Abstract

Images for 3D mapping are always recorded in such a way that relevant scene parts are seen from multiple viewpoints, so as to facilitate camera orientation and 3D point triangulation. Beyond geometric reconstruction, automatic mapping also requires the semantic interpretation of the image content, and for that task the redundancy provided by overlapping images has been exploited much less. Here we address the task of learning a classifier for pixel-wise semantic labeling of the observed scene. The main insight is that the mere fact that two regions in different images depict the same 3D scene point yields a constraint which can be exploited in the learning phase, namely that they should receive the same class label, even if it is not known which one. In analogy to geometric "tie points"—image correspondences with a priori unknown 3D coordinates, which nevertheless constrain camera orientation—we call these correspondences "semantic tie points". We show how to integrate this weaker form of supervision, which is readily available in any multi-view dataset, into a random forest classifier, and demonstrate improved classification performance of the resulting classifier in an aerial dataset.

1. Introduction

This paper addresses a standard problem of computer vision and remote sensing, pixel-wise semantic image labeling. The task is to assign each pixel in an image to one of a small number of semantically meaningful classes. The standard approach is supervised statistical learning, i.e. a model is learned based on training data with known class labels, and that model is then applied to previously unseen test pixels to estimate (pseudo-)probabilities of belonging to the different classes [14, 16, 15].

In several important cases, such as aerial mapping [13], camera-based driver assistance [7], and 3D image understanding [27], multiple overlapping images of the same scene are recorded, so as to enable 3D reconstruction by structure-from-motion (SfM) methods. The basic assumption of the present work is that the redundancy afforded by multiple views is not only useful for geometric reconstruction, but potentially also for training classifiers.

The basis of SfM computation is the epipolar constraint: corresponding image points, which depict the same physical point in the 3D world, constrain the relative orientation of the cameras, even though their 3D coordinates are unknown and need to be estimated in the process. The constraint is obviously weaker than if the 3D point coordinates are known (the PnP problem), nevertheless it is extremely useful, because image correspondences—in photogrammetric terminology called “tie points”—can be found in large numbers, whereas measuring 3D object coordinates is difficult.

The present paper builds on the insight that the situation is similar for the semantic labeling task: obtaining training data with known labels is costly, but the mere fact that image correspondences show the same point in the physical world means that they should be assigned the same label, and that already is a constraint for classifier training, even though the label itself is unknown. In analogy to the geometric case we call such points “semantic tie points” (STPs). We show how STPs can be used to improve pixel classification in the random forest [4] (RF) framework. To that end the training procedure is altered in such a way that the STP constraints are taken into account. When growing the individual decision trees, the extended scheme not only aims to separate the labeled training examples into different classes, but at the same time also to let corresponding points end up in similar leaf nodes. The rationale is that this biases the training in such a way that the classification is more invariant against variability due to different viewpoints, which should allow it to generalize better (see Figure 1).

The computational cost of including STPs is very low and incurred only during training: the only additional cost are a few simple operations per node when learning the decision trees, whereas STP selection is negligible compared to the training time, and testing is identical to the standard
classifier without STPs. On the downside, one can obviously expect only moderate gains, since all the STPs can contribute is a better regularization of the classifier—the class information still comes entirely from the labeled training data.\footnote{Here the structure-from-motion analogy ends: pixels with known class labels can be thought of as the equivalent to 3D ground control points, but for semantic labeling there is no useful counterpart to a geometric model in local coordinates, without absolute geo-reference.} Moreover, STPs clearly make most sense when few fully labeled training samples are available, such that classifier training is ill-posed (which is however the norm rather than the exception in remote sensing). As the amount of labeled training data increases, the class distributions will be represented more and more accurately, until at some point no more information can be added through STPs.

The proposed combination of fully supervised and weakly supervised learning is evaluated on different test regions taken from an aerial dataset. The experiments show improvements of up to 4.2 percent points in classification performance (Cohen’s $\kappa$ coefficient), compared to a conventional random forest trained without STPs.

