2021 43rd Annual International Conference of the
IEEE Engineering in Medicine & Biology Society (EMBC)
Oct 31 - Nov 4, 2021. Virtual Conference

2D to 3D Segmentation: Inclusion of Prior Information using Random Walk Kalman Filters

Peter Somers[1], Johannes Schüle[1], Cristina Tarín[1], and Oliver Sawodny[1]

Abstract— **Augmented reality is a quickly advancing field that has the potential to provide surgeons with computer generated diagnostic results during surgery. Visual classification of diseased tissue generated during a diagnostic procedure, for example, trans-urethral cystoscopy of the urinary bladder, can aid a surgeon during the following resection to ensure no tissue is inadvertently missed. Work with 2D segmentation of camera images is well developed and frameworks already exist to fuse this data real-time in a 3D reconstruction. These existing frameworks, however, maintain only the most recent segmentation information when building the 3D reconstruction. This work proposes a method to build a 3D point cloud classification using random walk Kalman filters. The method enables retention of prior classification information and additionally provides a framework to include additional sensor classifications contributing to a single, final 3D segmentation result. The method is demonstrated using a simulated environment intended to emulate the inside of a human bladder.**

I. Introduction

During endoscopic or similar visually obstructed, camera-driven surgeries, a surgeon maintains only a limited view during the work. This leads to difficulty when dividing a surgery such as a trans-urethral resection into separate diagnostic and resection operations. Locations of tumors or suspicious tissue discovered during the investigative phase must be memorized by the surgeon or rediscovered during resection. This could lead to missed tumors and the unusually high recurrence rate associated with bladder cancer and is supported by a 2010 study that found a 40% recurrence rate due to missed tumors during trans-urethral resections [1].

Virtual or augmented reality systems provide a framework to build a synthetic map embedded with additional information that could help to solve this problem. Many works have been published that are capable of reconstructing a 3-dimensional map of an environment in real-time [2], [3] and offline [4] using image data from a camera. Recent works have begun to combine 2D semantic classification of objects with the construction of these environments [5]. Application of these frameworks to endoscopic operations is a very promising surgical advancement, but still requires enhancements before it can gain the trust required for use in such a sensitive and critical environment.

During reconstruction and segmentation of the 3D maps previously mentioned, the segmentation is simply applied as a broadcasting of the resulting 2D segmentation mask of the current video frame to its corresponding 3D component (either a point or a triangular surface). The problem is this method leads to lost information as it is easy for a bad camera image angle or a poorly trained segmentation neural network (NN) to lead to a poor outline of the current object in question. This results in mislabeling of portions of the 3D map. The current frameworks maintain either the first classification obtained or the most recent, but neither method avoids this mislabeling problem.

This work introduces a method to maintain information from multiple 2D segmentations made during the construction of a 3D point cloud, ideally leading to convergence of the true segmented map. This capability is crucial when dealing with endoscopic data, since it very common that non-ideal or partial images can lead to improperly classified tissue. The solution proposed uses a random walk model for the classification and individual Kalman filters to update the 3D map as new segmentation images are obtained. This method not only allows for combining segmentation information of multiple images, but also provides a framework to include additional segmentation information that comes from a different, possibly sparser, modality. After collection and creation of the segmented 3D point cloud, smoothing algorithms, such as [6], may be used to improve the results further. The method is implemented in an online fashion as would be required for use during an investigative operation that uses an augmented reality assistive display.

The following in Section II will provide an overview of the random walk model, Kalman filter, and how it is formed to this problem. In Section III, the simulation environment built and used to test the algorithm is introduced. Section IV portrays some initial testing results showing the effectiveness of the data fusing capabilities of the method and in Section V concluding remarks are made along with comments regarding the future direction of the work.

II. Kalman Points

The foundation of this contribution relies on the sensor fusing capabilities of the Kalman filter to provide a base for combining successive 2D classifications of a 3D environment. The Kalman filter relies on the fusing of a model prediction and a current measurement of a given system. It is primarily used in dynamic system applications for dynamical models that evolve through time where a stream of incoming information from sensors such as an accelerometer are joined with an approximated system model. The states of this model are updated every time step using the model

*This work was sponsored by the Graduate School 2543/1 “Intraoperative Multisensory Tissue Differentiation in Oncology” (project ID 40947457) funded by the German Research Foundation (DFG - Deutsche Forschungsgemeinschaft).

