al. (Eds.): ECCV 2014, Part V, LNCS 8693, pp. 516–532, 2014.

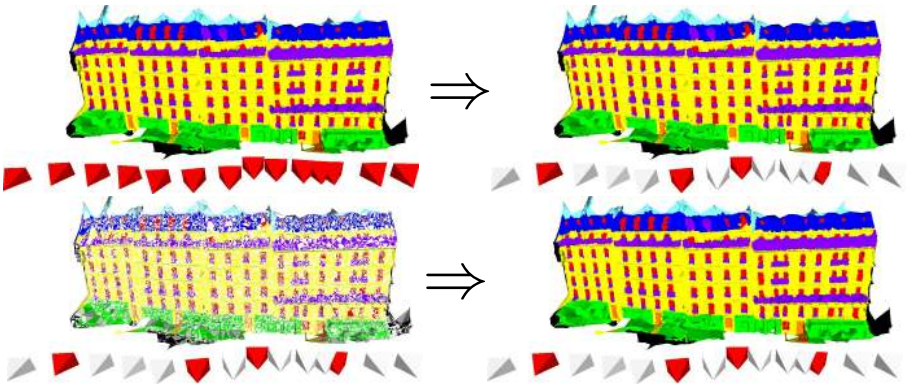


Fig. 1. View overlap is ignored by existing work in semantic scene labelling, and features in all views for all surface parts are extracted redundantly and expensively (top left). In turn, we propose a fine-grained view selection (top right), as well as to reduce scene coverage (bottom left) by only classifying regions essential in terms of classification accuracy. The labels of the classified regions are then spread into all regions (bottom right). This sparsity increases efficiency by orders of magnitude, while also increasing the accuracy of the final result (bottom right vs. top left).

In the aftermath of SfM/MVS reconstruction processes arise recent efforts to make these 3D models widely applicable. An important step in that direction is to augment the models with semantic labels, i.e. to identify parts of the 3D data to belong to certain object classes (e.g. building, tree, car, etc), or object part classes (e.g. door, window, wheel, etc). Typically, the semantic labelling is carried out in all the overlapping images used for 3D reconstruction [7, 8]. This implies that many parts of the scene get labeled multiple times, resulting in a large computational overhead in the order of the redundancy of the image set. The runtime of semantic classification pipelines still lies between 10 s and 300 s per image [8]. Worse, these speeds are reported for moderately sized images of 320×240 pixels, and not for the high-resolution megapixel-sized images common for SfM. The bottleneck of redundant labelling is not in the classification step [9–11], but rather in feature extraction and description. Also, an extra step is needed after labelling the images, namely, to fuse the different labels of the same 3D patch in order to obtain a consistently labelled model.

We propose an alternative strategy to semantically label the 3D model. We start by producing the mesh model and then determine for each of its faces which *single* image is best suited to well capture the true semantic assignment of the face. Not only do we avoid to needlessly process a multitude of images for the same mesh face, but we also have the advantage that we can exploit both geometry (3D model) and appearance (image). Moreover, the accuracy of the semantic labelling will be shown to improve over that of multi-view labelling.

A somewhat similar problem is known from texture mapping or image-based rendering. There decisions have to be made about which image to use to render the local appearance of the model. As to avoid the texture getting blurred, it is

also quite usual to look for the best source image among a set of possibilities. Most methods use criteria that are related to the size of the model patch in the image and the degree to which the view is orthogonal to the patch. One may expect to find the same criteria to dominate the choice in segmentation as well, but that intuition is misleading for our application, as we will also show.

On top of selecting a single view to get each face's label from, we speed the process up further by not providing explicit classification for all the faces. We will demonstrate that it suffices to do this for about 30% of the faces, whereas all remaining labels can be inferred from those that were extracted. Moreover, this second parsimony again increases the accuracy of labelling.

We demonstrate our semantic labelling approach for different street scenes. Yet the core of our method is general and can be applied to different types of scenes and objects. In keeping with the central goals of the paper, we achieve a speedup with about two orders of magnitude while improving the label accuracy. In summary our contributions are the following.

1. An alternative approach is proposed for multi-view semantic labelling, efficiently combining the geometry of the 3D model and the appearance of a single, appropriately chosen view - denoted as reducing view redundancy.
2. We show the beneficial effect of reducing the initial labelling to a well-chosen subset of discriminative surface parts, and then using these labels to infer the labels of the remaining surface. This is denoted as scene coverage.
3. As a result, we accelerate the labelling by two orders of magnitude and make a finer-grained labelling of large models (e.g. of cities) practically feasible.
4. Finally, we provide a new 3D dataset of densely labelled images.

2 Related Work

The research in the field of semantic segmentation has enjoyed much attention and success in the last years (+17% in 5 years on PASCAL [12]). Yet most semantic segmentation approaches still rely on redundant independent 2D analysis. Only recently some dived into the 3D realm and exploit joining the domains.

In the 2D domain, the initial works dealt mostly with feature description and learning. [13] introduced TextonBoost which exploits multiple texture filters with an effective boost learning algorithm. [14] uses the output of the trained classifier as new feature input for training several cascades. Additional works included higher-order terms [15, 16] and simplification by superpixels [17, 18]. Others focused on better graphical models [19, 20] or including detectors [7, 21].

None of the above focus on the scalability issue of large scenes and only operate on individual images. Pure 2D scalable semantic classification was addressed in [8], which reduces by nearest neighbor searching for images and superpixels.

For the 2D domain in streetside, where surfaces are more structured than in arbitrary scenes, fewer works have been carried out. [22] pioneered the feel for architectural scene segmentation. [23] carried out 2D classification with a generic image height prior. [24, 25] both used streetside object detectors on top

of local features to improve the classification performance. Yet classification is performed on 2D images. 3D is introduced only at a procedural level [26–28].

[29] exploit temporal smoothness on highway scenes. The idea is that redundant time-adjacent frames should be consistently labeled, where assumption is that between frames the motion is not too strong (always forward looking and high-frame rate) and scene content is redundantly present.