2. Related Work

2.1. Random Forests

Random Forests \cite{1, 4} are a state-of-the-art discriminative classification algorithm and have become a popular choice in many computer vision applications, ranging from segmentation \cite{25} through object recognition \cite{3} and detection \cite{10} to tracking \cite{19}. The main reasons for their popularity are (i) they are inherently multi-label classifiers, without having to combine multiple binary classifiers or incur the added complexity of an explicit multi-class margin, (ii) they are fast both to train and to evaluate and can be parallelized well, (iii) they have been shown to be more robust against label noise than other ensemble methods, in particular boosting \cite{4}.

2.2. Learning with weak supervision

Obtaining training data for supervised methods is a tedious and time-consuming task. Therefore there has been recent interest in semi-supervised learning methods \cite{28}. Their goal is to use, during training, a smaller amount of labeled data (which defines the classes) together with a large amount of unlabeled data (which contributes information about the densities in feature space).

Early attempts to utilize only partially annotated data in computer vision include \cite{22, 8}. Recently several authors have attempted systematically to adopt semi-supervised learning methods from the machine learning literature to vision problems or to extend existing supervised computer vision systems to also exploit unlabeled data \cite{18, 24, 9, 6}.

In contrast to semi-supervised learning approaches, our aim is not to improve the decision boundaries based on the feature space density of completely unlabeled samples. Rather, we propose to use STPs as a weak form of supervision, since it is known that they belong to the feature distribution of the same class.

The most similar work we are aware of is \cite{17}, which combines labeled data with unlabeled data extracted from video sequences in a Random Forest framework, in the context of tracking. The space-time consistency of moving objects is exploited by tracking patches over time and using these as weakly labeled samples that must belong to the same class. Corresponding patches in subsequent frames are encouraged to follow the same path in the individual decision trees, while non-corresponding patches are encouraged to follow different paths. Here we use multi-view correspondences rather than correspondences over time to obtain STPs.

2.3. Aerial Image Labeling

We focus on the classification of aerial images. Classification for mapping purposes has a long tradition in satellite remote sensing, e.g. \cite{2, 20, 11} and is still an active research topic. In the context of automatic object modeling, classification has later also been applied to high-resolution aerial images, e.g. \cite{12, 13, 21}. While semi-supervised learning has been investigated, e.g. \cite{5}, we are not aware of any attempts to use multiview correspondences in the way we propose here.

3. Learning with Semantic Tiepoints

In the following we briefly recap the structure and properties of the random forest classifier, mainly to introduce the notation. We then go on to describe the computational basis for using semantic tiepoints, namely how the framework can be extended to use weakly labeled samples for which it is only known that they are images of the same physical point, and thus expected to be similar enough to follow similar paths through the decision trees. Finally, we describe heuristics to select a relevant set of such semantic tiepoints.

3.1. Random Forests

A Random Forest, originally introduced in \cite{1, 4}, is an ensemble of $N$ decision trees $F = \{t_1, t_2, \ldots, t_N\}$. Each decision tree is trained independently to discriminate between $K$ different classes, where the training proceeds in top-down fashion, starting from the root node. At each non-leaf node, the tree holds a binary split function $\tau(f(u), \theta)$ which takes as input the feature vector $f(u)$ of a training sample $I$ and a threshold $\theta$. If $f(u) < \theta$ the sample is propagated to the “left” child node, otherwise to the “right” child node. During training, the thresholds $\theta$ are selected
from a randomly generated set of candidates, such that the information gain on the training data is maximized:

$$\Delta H = -\frac{|u_l|}{|u_l| + |u_r|} H(u_l) - \frac{|u_r|}{|u_l| + |u_r|} H(u_r),$$

(1)

where $u_l$ and $u_r$ denote the training samples propagated to the left, respectively right child nodes, $H(u)$ is a measure of information gain, and $|\cdot|$ denotes the cardinality of a set. The information gain is typically measured either by the entropy $-\sum_k p_k \log(p_k)$ or by the Gini index $\sum_k p_k (1 - p_k)$, with $p_k$ the proportion of samples in $u$ belonging to class $k$. The learning procedure is repeated until either the node is pure (meaning that all samples that reach it are of the same class), or too few training samples are left in the branch, or a given maximum depth is reached.

