SemanticPaint

Adam Kosiorek *

Abstract. The short abstract (50-80 words) is intended to give the reader an overview of the work.

1 Introduction

Capturing your own environment has never been easier. SemanticPaint can register your surroundings which, after undergoing a low-level 3D reconstruction, can be semantically segmented in an interactive way. Not only it works in real time but also requires no pretraining. Adding new object categories on the fly is facilitated by online model updates. The user is provided with instantaneous feedback and can re-label any object to correct errors. SemanticPaint makes capturing customized environment models with object classes particular to the user's interest easy and efficient.

The pipeline starts with capturing the environment as a stream of noisy RGBD images and combining them into a 3D model updated in an online fashion. The user can choose which objects to label and can do so by "touching" a small part of an object or encircling one with his hand and uttering the label. It is recognized by a standard speech recognition system. The label and the information about the affected data points are further passed to a Streaming Random Forest classifier which constantly learns and labels all visible voxels. To further improve classification results a spatially dense labeling is produced by an efficient mean-field inference algorithm. One of the biggest strength of SemanticPaint is the efficiency of each part of the pipeline, which translates to real time performance. Algorithms used in the pipeline were adapted to work on volumetric data in the TSDF format directly in order to avoiding the costly conversions to mesh or point-cloud formats. To allow this, the Voxel Oriented Patch features — a new type of a discriminative feature describing the voxel space — has been designed. The contributions can be summarized as follows: 3D semantic modeling system, Streaming Random Forest, efficient Mean-Field inference, Voxel Oriented Patch features.

Numerous applications are possible: (1) building large scale datasets of 3D objects or whole scenes for use in large-scale computer vision systems (2) using the dense semantic labeling of 3D environments in robot navigation or to aid people with impaired sight and (3) map environments for use in augmented reality scenarios or games.

The rest of the paper is organized as follows: Section 2. describes the related work, section 3 details internal data handling, section 4 describes the efficient

* Advisor: M.Eng. Keisuke Tateno, Chair for Computer Aided Medical Procedures & Augmented Reality, TUM, WS 2015/16.

mean-field inference algorithm, section 5 outline the Streaming Random Forest classifiers.....

2 Related Work

Acquisition and Reconstruction. Capturing the geometry of the surrounding world has been a long standing problem [1]. An offline processing of multiple images allowed to reconstruct digital heritage and construct world-scale 3D models with remarkable quality [6]. The inception of low-cost RGBD sensors and powerful GPUs enabled online 3D scanning, augmented reality or using 3D environment models for navigation purposes [7]. [3] enables 3D reconstruction of small scenes using a single off-the-shelf RGB camera. It uses a sparse tracking method to first estimate the camera's pose and then select key frames and relative to them secondary frames, from which 3D stereo reconstruction is performed. The achieved results are similar to KinectFusion with the only limitation being that the precision of texture-less surfaces' reconstruction is somewhat lacking.

Scene Understanding. Object recognition, detection and segmentation has been done with 2D RGB images, RGBD data, point clouds and volumetric representations. [2] uses a Voxel-based CRF for simultaneous reconstruction and segmentation. Each voxel contains information about visibility and occlusion as well as group membership. The first two are used to to improve reconstruction by mitigating depth-map noise. The visibility values are constrained by that each ray from the camera can hit only one visible vortex. The group membership information encodes priors given by bounding boxes of detected objects. A graph-cut algorithm is used for global inference. The whole scene has to be registered beforehand, since any change in the CRF's structure would require restarting the inference algorithm. No computational performance was reported. [4] is a first step towards online simultaneous registration and segmentation. Using RGBD images, the framework constructs a model of the environment and updates it each time a new frame comes in. When a significant change in the model is detected, the resulting model is split into a static and a dynamic part, where the latter is assumed to have moved in space. The movement is detected by comparing the expected and observed intensity values at each voxel. The system is online, but is far from real time with 0.7 to 2s processing time per frame. [5] shows that converting to another data representation can help to improve segmentation accuracy and inference speed. TSDF model is used to reconstruct a 3D scene from a sequence of RGBD images. The volumetric representation is triangulated and a 3D mesh is recovered. Next, visual features are computed directly on images and projected into the 3D model, while geometric features are computed on the mesh directly. Segmentation is done via a CRF. The approach is tested on the augmented KITTI and NYU datasets for indoor and outdoor scene segmentation where it delivers state-of-the-art results.