For the 3D domain in streetside, [5] were the first to combine sparse SfM and semantic classification. [3] interleaved 2.5D depth estimation and semantic labelling. In these lines [30] used dense 2.5D depth images for classification and [31] used semantic segmentation for deciding where to use 2.5D depth for plane fitting. [32] again worked only on sparse 3D data and yet provides a method for linking these different densities of the full 2D image and sparse 3D domain. [4] classified 2D images and then aggregated their labels to provide an overhead map of the scene. This uses a homography assumption to aggregate the birdseye map of the scenes. Most accuracy problems arise because of occlusions and averaging of multiple views. Recently, [33, 34] combined the creation of geometry with the semantic labelling implicitly evaluating all data redundantly.

Most related to our baseline are the works [9, 10] who used 3D meshes to directly label 3D scenes. This has the benefit of using 3D features and operating in one place to fuse the classification yet still requires description and classification. [10] showed how a common 3D classification can speed up the labelling over redundant 2D classification. [9] introduced decision tree fields for 3D labelling to learn which pairwise connections are important for efficient inference.

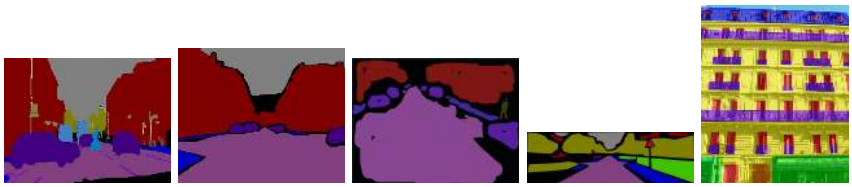
Yet in summary, all of the 3D semantic research uses all data redundantly. All images are fully analyzed, described and all its features classified.

Related work for the view selection has only been carried out on an image level. Before SfM, the visual graphs are analyzed and clustered for iconic scenes [1, 2] to split the data into coherent scene parts. After SfM, camera and geometry information are used to select clusters and non-redundant views - again only at the image level [35, 36].

Our work is inspired by the related world of 3D model texturing, where the goal is to find an optimal single texture file for a 3D model [37–39]. Usually, for finding the single best texture, the largest projection in terms of area size or most fronto-parallel view is used in addition to lighting constancy constraints.

We propose to change this paradigm and only analyze the most discriminative views of the data. To the best of our knowledge, we are the first to actively exploit this redundancy in a multi-view semantic labeling. Further, we propose a novel view to select the best such view by selecting the best view according to its ability to classify the scene correctly.

A further note on 3D datasets, most related work only shows examples on small outdoor scenes of single coarse buildings or small indoor scenes like the NYU 3D scenes [40]. The datasets for semantic streetside labeling consist of very few coarse labels (building, vegetation, road) and do not focus on the details of the scenes. For example, datasets like Leuven [3], Yotta [4], CamVid [5] and the KITTI [6] labelled for semantics by [10] only contain these coarse scenes



(a) CamVid (b) Leuven (c) Yotta (d) KITTI (e) Full428

Fig. 2. Dataset overview - most are coarsely labelled at low resolution. We use a pixel accurate labelling with fine details at 1-3 megapixel resolution. (rightmost).

labels, see Figure 2. Except for CamVid where there exist 700 accurately labelled ground truth images, the other datasets only contain coarsely labelled images (in order of user strokes) from 70 to 89 images for training and testing.

In this work, we move to finely detailed ground truth labels including building detail such as windows, doors, balconies, etc. Further, the dataset is used for SfM with high resolution images of 1-3 megapixels and pixel-accurate dense labels.

3 3D Surface and Semantic Classification

Our final goal is to label each part of the scene – a 3D mesh surface – by detailed semantic labels (wall, window, door, sky, road, etc). We briefly describe the multi-view reconstruction methods to obtain the surface, the cues for semantic scene labelling, and then dive into the multi-view scene labelling problem.

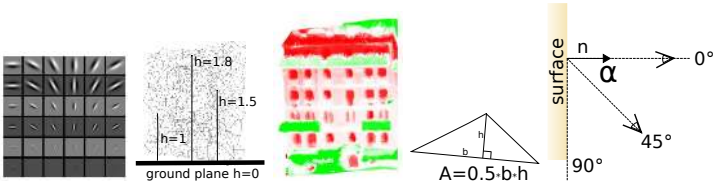
3.1 Multi-view Surface Reconstruction

Our input is a set of images which are initially fed to standard SfM/MVS algorithms to produce a mesh. SIFT features [41] are extracted and matched across the images, and reconstructed along with the cameras by using incremental bundle adjustment [42]. The estimated views are clustered and used to compute depth maps via dense MVS. Volumetric fusion is performed by tetrahedral partitioning of space over the obtained dense 3D point cloud, and by exploiting point-wise visibility information in a voting scheme [43, 44]. The final surface is recovered using a robust volumetric graph cuts optimization [45].

The output of the reconstruction procedure is the set of cameras $\mathcal{C} = \{c_j\}$ and a surface mesh $\mathcal{M}$, which consists of a set of 3D vertices, a set of face edges and a set of triangular faces $\mathcal{F} = \{f_i\}$. Since we are about to assign semantic labels to faces f_i, we will represent this mesh as a graph, where nodes correspond to mesh faces and edges correspond to face adjacencies.

3.2 Heavy vs. Light Features for Semantic Labelling

For semantic labelling, we extract simple 2D image and geometric features. The typical approach is to extract features for every location of every single image



(a) CIElab (b) MR8 (c) Height (d) Depth (e) Area (f) Angle+**n**

Fig. 3. Features like color and gradient filters are expensive since they are densely calculated in the entire image. Geometry-based are more light-weight. Extra features like denseSIFT should improve the baseline, yet are even heavier to calculate.

in the dataset. We deviate from this dense computational scheme to a sparse computation, which is a main contribution of this paper.