[1] Institute for System Dynamics, University of Stuttgart, 70563 Stuttgart, Germany {somers, schuele, tarin, sawodny}@isys.uni-stuttgart.de

978-1-7281-1178-0/21/$31.00 ©2021 IEEE

and measurements, which leads to convergence of the true state values. For more information on the Kalman filter and its prediction process, the reader is referred to [7].

In this work, the classification value c for every 3D reconstructed point of an online generated point cloud is stored as a state within a Kalman filter for the given point. If considering m different classes that the point could possibly belong to, a full state vector $\mathbf{x} = [c_1 \; c_2 \; \dots \; c_m]^\top$ is obtained. Since a model transition function is not known or, more likely, does not exist, i.e. the classification is not expected to change between measurements, the transition matrix is assumed to be identity. This results in random walk, discrete time, model dynamics at time step k

$$\mathbf{x}_{\mathrm{k}+1} = \mathbf{I}\mathbf{x}_{\mathrm{k}} + \mathbf{w} \tag{1}$$

driven entirely by the process noise $\mathbf{w} \sim \mathcal{N}(0, \mathbf{Q})$ with zero mean and covariance $\mathbf{Q}$.

Measurements $\mathbf{z}$ of the system are obtained using existing classification techniques that provide a prediction confidence that the point belongs to each internal state. These measurements can be described by

$$\mathbf{z} = \mathbf{H}\mathbf{x} + \mathbf{v} \tag{2}$$

where the measurement matrix $\mathbf{H} = \mathbf{I}$ since the states are measured directly and with measurement noise $\mathbf{v} \sim \mathcal{N}(0, \mathbf{R})$. This measurement noise in the context of a classifier is essentially an additional measurement of how much the classification can be trusted. For example, if after training, a neural network performs consistently worse for a particular class c_i, then the covariance value associated with state x_i should be increased, reflecting a higher uncertainty. Both, the measurement covariance $\mathbf{R}$ and process covariance $\mathbf{Q}$ are both assumed to be diagonal matrices, implying the individual states, or classifications, are independent from one another.

Upon creation of each point, the initial state is directly assigned the value of the corresponding segmentation value. Subsequent classification measurements are applied with the standard Kalman filter update algorithm. By applying classifications in this way, a newly obtained erroneous classification will not completely overwrite a previously applied classification. This also enables the system to accept measurements from a classifier that results in either a *soft* classification

$$c \in [0, 1] \tag{3}$$

or a *hard* classification

$$c \in \{0, 1\}. \tag{4}$$

III. Segmented Point Cloud Generation

A simulated endoscopic environment is constructed using openGL [8] in which synthetic RGBD (color and depth) images are generated. These images would be generated in practice by using an RGBD camera directly or with odometery data from a simultaneous localization and mapping algorithm such as [9] and then using reconstruction techniques such as those proposed in [5].

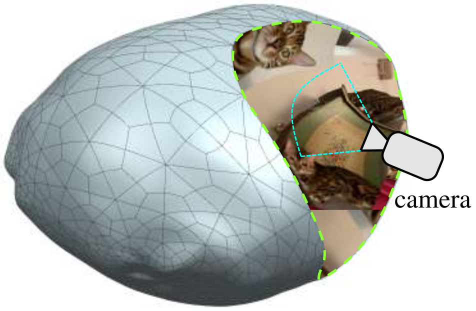


Fig. 1. Testing framework with a simulated 2D surface mesh of human bladder with cutaway showing interior image texture.

For trans-urethral surgeries a NN trained to recognize bladder tumors such as that developed in [10] would be used. However, as a publicly provided pre-trained NN is not currently available, an off-the-shelf general object pre-trained NN [11] is used. This network is lightly modified to provide soft classifications for each pixel and images of cats are projected to the representative bladder surface to represent tumorous regions. This is sufficient to prove the feasibility of the algorithm and also demonstrates the ability of the proposed method to improve the overall classification resulting from a conglomerate of lower quality classifications. Lower quality in this sense referring to poor segmentation results due to the augmented shape and angle of images projected to the surface.