3 Space Representation

The space is represented by a 3D volume constructed from the sequence of RGBD images using Truncated Signed Distance Function (TSDF) [8]. Large-scale scene handling is facilitated by the hashed volumetric representation of [9] with a 6mm^3 voxel. Each voxel contains: TSDF distance, colour information in the LAB space, a weight for averaging distance values, a user annotation class, a predictied class, an outcome of the mean-field inference from before and after classification and the probability of each class as inferred by the mean-field algorithm, for each class. The total number of bits per voxel is $96 + [\text{number of classes}] \cdot 16$.

4 Voxel Oriented Patches

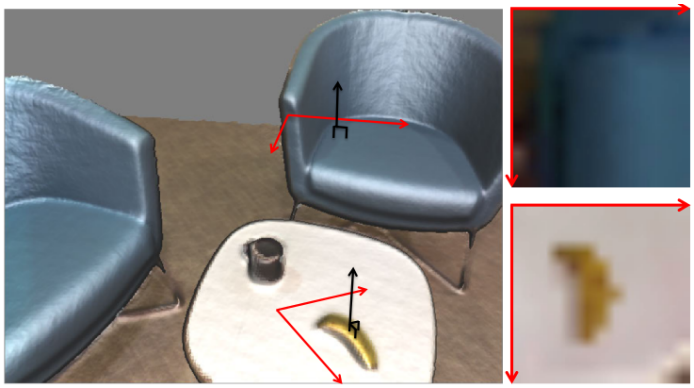

Fig. 1. Voxel Oriented Patch. Colours shown in RGB for illustration purposes.

Even the most powerful classifier will be useless if inappropriate features are supplied. Since speed is crucial, this paper contributes Voxel Oriented Patch (VOP) features, which can be computed efficiently from the TSDF volume representation (cf. figure 1. Let V_i be a VOP and $\mathbf{n}^i$ the normal of voxel i. We create an image patch of dimensions $r \times r$ centered on voxel i from the plane $(\mathbf{p} - \mathbf{p}_i) \cdot (n)_i = 0$. The patch containes colour values stored in the TSDF on the plane in CIELab to mitigate illumination effects. Additionaly, V_i stores distance to the nearest dominant horizontal surface. $r = 13$ with a resolution of $10 \frac{mm}{pixel}$ is used. Rotation invariance is achieved exactly as in [12]. The patch is rotated according to the dominant gradient direction. It is computed by creating a histogram of gradient directions with 36 bins, where the directions are weighted by gradient magnitudes and a gaussian window centered at the voxel. Furthermore, a quadratic polynomial is fit to the three biggest values of the histogram, whose maximum is the final directiion.

5 Streaming Random Forests

A random forest is an ensamble of classification or regression trees, whose outputs are combined to produce a result with lower variance [10]. Each tree is typically trained only on a subset of the training data S comprised of pairs (i, l) where i is a sample and l its label. Let $f(i;\theta) \in \{L, R\}$ denote the binary split function with learned parameters θ, which specify a feature this node uses. The tree's output amounts to the probability distribution $P(c_i = l|i)$ stored at each node. Trees learn in a greedy way: for each node a split function is chosen from a distribution Θ of candidate split functions to maximize the information gain.

$$G(S, S^L, S^R) = H(S) - \sum_{d\in\{L,R\}} \frac{|S^d|}{|S|} H(S^d) \tag{1}$$

Where $H(S) = -\sum_{(l,i)\in S} p(c_i = l) \log p(c_i = l)$ is Shannon Entropy. Training set is then split into subsets $S^d(\theta)$, $d \in \{L, R\}$ used to train child nodes. This procedure requires that all training data is available.

Online Random Forest [10] allow runtime model updates and they differ in that for a new node n they generate a distribution of split function Θ_n and maintain a set of label statistics for each of them, updated when a new sample comes to the node. Label statistics are sufficient to compute information gain and thus the node n is further split if it has seen a predefined number of samples or if the minimal value of the information gain is low enough. Storing and udpdating statistics in leaf nodes is computationally expensive which limits runtime performanc

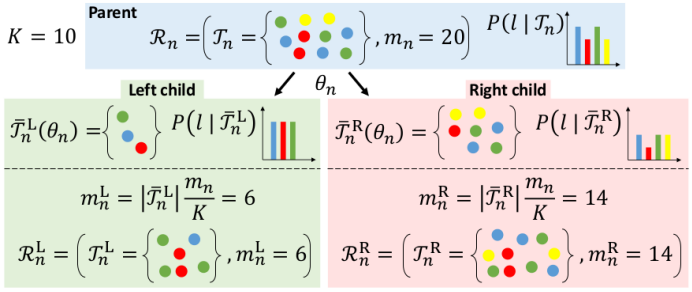

Fig. 2. Splitting reservoirs in Streaming Random Forest.