In contrast to related work [9, 10], we split the features into two sets. The first set consists of features that will take longer time to compute:

$$\mathcal{X}^{heavy} = (L^*, a^*, b^*, \mathbf{t}, h, d, \mathbf{n}), \tag{1}$$

This is a 16-dimensional feature vector containing the CIELAB Lab^* color components, 8 responses of the MR8 filter bank [46, 47] in vector $\mathbf{t}$, the height h defined as the distance from the ground plane, the depth d w.r.t. the dominant plane (e.g. facade plane), and the surface normal $\mathbf{n}$, shown in Figure 3. One could use additional features here, e.g. dense SIFT, etc. See [8, 10, 16] for inspiration.

To aggregate features over the projection of a face $f \in \mathcal{F}$ in any observing camera c, we use Sigma Points [48], which efficiently capture the first two statistical moments of the feature vectors.

The second set contains only lightweight features:

$$\mathcal{X}^{light} = (A_{2D}, A_{3D}, A_{2D}/A_{3D}, \alpha), \tag{2}$$

where A_{3D} is the area of a mesh face $f \in \mathcal{F}$, A_{2D} is the area of its 2D projection in a specific camera $c \in \mathcal{C}$, and α is the angle of observation of the face from c.

It should be emphasized that $\mathcal{X}^{heavy}$ relies on image content, whereas $\mathcal{X}^{light}$ relies on geometric information only. In practice, calculation of $\mathcal{X}^{light}$ takes only a fraction of the time (120 seconds for all 1.8 million faces and 428 camera views vs. 21+ hours needed to calculate $\mathcal{X}^{heavy}$ for the Full428 dataset).

3.3 Multi-view Optimization for 3D Surface Labelling

We define a mesh graph $\mathcal{G}_{\mathcal{M}} = (\mathcal{F}, \mathcal{E})$, where the nodes represent the triangular faces $\mathcal{F} = \{f_i\}$ of the surface mesh $\mathcal{M}$, and $\mathcal{E}$ is the set of graph edges, which encode 3D adjacencies between the faces. We aim to assign a label x_i from the set of possible semantic labels $\mathcal{L} = \{l_1, l_2, \ldots, l_L\}$ to each of the n faces f_i. A possible complete labelling of the mesh is denoted by $x = (x_1, x_2, \ldots, x_n)$.

A Conditional Random Field (CRF) is defined over this graph and we aim to find the Maximum-A-Posteriori (MAP) labelling x^* of the surface mesh $\mathcal{M}$. This is equivalent to an energy minimization problem of the general form

$$x^* = \underset{x \in \mathcal{L}^n}{\operatorname{argmin}} E(x),$$

which we solve by efficient multi-label optimization, namely, the alpha-expansion graphcuts [49–51]. Our energy consists of unary data terms for every face f_i, and pairwise regularity terms for every pair of adjacent faces (f_i, f_j).

$$E(x) = \sum_{f_i \in \mathcal{F}}^{n} \sum_{c_j \in \mathcal{C}} \Theta(f_i, c_j, x_i) + \lambda \cdot \sum_{(f_i, f_j) \in \mathcal{E}} \Psi(f_i, f_j, x_i, x_j) \tag{3}$$

where $\sum_{c_j} \Theta(f_i, c_j, x_i)$ is the potential (penalty) for face f_i obtaining label x_i. $\Theta(f_i, c_j, x_i)$ is a per-view subterm, which relies on the single specific projection (an observation) of face f_i into view c_j. It can be written as the log-likelihood

$$\Theta(f_i, c_j, l) = -\log p(l \mid \mathcal{X}_{ij}), \tag{4}$$

where $\mathcal{X}_{ij} = \mathcal{X}(f_i, c_j)$ denotes the feature vector associated to the projection of face f_i into camera c_j, and $p(l \mid \mathcal{X}_{ij})$ is the likelihood of label $l \in \mathcal{L}$ for this particular projection of the face. In our scenario, the likelihoods $p(l \mid \mathcal{X})$ are provided by a random forest classifier trained on ground truth labels using the features described in Section 3.2.

The pairwise potential $\Psi(f_i, f_j, x_i, x_j)$ in Eq. 3 enforces spatially smooth labelling solutions over the mesh faces by penalizing occurrences of adjacent faces f_i and f_j obtaining different labels $(x_i \neq x_j)$. We use a Potts model

$$\Psi(x_i, x_j) = \begin{cases} 0 & \text{if} \quad x_i = x_j \\ \nabla & \text{if} \quad x_i \neq x_j \end{cases}, \tag{5}$$

where $\nabla = 1$ is a constant penalty. In the future, we plan to weight ∇ in function of the dihedral angles or plane distances between neighboring faces.

The coefficient λ in Eq. 3 controls the balance between unary and pairwise, data and smoothness terms. A grid search showed that $\lambda = 0.5$ works best.

Now for the fun part, it should be emphasized that each triangle f_i is typically observed from multiple cameras c_j. This redundant set of observations poses a computational challenge when extracting the feature vectors $\mathcal{X}(f_i, c_j)$ over all views c_j and for each face f_i. In the classical formulation, every view is considered and the final unary potential is aggregated over all views (see the second sum over the camera set $\mathcal{C}$ in the unary term of Eq. 3). In our findings, this is unnecessary. In the following section we describe our model of view importance and how it can be used to reduce the redundant set of views to the single most discriminative view for a more efficient semantic scene classification.

Fig. 4. Geometric link between 3D model and 2D image space. Contrary to related work in view clustering, we look for the best view $c^*(f_i)$ per mesh triangle f_i. For small viewing angles the texture is visually pleasing but not best for semantic classification.

4 Multi-view Observation Importance

In a multi-view scenario redundancy is inherent due to the view overlaps needed for SfM/MVS. Prior work ignored the relationship between these views. In turn, we start by defining two characteristics of the computational burden.