Each leaf node effectively counts the (relative) frequencies of the $K$ classes among the incoming training samples, which directly serves as estimate of the class-conditional distribution $p(k|v)$ for a test sample $v$ arriving at that particular leaf.

The trick of RFs is simply to construct the individual trees with relatively small, randomly chosen sets of candidate splits at each node and/or with randomly selected subsets of the training data. In this way, the trees are decor-related, such that combining the outputs of all trees acts as a strong regularizer and the ensemble decision generalizes better than that of any individual tree. The overall probability of a test sample belonging to class $k$ is given by:

$$p(k|v) = \frac{1}{N} \sum_{n=1}^N p_n(k|v).$$

(2)

Test samples are then assigned the class with the highest likelihood, i.e. the multi-class decision function of the plain forest is then defined as:

$$C(v) = \arg \max_{1 \ldots K} p(k|v)$$

(3)

Note that the likelihoods estimated by the tree can of course serve as input to more complex probabilistic models (for example as unary potentials in a graphical model).

### 3.2. Extension to Tiepoints

Our aim is to exploit the fact that different image points arise from projections of the same scene point. From that fact it is obvious that any two such points (in the following also called a “pair”) should be assigned the same label. Furthermore, the intuition is that if the viewpoint does not change strongly, the neighborhood in different images should be the same up to simple geometric and radiometric transformations, thus the two samples are expected to belong to the same mode of the distribution in feature space, and to follow similar branches down the decision tree.

To achieve this behavior one can modify the Random Forest by extending the split function, such that it encourages not only low entropy but also similar output for semantic tiepoints (“positive pairs”), respectively different output for “negative pairs” not originating from the same physical point [17]. We use a linear combination of the entropy and the error $Q_{STP}$ committed on the pairs as split function:

$$\Delta H^* = (1 - \lambda) \cdot \Delta H - \lambda \cdot Q_{STP}$$

(4)

where $\lambda \in [0,1]$ controls the influence of the semantic tie points. The new objective function (4) aims to reduce a generalized notion of error defined over both labeled samples and unlabeled pairs, in which the conventional entropy minimization over the labeled samples is regularized with the (soft) tie point constraint (see Figure 1).

The tie point term could be defined in many different ways. In our approach, we count positive pairs being sent
down different branches and negative pairs being sent down the same branch, and add the weighted sum of the (normalized) counts as a penalty to the objective function. Let samples from different images \( \alpha \) and \( \beta \) be denoted \( v^+_\alpha \) and \( v^+_\beta \) if they correspond to the same 3D scene point, respectively \( v^-_\alpha \) and \( v^-_\beta \) if they correspond to different scene points. Then the split function becomes

\[
Q_{STP} = \frac{z}{|v^+|} \sum_{v^+} \left[ \tau(v^+_\alpha, \theta) \neq \tau(v^+_\beta, \theta) \right] + \\
+ \frac{1-z}{|v^-|} \sum_{v^-} \left[ \tau(v^-_\alpha, \theta) = \tau(v^-_\beta, \theta) \right]
\]

(5)

\( z \in [0..1] \) determines the relative weight between positive and negative pairs, and \([expr]\) denotes the Iverson bracket, returning 1 if the enclosed expression is true and 0 otherwise. Intuitively speaking STPs enable the classifier to learn that certain variations in image appearance, namely those observed between corresponding points, are due to (small to moderate) viewpoint changes, and should not be misinterpreted as evidence for different classes. Meanwhile the negative pairs provide information what variations are typically not caused by viewpoint changes. While the negative pairs are not strictly required, they help to avoid an overly strong bias towards uneven splits: thresholds that propagate all data down the same branch trivially avoid breaking positive pairs, but lead the concept of a decision tree ad absurdum.