The 3D surface mesh of the testing framework can be seen in Figure 1. The 3D geometry comes from an artistic rendering of a human urinary bladder and the texture is applied using a hand labeled segmented image. This allows for maintaining a reference to what the true classification of each point should be when evaluating the results.

The point cloud is generated in an online fashion using key frames and the open source framework open3d [12]. Every key frame is broken into a color and depth image and a new set of points is generated. In order to associate overlapping images with existing points, an approximated nearest neighbors (ANN) search is implemented using Annoy [13] for each of the new points. This prevents the overall point cloud from containing redundant information and provides the foundation for deciding what data is supplied as measurement info to already constructed Kalman points. An example of this is shown in Figure 2 where the overlapping new points are reassigned to existing ones before adding non-matched new points to the overall point cloud. The full process is outlined in Algorithm 1.

In the current implementation, the ANN algorithm needs to be rebuilt every key frame to include new points. For small regions, such as those experienced in a bladder cystoscopy, this is not a large problem. In an ideal case, the organ of interest can be reconstructed in advance with an imaging technique such as magnetic resonance imaging (MRI). From this, a fixed set of points can be constructed eliminating the task of dynamically adjusting the ANN search as new areas are segmented. It is unlikely, however, that this approach would be very reliable for such a geometrically dynamic

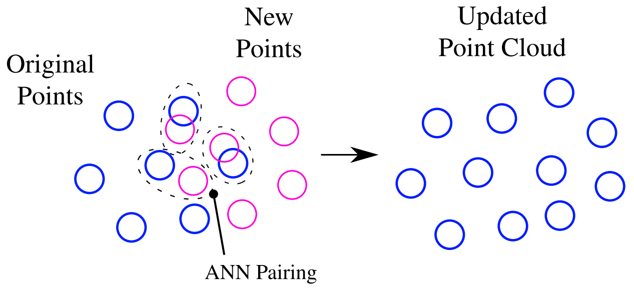


Fig. 2. Example reassignment of overlapping points. Reassigned point values are used to update the Kalman filters of the associated points.

Algorithm 1: Segmented point cloud generation.

```
Result: Point cloud containing classification data
// initialize states
pt_cloud // empty point cloud
while new_img // new key frame image
 do
    rgb, d = generate_rgbd_image(new_img)
    new_points = open3d_points_from_RGBD(rgb, d)
    segm = NN(rgb) // segment image with
        NN
    foreach $pt \in new\_points$ do
        nearest_pt = ANN(pt) // nearest
            existing point
        if $norm(pt, nearest\_pt) < \alpha$ then
            kalman_pt = nearest_pt
        else
            kalman_pt = new_kalman_pt
            pt_cloud += kalman_pt
        end
        kalman_pt.update(segm[pt]) // update
            Kalman filter with
            corresponding segmentation
            value
    end
    ANN.initialize(pt_cloud) // rebuild ANN
        with updated point cloud
end
```

organ as the bladder. For this reason, the described online generation method is used.

IV. Results

For testing, the process noise covariance $\mathbf{Q} = [0.5]$ is used for each generated point to reduce the likelihood that the classification of each given point changes between measurements. The measurement covariance is simply $\mathbf{R} = [1]$. Note that only one state is considered. This binary case represents either cancerous tissue or not and comes from the fact that only a single classifying tool is used. Sample key frame images from the test model are evaluated that can be seen in Figure 3. The top three images represent what is seen by the monocular camera and the corresponding point cloud classification is generated and shown below. Corresponding areas between the images and the point cloud are marked to help the reader identify with orientation. A red point value corresponds to a high confidence that the point belongs to the given class, while blue is simply background material.

TABLE I
Jaccard distance (Intersection over Union) for the image sequence of figures 3(a)-3(c).