First, **view redundancy** R_i is the number of redundant camera views a mesh face f_i is observed in. See the top of Figure 7 and Table 1 for some typical average view redundancy values. Each triangle of the scene is visible in up to 50 cameras ($\bar{R} = 49$) on average! We aim for zero view redundancy ($R_i \equiv 0, \forall f_i$).

Second, we define (**prior**) **scene coverage** S as the percentage of mesh faces used for feature extraction and semantic classification. Traditionally, the entire scene is classified ($S = 100\%$). However, small areas or parts of homogeneous areas may not need to be classified individually, as the graphcut optimization in Section 3 is capable of spreading the correct labelling into these regions from "covered" regions, i.e. regions where the unaries in Eq. 3 are actually evaluated.

Our method aims at reducing both the view redundancy and the scene coverage for an efficient classification, while also improving accuracy. An initial idea could be to use a single global texture by fusing all images, and to only use this texture for extracting and classifying the heavy features. However, as we will show, the visually best texture is not always the best for semantic classification. Hence, we avoid using a fused texture, and rather keep the rich multi-view environment to decide which views are discriminative, yet before classification.

4.1 Ranking Observations by Importance

In this section, we are looking for the most discriminative view per mesh face in terms of semantic classification. Since SfM also delivers the exact camera models $\mathcal{C} = \{c_j\}$, we can accurately relate each 3D surface element f_i (triangular mesh face) to each of the views c_j, as shown in Figure 4. For efficiency, we aim to eliminate observations which are redundant or less important.

For this, we introduce the term **observation importance** $\mathcal{I}$, which deviates from the existing paradigms of pairwise view clustering and ranking. In our work, we require a relationship to the 3D scene, and define $\mathcal{I}_{ij}$ per observation of a mesh face f_i in any camera c_j. Furthermore, our observation importance

ranks according to usefulness for final semantic scene classification rather than for camera clustering or texturing.

Inspired by its success in texture mapping, we will rank the views by the simple texture features such as area and angle. However generally, we define a ranking function that weights the cheap geometric cues for predicting the likelihood of the final classifier performance. The goal is to rank each triangle projection without the heavy feature set. Our importance rank is defined as

$$\mathcal{I}_{ij} = p(f_i \text{ is classified correctly in } c_j | \mathcal{X}_{ij}^{light}). \tag{6}$$

We learn to regress these probabilities, by requiring that $\mathcal{I}_{ij}$ correlates with view and face-wise classification accuracies resulted from the classical scenario, i.e. when all views and all faces are used to extract all features. A view c_j is reliable for classifying face f_i if the semantic label $x_i^* = \text{argmin}_{l \in \mathcal{L}} \Theta(f_i, c_j, l)$ equals the ground truth label. Hence, for the training set, we extract all features and classify all observations of every mesh face. This provides binary labels for reliability (correct/incorrect). We use these and the features $\mathcal{X}^{light}$ (including, e.g. area A_{ij}^{2D}, observation angle α_{ij}) to train a meta-classifier. For this, we use random forests again and, according to Eq. 6, we use the final leaf probability, i.e. classifier confidence, as a measure of the importance $\mathcal{I}_{ij}$. Intuitively, views c_j with small apparent area A_{ij}^{2D} of face f_i, or views observing the face from a sharper angle α_{ij} should be less reliable. For completeness, we also experimented using individual features, such as area A_{ij}^{2D}, angle α_{ij}, class likelihood Θ defined in Eq. 4, or its entropy $H[\Theta]$, to replace the importance $\mathcal{I}_{ij}$.

4.2 Reducing View Redundancy and Scene Coverage

For both characteristics – view redundancy and scene coverage – we use the observation ranking in Eq. 6 to remove redundant views.

For **view redundancy**, we optimize for the best observation $c^*(f_i)$ of each face f_i over all views $c_j \in \mathcal{C}$. This simplifies the energy function in Eq. 3 to

$$E_R(x) = \sum_{f_i \in \mathcal{F}} \Theta(f_i, c^*(f_i), x_i) + \ldots, \quad \text{with} \quad c^*(f_i) = \underset{\forall c_j \in \mathcal{C}}{\text{argmax}}(\mathcal{I}_{ij}), \tag{7}$$

where we select only the maximally informative view per triangle instead of merging unary potentials from all observations. Thus, X^{heavy} only needs to be extracted, described and classified in these most informative views.

For **scene coverage**, we only classify a subset of all triangles that are present in the surface mesh. We choose for each face f_i the most informative view $c^*(f_i)$ having importance I_{i*}. We then rank faces according to their values I_{i*} and only use the set of top k faces $\mathcal{F}^k \subset \mathcal{F}$ for further heavy feature extraction, rather than the full set $\mathcal{F}$. This further simplifies the energy to

$$E_S(x) = \sum_{f_i \in \mathcal{F}^k} \Theta(f_i, c^*(f_i), x_i) + \lambda \cdot \sum_{(f_i, f_j) \in \mathcal{E}} \Psi(f_i, f_j, x_i, x_j) \tag{8}$$

Table 1. Summary of all results (details in supplemental). Semantic Segmentation accuracy (PASCAL IOU in %) for Full428, Sub28 and CamVid102 datasets. By reducing redundancy to zero and also scene coverage to 1/6th, we speedup by 2 orders of magnitude. Ranking by area is better than angle yet the 1st ranks are not best (bold).