In the extended scheme, the random split candidates are generated in the same way during training, but evaluated using the new test function. At each node, pairs that are propagated correctly—both down the same branch for positive ones, or each down a different branch for negative ones—remain in the training set for the child nodes. Pairs for which the constraint is violated are not used for training the remaining nodes in the branch.

Testing is identical to the standard Random Forest classifier: once the split functions have been trained, unseen test samples are classified by passing them through the ensemble of trees, just like in a conventional RF.

### 3.3. Finding Tiepoints

With the training framework in place, the remaining task is to select suitable positive and negative pairs. In principle, any pair of corresponding points could serve as tiepoint, and any two points not corresponding to the same scene point could serve as negative example. However, such a procedure will in practice be impaired by several types of errors:

- image orientation and matching are not perfect, thus STPs will be misaligned;
- randomly selected patches can fall into unstructured image regions and be uninformative;
- negative pairs can by chance fall on visually similar regions of the same class, introducing label noise.

We found that to achieve good performance some prefiltering of the pairs is helpful, which will be explained next. For the remainder of this section, we will assume without loss of generality that the images have been densely matched with some stereo method and warped according to the disparity field, such that matching points in both images \( \alpha \) and \( \beta \) have the same image coordinates \((x, y)\). This assumption is particularly well-suited for our application in aerial imaging, where raw images are routinely ortho-rectified by creating a dense model of the terrain surface and applying the image texture to it.

To generate positive pairs, i.e. the actual semantic tiepoints, image points \( x_\alpha \) are sampled at random. To detect uninformative points and residual errors in the registration, we then crop a patch around \( x_\alpha \) from the ”reference” image \( \alpha \), and compute the normalized cross-correlation (NCC) values of that patch over a small region of the second image \( \beta \), centered at \( x_\beta \). If the highest NCC-value is above a threshold \( \psi^+ \), the reference point \( x_\alpha \) and the detected best match \( x_\beta \) form a tie point. Otherwise the pair is discarded and sampling resumes.

To generate negative pairs, two points \( x_\alpha \) and \( x_\beta \) are sampled randomly from the two images at different locations. To reduce label noise the sampled regions are again compared with NCC, and pairs with high cross-correlation above a threshold \( \psi^- \) are rejected.

### 4. Experiments

#### 4.1. Dataset

The GRAZ dataset consists of 4 regions from an aerial mapping campaign in Graz/Austria, each seen in multiple views. The images depict an urban area (see Figure 2). Each image has \(240 \times 240\) pixels and three spectral channels (RGB) and has been resampled to a ground sampling distance of 25 cm. To generate ground truth data, the entire data has been manually labeled into the 4 main land-cover classes \{road, building, grass, tree\}. One of the four areas serves as training set, while the other three are used for testing, i.e. the total amount of test samples is 172'000.

The three test regions were selected to have different characteristics. Region 1 has (as far as possible) similar proportions of the classes, and represents the classical multi-class case. Region 2 is an extreme (but practically rather frequent) case with two dominant classes (mostly buildings and streets, very little vegetation). In fact the result for the tree and grass classes are less relevant here because of the small number of test pixels (\(\approx 2000\)). Region 3 is a relatively “easier” one with mostly well-recognizable building roofs and no cars, and the overall accuracy for that region is noticeably higher. As a side note, region 3 also illustrates
the dilemma of obtaining “ground truth” in practical scenarios. E.g. should the footpath across the lawn be annotated as grass or road?

It should be pointed out that the classification problem is significantly harder than the more common classification of satellite images. On the one hand, it is based only on the intensities in the red, green, and blue channels, whereas multispectral satellite sensors and large format cameras commonly record data also in the near infrared region. These spectral bands make it a lot easier to discriminate vegetation classes. On the other hand, other than one might expect, the higher spatial resolution is not only an advantage: smaller pixels in object space mean less averaging over different materials, surface orientations, and lighting conditions (e.g. individual chimneys, cars etc. become visible). Thereby the spectral variability within each class is greatly increased, and the wider, more diffuse spectral distributions make it harder to assign individual pixels to the right class.