Streaming Random Forest begins by storing a reservoir R_n of at most K samples stored in a list T_n at the root node. It represents an unbiased sample of all training data seen so far. Initially, the first K samples are stored. Then, a sample in the reservoir is exchanged with an incoming one with probability $p = \frac{K}{m_n}$, where m_n is the number of samples seen so far at node n. Splitting the node requires computing the objective function for each $\theta \in \Theta_n$ over R_n. $H(R_n)$

is computed from the normalized class label histogram $P(l|R_n)$. Let $|R_n|$ be the amount of samples stored in the reservoir n and let R_n^d be the splitting induced by θ. Finally, the split function is chosen as $f_n = \arg\max_{\theta \in \Theta_n} G(S, S^L, S^R|\theta)$ and set the number of samples seen by the child node to $|m_n^d| = |R_n^d| \max\left(1, \frac{m_n}{K}\right)$. Since the total number of samples stored is bounded by K this approach uses less memory. Constructing child reservoirs from parent reservoirs (cf. fig 2) lessen the computational load. It also outperform Online Random Forests in classification accuracy.

6 Dynamic Conditional Random Field

Conditional Random Fields are often used for segmentation [**?**], but they usually assume availability of all data. Here, a pairwise CRF [13] with a time dependent underlying model is used. It requires a novel inference alogorithm that can handle updates of the geometry and user specified labes. The class of each voxel is modeled by a random variable x_i. The label distribution over the volume $\mathcal{V}$ factorizes into likelihood terms $\psi_i(x_i)$ and prior terms $\psi_{ij}(x_i, x_j)$. Let $\mathcal{E}_i$ be a neighbourhood of v_i, $\mathbf{D}_t$ volumetric data at time t and $\mathbf{x}$ a vector of all x_i. The posterior distribution is given by

$$P(\mathbf{x}|\mathbf{D}) = \prod_{i \in \mathcal{V}} \left(\psi_i(x_i) \prod_{j \in \mathcal{E}_i} \psi_{ij}(x_i, x_j) \right) \tag{2}$$

Neighbourhood of 6 cm is used. Negative log likelihood of eq. 2 defines the energy

$$E_t(\mathbf{x}) = \sum_{i \in \mathcal{V}} \left(\phi_i(x_i) + \sum_{j \in \mathcal{E}_i} \phi_{ij}(x_i, x_j) \right) + K \tag{3}$$

Unary and pairwise potentials $\phi_i(x_i)$ and $\phi_{ij}(x_i, x_j)$ encode the cost of assigning a label to v_i and a cost of assuming different labels by voxels in a neighbourhood, respectively. These costs are constatly changing due to the chainging predictions of the random forest and user interaction.

Initially, all voxels are encouraged to take a background label by setting $\phi_i(l) = 0$ for background class and grater than zero otherwise. When the user touches an object, a set of touched voxels $\mathcal{H}_S$ is registered and their unary potentials set to $\phi_i(l) = \infty$ if their labels are different then the specified one, and zero if they are the same, thus imposing the labeling. If user labels a region more than once, the old labelings are simply overwritten.

To process the encircling action, the labeling is projected into the current frame. Next, a GMM is fitted with the foreground class taken as a convex hull of the encirvled region and background as the rest of the image. For every pixel in a bounding box around the user annotation the unary potential are updated as

$$\phi_i(l) = \begin{cases} \log P_E(fg|\mathbf{a}_i) & \text{if } l = fg \\ \log(1 - P_E(fg|\mathbf{a}_i)) & \text{if } l = bg \end{cases} \tag{4}$$

with $P_E(fg|\mathbf{a}_i)$ the probability of assuming the foreground label, which is a normalized likelihood $P_(\mathbf{a}_i|fg)$ from the GMM.

$$P_E(fg|\mathbf{a}_i) = \frac{P_(\mathbf{a}_i|fg)}{P_(\mathbf{a}_i|fg) + P_(\mathbf{a}_i)|bg} \tag{5}$$

For all voxels that haven't been explicitly labaled by the user unitary potentials are updated with the random forest predictions as $\phi_i(l) = -\log P_F(x_i = l|\mathbf{D})$. Finally, smoothness is achieved by the standard Potts model by assigning a disconuity cost $\phi_{ij}(x_i, x_j) = \lambda_{ij}$ if voxels have different labels and zero otherwise, with λ_{ij} a function of difference in position, intensity and normal directions.