	Full428	Sub28	CamVid102	Description
Stats	1794k	185k	46k	# Triangles
	428 (8)	28 (8)	102 (11)	# Images (# Categories)
	9 ± 3	8 ± 2	50 ± 27	Redundancy
Baseline Eq. (3)	35.77	26.05	42.61	MAP SUMALL ($\lambda = 0$)
	35.25	25.13	29.25	MAP MINENTROPY ($\lambda = 0$)
	35.57	25.19	33.21	MAP BESTPROB ($\lambda = 0$)
	37.33	26.63	**50.80**	GC SUMALL (baseline)
	37.82	**26.93**	36.73	GC MINENTROPY $\forall C_j$
	38.27	25.42	37.31	GC MAXPROB $\forall C_j$
SingleView Eq. (7)	37.38 (8px)	26.09 (18px)	52.19 (135px)	Ranked 1st GC AREA (avg)
	37.38 (8px)	26.60 (15px)	54.60 (62px)	Ranked 4th GC AREA (avg)
	35.73 (9°)	25.64 (8°)	47.84 (37°)	Ranked 1st GC ANGLE (avg)
	36.06 (15°)	26.34 (24°)	50.04 (41°)	Ranked 4th GC ANGLE (avg)
	37.04 (0.19)	26.19 (0.49)	52.62 (0.70)	Ranked 1st GC LEARN (avg)
	37.64 (0.18)	**26.86** (0.47)	**56.01** (0.63)	Ranked 4th GC LEARN (avg)
Coverage Eq. (8)	**38.37** (15%)	**28.28** (27%)	**61.07** (35%)	Best Accuracy (AREA)
	37.68 (14%)	26.39 (12%)	57.08 (20%)	1st as Baseline (AREA)
	35.73 (35%)	26.83 (74%)	54.37 (16%)	Best Accuracy (ANGLE)
	35.67 (35%)	25.76 (22%)	52.20 (13%)	1st as Baseline (ANGLE)
	37.08 (35%)	27.97 (40%)	60.57 (31%)	Best Accuracy (LEARN)
	36.15 (33%)	25.96 (34%)	52.98 (13%)	1st as Baseline (LEARN)
Timing	1280min	88min	184min	TIME Full View Redundancy
	11.9x	8.6x	52.6x	SPEEDUP Zero Redundancy
	108min	10.2min	3.5min	TIME Zero Redundancy
	7.1x	8.3x	5.0x	SPEEDUP 1st Coverage as Eq. (3)
	15min	1.2min	0.7min	TIME 1st Coverage as Eq. (3)
	85x	72x	262x	SPEEDUP Overall
	+1.04%	+1.65%	+11.81%	GAIN Overall (absolute)
	103%	106%	124%	GAIN Overall (relative)

which contains unary potentials for only the top k mesh faces, i.e. we set the unaries of all remaining faces to zero. The smoothness term will take care of propagating labels into these areas. An optimal labelling over the complete face set $\mathcal{F}$ defines our final labelling solution (see bottom right of Figure 1).

This is where we again deviate from existing approaches, which evaluate all potentials as they have no means to rank them. Only a recent work [11] introduced the so-called Expected Label Change (ELC) ranking after sampling where to evaluate Θ and running full optimization multiple times. In a multi-view scenario, our methods avoids such a redundant graphcut optimization to estimate the ranking, as we propose the light geometric features to directly estimate the ranking.

5 Experimental Evaluation

In this section we analyze the effect of eliminating view redundancy and reducing scene coverage at the classification stage. As shown below our method considerably reduces computational burden, while showing that we not only maintain but can also improve the final classification accuracy.

We divide our experiments into two investigations summarized in Figure 2 and Table 1. First, we evaluate various importance measures as detailed in Section 4.1 to find the most discriminative view per mesh face. Second, we evaluate the effect of reducing the scene coverage at the classification stage.

Our datasets consist of three outdoor urban scenes annotated with ground truth labels, such as road, wall, window, door, street sign, balcony, door, sky, sidewalk, etc. CamVid [5] is a public dataset. We use its sequence 0016E5, which contains the most buildings and frames. Note that SfM/MVS was only stable for a subset sequence of 102 of its 300 frames. We introduce the larger ETHZ RueMonge 2014 dataset (short: Full428) showing 60 buildings in 428 images covering 700 meters along Rue Monge street in Paris. It has dense and accurate ground-truth labels (Figure 2). Sub28 is a smaller set of 28 images showing four buildings. The CamVid dataset is taken from a car driving forward on a road (with an average viewing angle of 40°) while in the other two datasets the human camera man points more or less towards the buildings (avg. angle $\approx 10°$).

We split each dataset into independent training and testing buildings of roughly 50% of the images and train using all observations of all triangles of the training set. We train both classifiers using a random forest [52, 53] because of its inherent abilities to handle multiple classes, label noise and non-linearity of the features. The number of trees is optimized to 10 and depth to 20 levels.

Please note that our method to reduce view redundancy and scene coverage is general and the speedup generalizes to other semantic classification pipelines. Hence, to study the exact differences, we use the graphcut optimization explained in Eq. (3) over all views as our main baseline (see Table 1).

5.1 Single Discriminative Views – Zero Redundancy

In this first experiment, we determine the most discriminative measure for observation importance. We evaluate the measures in terms of semantic scene classification using PASCAL IOU accuracy averaged over all classes. Table 1 is a summary of our findings. Please look in the supplemental material and website for more detailed results.

As one would expect, exploiting all of the view redundancy and averaging the classifier confidence from each observation (SUMALL) provides stable results. However, these approaches do not provide any speedup and require all the heavy features to be extracted over all observations.

Yet calculating all potentials is the time consuming task, hence we focus on how to find the best observation from cheap geometric features only. The measures to rank are apparent face area A^{2D} (AREA), viewing angle α (ANGLE), and our importance in Eq. 6 (LEARN).

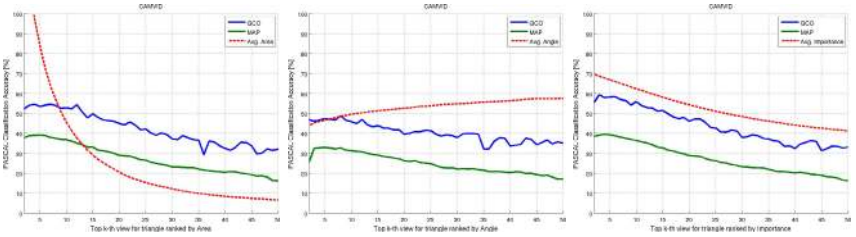


Fig. 5. Removing View Redundancy: showing accuracy for the single k-th ranked feature on x-axis (e.g. 1st largest area, 10th smallest angle, 4th learned importance) and average feature value (red dash). The smaller the area or the larger the angle, the worse performance gets. Our learned performance captures the combination of area and angle better. This is CamVid, other datasets are in supplemental material.