	Image 3(a)	Image 3(b)	Image 3(c)
Initial Value	0.84573	0.77935	0.77495
Latest Value	0.84573	0.75930	0.78788
KF Updated Value	0.84573	0.77006	0.80836

It is possible to see that in the third key frame image the neural network was not capable of detecting the portion of the suspected region in the upper left of the image inside the "C" marker. However, the Kalman filter update prevented the region from completely being reclassified as the background. Therefore, the 3D point cloud view successfully maintains the helpful information from prior classifications. With the newly updated view, a surgeon would be alerted to the need to double check this region and perhaps focus on taking an additional or different sensor measurement at the region in order to confirm the suspected tissue.

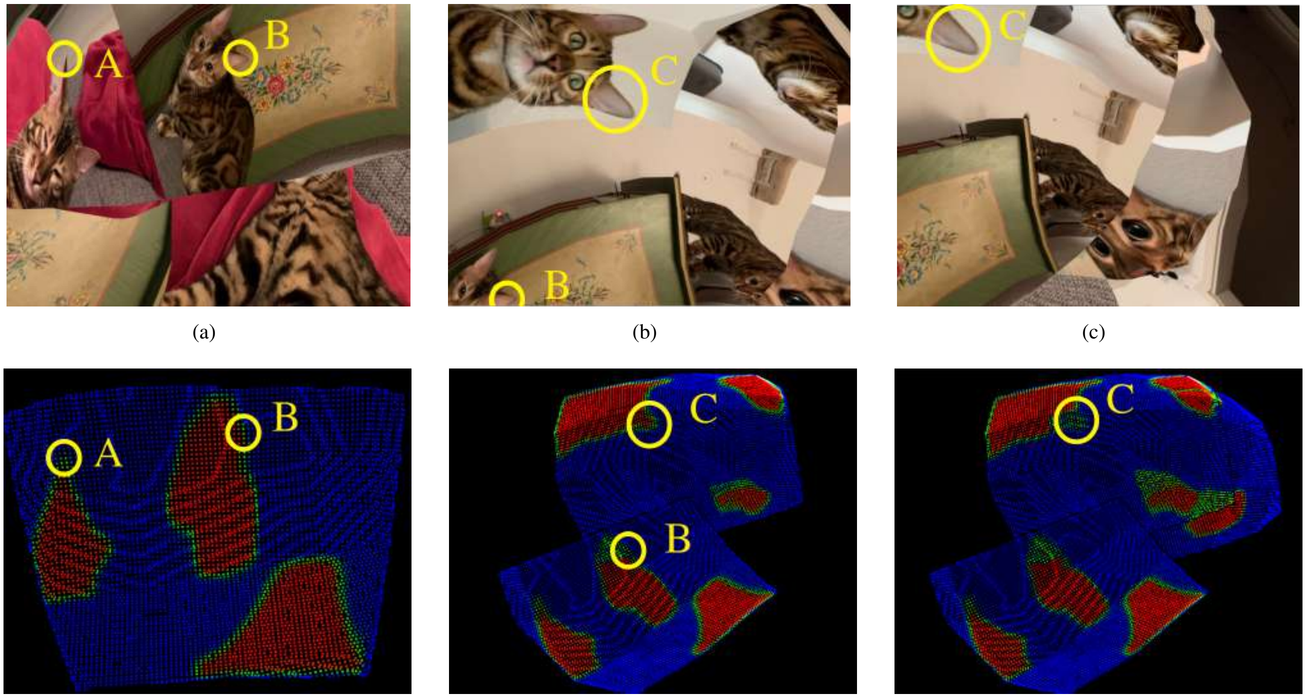
The results of the point cloud segmentation shown in Figure 3 are listed in Table I, and simultaneously compared with results that assume only either the segmentation value of the latest or initial measurement for a given point is retained. The Jaccard distance, or Intersection over Union value, is used as a comparison metric for all cases and is given as

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} \tag{5}$$

where A and B are the point classifications of the NN and the true values, respectively. A cut-off threshold of 0.25 for the NN classification is applied in order to obtain a binary value for evaluation of (5). This cutoff value was found not to be very influential for the results as the NN used resulted in a near binary output already. This is seen by the apparent green outlines in Figure 3 that actually occur simply due the color blending between a confidence of 0 (blue) and 1 (red).