4.2. Features

As features to describe the local color and texture at a pixel, we have used the 17-dimensional filter bank proposed by Winn et al. [26]. The filter bank consists of three Gaussians, four first-order Gaussian derivatives, and four Laplacian-of-Gaussians (LoGs).

The three Gaussian kernels at scales \( \{ \sigma, 2\sigma, 4\sigma \} \) are applied to each \( R, G, B \) color channel\(^2\) independently, thus producing 9 filter responses. The first-order Gaussian derivatives are computed separately in \( x- \) and \( y- \) direction at scales \( \{ 2\sigma, 4\sigma \} \). Derivatives are computed only for the first channel, resulting in 4 additional filter responses. Finally, the four LoGs at scales \( \{ \sigma, 2\sigma, 4\sigma, 8\sigma \} \) were also applied to the first image channel to obtain 4 more filter responses.

Our implementation uses a standard deviation of \( \sigma = 0.7 \) for the highest scale. We empirically found the filter bank to perform as well as other higher-dimensional texture descriptors (e.g. multi-scale Gabor filters).

4.3. Quantitative Evaluation

To assess the influence of STPs we compare classification results with and without using them, based on the same features and classifier. In order not to distort the comparison and really evaluate the effect of STPs we refrain from

\(^2\) We also tried Lab color space as in the original paper, but did not observe significant differences.
the rest of the image (the “background”). The detection task is repeated for the three most frequent classes buildings, roads, and trees.

In the second setting (Subsection 4.4.1), we consider the multi-class problem, where the classifier is directly trained to discriminate all four present classes buildings, roads, trees, grass.

In both settings the performance of the classifiers is evaluated in terms of the $\kappa$-coefficient. Cohen’s $\kappa$ is widely used as a performance metric in remote sensing [23]. The $\kappa$-value (here given in percent) measures how much the predicted labels differ from a random label image with the same class frequencies. With the true positive ratio $P_{tp}$ and the chance agreement $P_\sim$, the metric is defined as:

$$\kappa = \frac{P_{tp} - P_\sim}{1 - P_\sim} = \frac{Y \sum_i c_{ii} - \sum_i (\sum_j c_{ij} \cdot \sum_j c_{ji})}{Y^2 - \sum_i (\sum_j c_{ij} \cdot \sum_j c_{ji})},$$

where the $c_{ij}$ are the entries of the confusion matrix and $Y$ is the number of pixels. By measuring the improvement over a chance agreement (as opposed to the one over a 100% wrong result, which is measured by the overall accuracy) $\kappa$ is more sensitive and better compensates frequency biases.\(^3\) For completeness, overall accuracies and per-class accuracies are also given for the multi-class case in Table 4.4.1.

In all experiments, we have used the following parameters: random forests with $N = 20$ trees of maximum depth 15; and NCC thresholds for tiepoint refinement of $\psi^+ = 0.4$, respectively $\psi^- = 0.2$. All results have been averaged over 50 runs to suppress the fluctuations induced by the randomized selection of training samples as well as the randomization during random forest training.

4.3.1 Binary labeling

To simulate the frequent application problem of detecting the pixels belonging to a specific class, we train binary classifiers for buildings, streets and trees, respectively. Each classifier is trained from 100 labeled training pixels randomly picked from the training region (see Figure 2).

These classifiers are then applied to each of the three test regions in turn, reaching average $\kappa$-values of 37.1% for buildings, 17.1% for roads, and 53.8% for trees (respectively overall classification accuracies of 69.0%, 74.6% and 89.1%). The procedure is repeated with STPs. 8000 positive as well as 8000 negative pairs are extracted randomly as described in section 3.3, always from the test region the classifier will be applied to. Overall we thus get 9 different binary classifiers (3 classes $\times$ 3 STP sets).

Table 1 summarizes the results. For all binary classifications and all test regions the standard Random Forest (RF) is compared to the proposed Random Forest trained with STPs. It can be seen that on average there is an improvement in classification accuracy for the building and road classes, while there is only a negligible gain for the trees (in some test regions the performance for trees slightly decreases).