From the evaluations, we have three conclusions. First, on average using the 2D projection area works better than the viewing angle. This is likely due to more robust statistics of larger areas and implicit preference for closer views, as the viewing angle is scale-invariant. Despite the challenging datasets of hugely varying appearance (training to testing performance drops roughly by 30%), other experiments show that the view invariance of the classifier is inherently quite high, which could further explain why the minimum angle is not as useful.

Second and surprisingly, our findings show that neither the largest 2D area nor the most fronto-parallel view deliver the best performance. Rows 10-14 in Table 1 show the average area/angle to change several units for slightly better results. This gain is higher for CamVid because of the steep forward-looking camera and also because of the different semantic classes. For more detail over the class-averaged measures in Figure 5, we also looked at classwise results for area. For all datasets, the classes captured by changing thin 3D surfaces (pole, fence, door, window, sign/pole, etc) experience a gain in accuracy with less frontal projections. These findings suggest that for these classes slanted views better capture the 3D structure.

Overall, our learned combination of the light features works best, since it can balance the distortion of the area and the extreme viewing angles.

5.2 Reduction of Scene Coverage

In the second experiment, we investigate how many total mesh faces are really essential for good performance semantic classification in multi-view scenarios. Going one step further, we reduce the scene coverage and only select the top k triangles after selecting the most discriminative view per triangle.

Here our baselines are a) using all redundancies and the zero redundancy of b) area and c) angle - all at full coverage. The results are shown as average over all classes (top) and as classwise results (bottom) in Figure 6. First conclusion is that the area is usually better at selecting the important triangles for coverage. Its curve climbs faster and overall its accuracy is higher, except for steep-angled

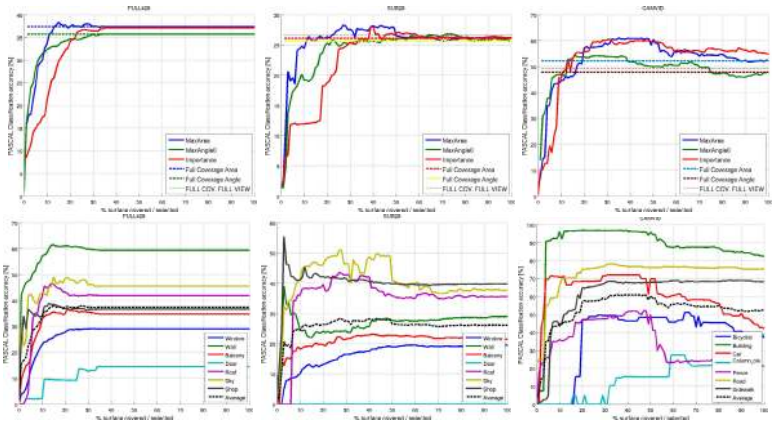


Fig. 6. Reducing Scene Coverage: showing accuracy over percentage of selected triangles within graph optimization. Dashed lines are accuracy at full coverage (allviews, maxarea, minangle, importance). On average 30% are sufficient to label the entire scene as correctly as 100% coverage! Last rows show classwise results (see text for details).

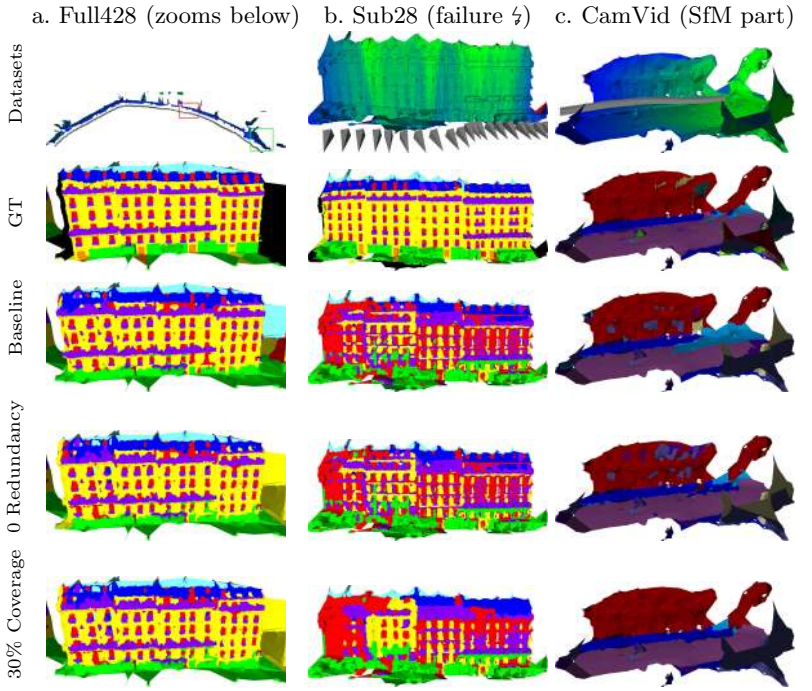


Fig. 7. Overview of results - top left is full street, view redundancy as heatmap (more redundancy, the greener), ground truth (zoomed for two parts of street), and results for full redundancy, single best view and best score for coverage (at stable 30%). Overall, the accuracy are the same after all our speedups. Middle column shows failure cases (ϟ), where the initial classifier already fails and gracefully further smoothes the results.

CamVid dataset. Here the angle measure works better, and overall our learned importance combining the two is best.

