University of Freiburg

Seminar Summaries

Igor Bogoslavskyi

February 13, 2013

1 What Makes Paris Look Like Paris

1.1 Which is the main problem in generating good discriminative patches?

In the given paper the visual features should represent a given geographical locale (e.g. the city of Paris). That means that these patterns should be both frequently occurring within given locale and geographically discriminative, i.e. they appear in this locale and don't appear elsewhere. Neither of these two requirements is enough on their own.
In the current work, the visual elements are represented by patches at various resolution that are mined from a huge image-database.
The overwhelming majority of this data is uninteresting, so matching the occurrences of the rare interesting elements is extremely difficult.
There are a few possible approaches to do that. Below, the main ideas of them are represented as well as why some of them don't work for current problem.

- Bag Of Words. Unfortunately, standard visual words tend to be dominated by low-level features, like edges and corners. Higher-dimentional feature descriptors are hard to cluster on the other hand.

- An alternative approach is to use the geographic information as part of the clustering, extracting elements that are both repeated and discriminative at the same time. However, this includes clustering of the whole space at least once and the rare discriminative elements get mixed with, and overwhelmed by, less interesting patches, making it unlikely that a distinctive element could ever emerge as its own cluster.

1.2 Which are the shortcoming of using standard distance metrics for clustering? How do the authors overcome this problem?

The main problem is that a standard distance metric, such as normalized correlation, does not capture what the important parts are within an image patch, and instead treats all pixels equally. For example, the the street sign candidate (Figure 3 center), has a dark vertical bar along the right edge, and so all the retrieved NN matches also have that bar, even though it's irrelevant to the street sign concept.
Recently, it was shown how one can improve visual retrieval by adapting the distance metric to the given query using discriminative learning. We adopt similar machinery, training a linear SVM detector for each visual element in an iterative manner while also adding a weak geographical constraint. The procedure produces a weight vector which corresponds to a new, per-element similarity measure that aims to be discriminative.

2 Moving Object Segmentation Using Motor Signals

2.1 How is the consistency between tracked feature points and motor signals checked?

For image I_t at time t, two types of features are detected: corners and edges. The locations of the corners and sampled edge points form sparse feature set P_t. These sparse features are tracked in I_t's neighboring frames I_{t+k} ($k = \{M, ..., 1, 1, ..., M\}$). The tracked features in frame I_{t+k} are denoted as P_{t+k}. Given the motor signals at two frames t and $t+k$ and a mapping function f, the homography or fundamental matrix between the two frames is calculated. Function f serves as a mapping function from motor signal change to visual change and should be computed beforehand, once for each new robot setup.
From frame I_t to I_{t+k}, the background features should be consistent with the computed transformation H_k or F_k, while the foreground features will violate this transformation. Therefore the features can be classified based on the errors between the actual tracked feature locations and their estimated locations predicted from H_k or F_k

2.2 How is it used for sparse foreground-background segmentation?

After the classification one can cluster the tracked features based on the error set. As it is hard to set some threshold to divide these features in two classes, the Expectation Minimization algorithm is used to fit a two-component Gaussian Mixture model (corresponding to background/foreground).

3 RGB-(D) Scene Labeling: Features and Algorithms

3.1 What is an RGB-D image?

RGB-D image is a data structure where for each pixel we have not only the RGB value, but also the distance to the corresponding point from the sensor.

3.2 What are the contributions of this work that lead to significantly improved labeling accuracy?

The main contribution of this work is adapting the framework of kernel descriptors that converts local similarities (kernels) to patch descriptors for RGBD images, that are able to capture a variety of RGB-D features such as gradient, color, and surface normals.

4 Highly Scalable Appearance-Only SLAM

4.1 What is the main contribution of the paper and what are the main differences with respect to previous works on the same problem?

A key contribution of this paper is that the authors describe FAB-MAP 2.0, the appearance-only SLAM system - a modified version of the probabilistic model over bag-of-words which extends its applicability by more than two orders of magnitude in scale. The dependencies between co-occurrences of the visual words are captured via learning a tree-structured Bayesian Network using the Chow Liu algorithm, which yields the optimal approximation to the joint distribution over the work occurrence. The tree-structured network also allows efficient learning and inference even for large vocabulary sizes. One of the modifications made to previous approach was to the probabilities in the location models, so that the belief of non-existence of the feature does not change as more supporting observations become available. This change allows sparse likelihood update.

4.2 What is a Chow Liu tree and why it is needed for FAB-MAP?

In probability theory and statistics Chow-Liu tree is an efficient method for constructing a second-order product approximation of a joint probability distribution.
As the visual words do not occur independently, these dependencies are important to capture. This is done by learning a tree-structured Bayesian network using the Chow Liu algorithm, which yields the optimal approximation to the joint distribution over word occurrence within the space of tree-structured.

5 Kintinuous: Spatially Extended KinectFusion

5.1 What are the main ideas and techniques behind KinectFusion?

At the core of the KinectFusion algorithm is a truncated signed distance function (TSDF), a volumetric representation of the scene, where each location stores the distance to the closest surface. A weight that is proportional to the uncertainty of the surface measurement is also stored for each value in the TSDF.
To integrate the raw data from each new frame into the TSDF, KinectFusion first computes current point cloud and normals. These are then used to compute the pose of the camera using ICP in conjunction with a predicted surface model derived from the current TSDF. Extraction of a predicted surface from the TSDF is achieved by detecting the zero crossings through a GPU based ray casting operation. Given the output of the ICP procedure, new measurements are integrated by first transforming them into the frame of reference of the global TSDF. The TSDF is then updated via a weighted running average using the weights mentioned above.

5.2 What is Kintinuous and how does it relate to KinectFusion?

Kintinuous is an extension of the KinectFusion Algorithm. In the KinectFusion algorithm the camera was not allowed to move outside of the TSDF, but in the new algorithm the TSDF is re-centered on the camera once it moves too far from the origin.This allows the volume of the model to increase.