In Figure 3, the differences between the two classifiers are depicted visually for three example areas. For each area we show, as overlays on the test image, the ground truth, the RF classification, the STP classification, and the difference between the RF and STP predictions. Green denotes pixels where the proposed method improves the result (i.e. STP predicts correctly, whereas the RF does not), red denotes pixels where the result is degraded. It can be seen that the gain by using STPs occurs mostly in radiometrically ambiguous areas.

4.4. Number of labeled samples

As explained, STPs are expected to be most useful if little labeled training data is available—once there are enough labeled samples to represent the class-conditionals, no further gain can be expected by adding STPs. We thus ran one more experiment in which we increase the number of

\(^3\)E.g. consider a binary problem with 10% foreground and 90% background pixels. A classifier which always returns background will have 90% overall accuracy, but $\kappa$=0%.
 labeled training examples until the gain through STPs vanishes. The results are depicted in Figure 4. As expected the STPs are most useful when very few (100) training samples are used, and their influence diminishes until at (four our data and classes) 1000 labeled samples they no longer improve the overall result, and in fact slightly weaken the classifier for some classes.

4.4.1 Multiclass labeling

In this setting, the random forest is trained to discriminate four different labels. Since this is a more difficult problem, 1000 training samples are used, again randomly picked from the training area. For each test region the STPs are again extracted from that same region, to stay close to the application problem where one typically has not got unused “auxiliary data” which need not be classified.

The results are summarized in Table 2. Using STPs yields consistent gains in classification accuracy. The results vary across different classes and test regions: while there is a noticeable improvement of 4.2% in region 1, the gains are smaller for the other two test regions.

Table 2. Classification accuracies (κ in %) in different test regions for the multiclass case.

<table>
<thead>
<tr>
<th>region</th>
<th>test region 1</th>
<th>test region 2</th>
<th>test region 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>RF</td>
<td>30.7</td>
<td>33.9</td>
<td>48.9</td>
</tr>
<tr>
<td>STP</td>
<td>34.9</td>
<td>34.5</td>
<td>49.3</td>
</tr>
<tr>
<td>gain</td>
<td>+4.2</td>
<td>+0.6</td>
<td>+0.4</td>
</tr>
</tbody>
</table>

Table 3. Per-class accuracies and overall accuracies for the multiclass case.

<table>
<thead>
<tr>
<th>region</th>
<th>test region 1</th>
<th>test region 2</th>
<th>test region 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>RF</td>
<td>70.8</td>
<td>21.2</td>
<td>60.2</td>
</tr>
<tr>
<td>STP</td>
<td>70.1</td>
<td>21.6</td>
<td>62.7</td>
</tr>
<tr>
<td>gain</td>
<td>-0.7</td>
<td>+0.4</td>
<td>+2.5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>region</th>
<th>test region 2</th>
<th>test region 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>RF</td>
<td>82.7</td>
<td>46.8</td>
</tr>
<tr>
<td>STP</td>
<td>81.5</td>
<td>48.5</td>
</tr>
<tr>
<td>gain</td>
<td>-1.2</td>
<td>+1.7</td>
</tr>
</tbody>
</table>

Figure 4. Gains through using STPs as the number of fully labeled training samples increases. As expected STPs only contribute relevant information as long as the labeled samples are insufficient to model the class-conditional distributions.

5. Conclusion

We have explored the possibility to use two-view correspondences with unknown labels (“semantic tie-points”) as a form of weak supervision for pixel-wise image labeling. In the proposed approach, the fact that the classifier should be invariant to the differences of appearance induced by a small change in viewpoint is translated into the condition that the two corresponding feature vectors should follow similar paths through a decision tree. That condition can be readily included in the objective function for training decision trees, respectively Random Forests, with only little additional computational effort during training and no additional cost at test time. Experimental results on an aerial dataset show that the method improves classification of aerial images by up to 4.2 percent points. Still, the gains are not always consistent across classes, and it remains an interesting open question whether there are more effective ways to exploit the semantic tie-point constraint.

References