Second conclusion may surprise again, we can even get better than the baselines at full coverage (dashed lines)! This is explained by the smaller classes (which occur less frequently and cover less space). Not sampling these early, removes competition for the large classes, which perform much better here. Hence, it is the size of the area that matters. As the importance measure is less good at the early coverage (below 10% coverage), we visualized the three measures and learned that the area is spread across the scene where our learned ranking focuses more on high confidence classes like building and road.

Third and most important conclusion, for large classes it is enough to use 10% of the scene coverage to reach the baselines. Overall, around 30% scene coverage stable results are obtained for all classes. This means that 70% of the potentials usually calculated for semantic scene segmentation are not necessary. The same accuracy can be achieved by using our proposed observation importance and optimization over the graph neighborhood.

6 Conclusions

In this work we investigated methods for reducing the inherent data overlap of multi-view semantic segmentation. As the speeds for other parts have been improved, the bottleneck is the redundant feature extraction and classification.

By exploiting the geometry and introducing single discriminative views per detailed scene part (a triangle), we avoid the redundancy and only classify a single time. This provides a speedup in the order of the data redundancy.

Further, we showed that simple features used for texture mapping are not best when the goal is semantic scene classification. Our learned importance better combines the features like area and viewing angle and improves the ranking.

Lastly, we proposed further efficiency by reducing the scene coverage and classifying only 30% of the scene and still obtain accurate labels for the entire scene. All in all, after reducing the redundancy and coverage we even increase the overall accuracy.

For future work we noticed that the overall accuracy of the scene classification depends on the resolution of this mesh as too large triangles cover semantic units and small triangles are not reliable for classification. Hence we plan to find the best resolution and rank even features in terms of the their computational effort.

Acknowledgements. This work was supported by the European Research Council (ERC) under the project VarCity (#273940) at www.varcity.eu.

References

1. Gammeter, S., Quack, T., Tingdahl, D., van Gool, L.: Size does matter: Improving object recognition and 3D reconstruction with cross-media analysis of image clusters. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part I. LNCS, vol. 6311, pp. 734–747. Springer, Heidelberg (2010)

2. Li, X., Wu, C., Zach, C., Lazebnik, S., Frahm, J.-M.: Modeling and Recognition of Landmark Image Collections Using Iconic Scene Graphs. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 427–440. Springer, Heidelberg (2008)
3. Ladicky, L., Sturgess, P., Russell, C., Sengupta, S., Bastanlar, Y., Clocksin, W., Torr, P.: Joint Optimization for Object Class Segmentation and Dense Stereo Reconstruction. Intern. Journal of Computer Vision (IJCV) 100(2), 122–133 (2012)
4. Sengupta, S., Sturgees, P., Ladicky, L., Torr, P.: Automatic dense visual semantic mapping from street-level imagery. In: Proc. Intern. Conf. on Intelligent Robots Systems, IROS (2012)
5. Brostow, G.J., Shotton, J., Fauqueur, J., Cipolla, R.: Segmentation and recognition using structure from motion point clouds. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 44–57. Springer, Heidelberg (2008)
6. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, CVPR (2012)
7. Ladický, Ľ., Sturgess, P., Alahari, K., Russell, C., Torr, P.H.S.: What, Where and How Many? Combining Object Detectors and CRFs. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part IV. LNCS, vol. 6314, pp. 424–437. Springer, Heidelberg (2010)
8. Tighe, J., Lazebnik, S.: SuperParsing: Scalable Nonparametric Image Parsing with Superpixels. Intern. Journal of Computer Vision (IJCV) 101(2), 329–349 (2012)
9. Koehler, O., Reid, I.: Efficient 3D Scene Labeling Using Fields of Trees. In: Proc. IEEE Intern. Conf. on Computer Vision, ICCV (2013)
10. Sengupta, S., Valentin, J., Warrell, J., Shahrokni, A., Torr, P.: Mesh Based Semantic Modelling for Indoor and Outdoor Scenes. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, CVPR (2013)
11. Roig, G., Boix, X., Ramos, S., de Nijs, R., Van Gool, L.: Active MAP Inference in CRFs for Efficient Semantic Segmentation. In: Proc. IEEE Intern. Conf. on Computer Vision, ICCV (2013)
12. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge (VOC 2012) Results (2012), `http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html`
13. Shotton, J., Winn, J.M., Rother, C., Criminisi, A.: *textonBoost*: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006, Part I. LNCS, vol. 3951, pp. 1–15. Springer, Heidelberg (2006)
14. Tu, Z.: Auto-context and its application to high-level vision tasks. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, CVPR (2008)
15. Kohli, P., Ladicky, L., Torr, P.: Robust higher order potentials for enforcing label consistency. Intern. Journal of Computer Vision (IJCV) 82(3), 302–324 (2009)
16. Ladicky, L., Russell, C., Kohli, P., Torr, P.: Associative Hierarchical CRFs for Object Class Image Segmentation. In: Proc. IEEE Intern. Conf. on Computer Vision, ICCV (2009)
17. Kluckner, S., Mauthner, T., Roth, P., Bischof, H.: Semantic image classification using consistent regions and individual context. In: Proc. British Machine Vision Conference, BMVC (2009)
18. Gould, S., Rodgers, J., Cohen, D., Koller, D., Elidan, G.: Multi-class segmentation with relative location prior. Intern. Journal of Computer Vision (IJCV) 80(3), 300–316 (2008)