7 Efficient Mean-Field Inference

The mean-field inference algorithms follows [14]. It is adapted to the GPU programming model and to the constantly changing energy landscape. First, the original prbability distribution $P(\mathbf{x})$ is approximated by $Q(\mathbf{x})$ under KL-divergence $KL(Q||P)$. It is chosen such that the marginal of each random variable is independent, that is $Q(\mathbf{x}) = \prod_i Q_i(\mathbf{x})_\mathbf{i}$). It is iteratively refined with updates derived from the fixed point of KL-divergence [15] as

$$Q_i^t(l) = \frac{1}{Z_i} e^{M_i(l)},\ t = 1, \ldots, T \tag{6}$$

$$M_i(l) = \phi_i(l) + \sum_{l' \in \mathcal{L}} \sum_{j \in \mathcal{N}(i)} Q_j^{t-1}(l') \phi_{ij}(l, l') \tag{7}$$

$$Z_i = \sum_{i \in \mathcal{L}} e^{-M_i(l)} \tag{8}$$

$$x_i^\star = \arg\max_{l \in \mathcal{L}} Q_i^T(l) \tag{9}$$

It is guaranteed to converge under some conditions [14]. However, these guarantees do not hold here due to changing energy landscape. Moreover, the energy distribution is assumed to change only gradualy instead of initializing enerrgies at a given frame from a uniform distribution, it is initialized with the energies of the previous frame. It works well in practice, speeds up the process and allows visually pleasing label propagation effect.

SRF classification results would impact the final segmentation after several frames due to their effect on unitary potential. To speed up the process, however, the initial energy value at a new frame is taken as a weighted sum of the value from the previous frame and the SRF prediction, that is

$$\widetilde{Q}_i^t(x_i) = \gamma Q_i^{t-1}(x_i) + (1 - \gamma) P_F^{t-1}(x_i = l|\mathbf{D}),\ \gamma \in [0, 1] \tag{10}$$

8 Putting it All Together

9 Results

10 Future Work

11 Conclusions

References

1. ROBERTS , L. G. 1963. Machine perception of three-dimensional solids. Ph.D. thesis, Massachusetts Institute of Technology.
2. KIM , B.-S., KOHLI , P., AND SAVARESE , S. 2013. 3D scene understanding by voxel-CRF. In Proc. ICCV.
3. PRADEEP , V., RHEMANN , C., IZADI , S., ZACH , C., BLEYER , M., AND BATHICHE , S. 2013. Monofusion: Real-time 3D reconstruction of small scenes with a single web camera. In Proc. ISMAR.
4. HERBST , E., HENRY , P., AND FOX , D. 2014. Toward online 3-d object segmentation and mapping. In IEEE International Conference on Robotics and Automation (ICRA).
5. VALENTIN , J. P., SENGUPTA , S., WARRELL , J., SHAHROKNI , A., AND TORR , P. H. 2013. Mesh based semantic modelling for indoor and outdoor scenes. In Proc. CVPR.
6. LEVOY , M., PULLI , K., CURLESS , B., RUSINKIEWICZ , S., KOLLER , D., EREIRA , L., GINZTON , M., ANDERSON , S., DAVIS , J., G NSBERG , J., ET AL . 2000. The digital Michelangelo project: 3D scanning of large statues. In Proc. SIGGRAPH. ACM.
7. NEWCOMBE , R. A., IZADI , S., HILLIGES , O., MOLYNEAUX , D., KIM , D., DAVISON , A. J., KOHLI , P., SHOTTON , J., HODGES , S., AND FITZGIBBON , A. 2011. KinectFusion: Real-time dense surface mapping and tracking. In Proc. ISMAR.
8. CURLESS , B. AND LEVOY , M. 1996. A volumetric method for building complex models from range images. In Proceedings of the 23rd annual conference on Computer graphics and interactive techniques. ACM, 303312.
9. NIESSNER , M., Z OLLH OFER , M., IZADI , S., AND STAMMINGER , M. 2013. Real-time 3D reconstruction at scale using voxel hashing. ACM TOG 32, 6
10. SAFFARI , A., LEISTNER , C., SANTNER , J., G ODEC , M., AND BISCHOF , H. 2009. On-line random forests. In IEEE ICCV Workshop.
11. VITTER , J. S. 1985. Random sampling with a reservoir. ACM TOMS 11, 1.
12. LOWE , D. G. 1999. Object recognition from local scale-invariant features. In Proc. ICCV.
13. LAFFERTY , J., M C C ALLUM , A., AND PEREIRA , F. C. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data.
14. [Kr ahenb uhl and Koltun 2011]
15. [Koller and Friedman 2009]