19. Munoz, D., Bagnell, J.A., Hebert, M.: Stacked Hierarchical Labeling. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part VI. LNCS, vol. 6316, pp. 57–70. Springer, Heidelberg (2010)
20. Kraehenbuehl, P., Koltun, V.: Efficient Inference in Fully Connected CRFs with Gaussian Edge Potentials. In: Advances in Neural Information Processing Systems, NIPS (2011)
21. Wojek, C., Schiele, B.: A dynamic conditional random field model for joint labeling of object and scene classes. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part IV. LNCS, vol. 5305, pp. 733–747. Springer, Heidelberg (2008)
22. Berg, A., Grabler, F., Malik, J.: Parsing images of architectural scenes. In: Proc. IEEE Intern. Conf. on Computer Vision, ICCV (2007)
23. Xiao, J., Quan, L.: Multiple view semantic segmentation for street view images. In: Proc. IEEE Intern. Conf. on Computer Vision, ICCV (2009)
24. Riemenschneider, H., Krispel, U., Thaller, W., Donoser, M., Havemann, S., Fellner, D., Bischof, H.: Irregular lattices for complex shape grammar facade parsing. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, CVPR (2012)
25. Martinović, A., Mathias, M., Weissenberg, J., Van Gool, L.: A Three-Layered Approach to Facade Parsing. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part VII. LNCS, vol. 7578, pp. 416–429. Springer, Heidelberg (2012)
26. Teboul, O., Simon, L., Koutsourakis, P., Paragios, N.: Segmentation of building facades using procedural shape prior. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, CVPR (2010)
27. Simon, L., Teboul, O., Koutsourakis, P., Van Gool, L., Paragiosn, N.: Parameter-free/pareto-driven procedural 3d reconstruction of buildings from ground-level sequences. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, CVPR (2012)
28. Müller, P., Wonka, P., Haegler, S., Ulmer, A., Van Gool, L.: Procedural modeling of buildings. In: Proc. of the Intern. Conf. on Computer graphics and interactive techniques, SIGGRAPH (2006)
29. Floros, G., Leibe, B.: Joint 2D-3D Temporally Consistent Semantic Segmentation of Street Scenes. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, CVPR (2012)
30. Zhang, C., Wang, L., Yang, R.: Semantic segmentation of urban scenes using dense depth maps. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part IV. LNCS, vol. 6314, pp. 708–721. Springer, Heidelberg (2010)
31. Gallup, D., Frahm, J., Pollefeys, M.: Piecewise planar and non-planar stereo for urban scene reconstruction. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, CVPR (2010)
32. Munoz, D., Bagnell, J.A., Hebert, M.: Co-inference for Multi-modal Scene Analysis. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part VI. LNCS, vol. 7577, pp. 668–681. Springer, Heidelberg (2012)
33. Haene, C., Zach, C., Cohen, A., Angst, R., Pollefeys, M.: Joint 3D Scene Reconstruction and Class Segmentation. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, CVPR (2013)
34. Kim, B., Kohli, P., Savarese, S.: 3D Scene Understanding by Voxel-CRF. In: Proc. IEEE Intern. Conf. on Computer Vision, ICCV (2013)
35. Furukawa, Y., Curless, B., Seitz, S., Szeliski, R.: Towards Internet-scale Multi-view Stereos. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, CVPR (2010)

36. Mauro, M., Riemenschneider, H., Van Gool, L., Leonardi, R.: Overlapping camera clustering through dominant sets for scalable 3D reconstruction. In: Proc. British Machine Vision Conference, BMVC (2013)
37. Debevec, P., Borshukov, G., Yu, Y.: Efficient View-Dependent Image-Based Rendering with Projective Texture-Mapping. In: Eurographics Rendering Workshop (1998)
38. Laveau, S., Faugeras, O.: 3-D scene representation as a collection of images. In: Proc. IEEE Intern. Conf. on Computer Vision, ICCV (1994)
39. Williams, L., Chen, E.: View interpolation for image synthesis. In: Proc. of the Intern. Conf. on Computer graphics and interactive techniques, SIGGRAPH (1993)
40. Silberman, N., Hoiem, D., Kohli, P., Fergus, R.: Indoor Segmentation and Support Inference from RGBD Images. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part V. LNCS, vol. 7576, pp. 746–760. Springer, Heidelberg (2012)
41. Lowe, D.: Distinctive image features from scale-invariant keypoints. Intern. Journal of Computer Vision (IJCV) 60(2), 91–110 (2004)
42. Wu, C.: Towards linear-time incremental structure from motion. In: Proc. of Intern. Symp. on 3D Data, Processing, Visualiz. and Transmission (3DPVT) (2013)
43. Labatut, P., Pons, J., Keriven, R.: Efficient Multi-View Reconstruction of Large-Scale Scenes using Interest Points, Delaunay Triangulation and Graph Cuts. In: Proc. IEEE Intern. Conf. on Computer Vision, ICCV (2007)
44. Hiep, V., Labatut, P., Pons, J., Keriven, R.: High Accuracy and Visibility-Consistent Dense Multi-view Stereo. IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI) 34(5), 889–901 (2012)
45. Jancosek, M., Pajdla, T.: Multi-View Reconstruction Preserving Weakly-Supported Surfaces. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, CVPR (2011)
46. Varma, M., Zisserman, A.: A statistical approach to texture classification from single images. Intern. Journal of Computer Vision (IJCV) 62(1-2), 61–81 (2005)
47. Geusebroek, J., Smeulders, A., van de Weijer, J.: Fast Anisotropic Gauss Filtering. IEEE Trans. on Image Processing (TIP) 12(8), 938–943 (2003)
48. Kluckner, S., Mauthner, T., Roth, P.M., Bischof, H.: Semantic classification in aerial imagery by integrating appearance and height information. In: Zha, H., Taniguchi, R.-i., Maybank, S. (eds.) ACCV 2009, Part II. LNCS, vol. 5995, pp. 477–488. Springer, Heidelberg (2010)
49. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI) 23(11), 1222–1239 (2001)
50. Boykov, Y., Kolmogorov, V.: An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI) 26(9), 124–1137 (2004)
51. Kolmogorov, V., Zabih, R.: What energy functions can be minimized via graph cuts? IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI) 26(2), 147–159 (2004)
52. Amit, Y., August, G., Geman, D.: Shape quantization and recognition with randomized trees. Neural Computation 9, 1545–1588 (1996)
53. Breiman, L.: Random forests. Machine Learning 45(1), 5–32 (2001)