From Table I it can be seen that as more images are taken and new areas are simultaneously explored, the retention of all information through the Kalman filter, slowly drives the classified point cloud to a more accurate overall segmentation than the simple direct assignment methods. This result is to be expected and shows that the provided method can provide a tool for construction of more accurately segmented point clouds during exploration of a localized region such as that of the inside of a human bladder.

Naturally, the final segmentation is also dependent on the accuracy of the 2D segmentation method used. An example of an artificial inaccuracy simply due to the 2D segmentation can be seen in the upper left of the classification images in Figure 3. Despite a complete overlap of the detected region, the resulting classification is always taken as the background along the border of the images. This is likely due to padding and original training conditions of the NN. Effects such as



(a) (b) (c)

(d) (e) (f)

Fig. 3. Example point cloud generation: (a)(b)(c) key frame image sequence taken from camera; (d)(e)(f) classified point cloud generation. Point coloring from blue to red corresponds to a classification confidence (0-1) between background or suspicious regions, respectively. Yellow markers highlight select corresponding areas between each image.

these should be accounted for, however, when using the provided method the mentioned effect is reduced as seen in the upper right corner of Figure 3(f).

V. Conclusion and Future Work

A method of retaining 2D segmentation information from prior images in a video sequence, either live or pre-recorded, through the embedding of information as states in a Kalman filter in individual points within a 3D point cloud has been proposed. Each point within the cloud utilizes a random walk dynamic model with a Kalman filter update algorithm to fuse classification data from the different images. A simulated test environment was used to demonstrate example cases to show how the proposed method is able to push a final segmented point cloud to a more accurate result. The method can be combined with any existing image segmentation method that provides a soft or hard pixel classification value in order to fuse the segmentation of an object using images from multiple angles. For evaluation purposes in this work, the soft classifications were reduced to hard classifications using thresholding.

The next major challenge in moving forward with this framework is to reduce the computation time through the use of pre-initialized surface data. In a surgical environment, this can be obtained from pre-operative MRI images and a registering step would need to be initially performed to match the intraoperative navigation of the camera to the corresponding initialized surface. The segmentation values may then be stored as texture information and quicker, already well-established algorithms can be used for processing and evaluating the data.

References

[1] J. A. Witjes, J. P. Redorta *et al.*, "Hexaminolevulinate-guided fluorescence cystoscopy in the diagnosis and follow-up of patients with non–muscle-invasive bladder cancer: Review of the evidence and recommendations," *European Urology*, vol. 57, no. 4, pp. 607 – 614, 2010.

[2] V. Pradeep, C. Rhemann *et al.*, "Monofusion Real-time 3d reconstruction of small scenes with a single web camera," in *2013 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, 2013, pp. 83–88.

[3] L. Chen, W. Tang *et al.*, "Slam-based dense surface reconstruction in monocular minimally invasive surgery and its application to augmented reality." *Computer methods and programs in biomedicine*, vol. 158, pp. 135–146, May 2018.

[4] J. L. Schönberger and J.-M. Frahm, "Structure-from-motion revisited," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[5] A. Rosinol, M. Abate *et al.*, "Kimera: an open-source library for real-time metric-semantic localization and mapping," *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1689–1696, 2020.

[6] L. Landrieu, H. Raguet *et al.*, "A structured regularization framework for spatially smoothing semantic labelings of 3d point clouds," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 132, pp. 102–118, 2017.

[7] S. Haykin, *Adaptive filter theory*, 4th ed. Upper Saddle River, NJ: Prentice Hall, 2002.

[8] "Pyopengl," https://pypi.org/project/PyOpenGL, Mar. 2020.

[9] J. Lamarca, S. Parashar *et al.*, "Defslam: tracking and mapping of deforming scenes from monocular sequences," 2019.

[10] E. Shkolyar, X. Jia *et al.*, "Augmented bladder tumor detection using deep learning," *European Urology*, vol. 76, no. 6, pp. 714 – 718, 2019.

[11] W. Abdulla, "Mask r-cnn for object detection and instance segmentation on keras and tensorflow," https://github.com/matterport/Mask_RCNN, 2017.

[12] Q.-Y. Zhou, J. Park, and V. Koltun, "Open3D: A modern library for 3D data processing," *arXiv:1801.09847*, 2018.

[13] E. Bernhardsson, "Annoy: Approximate nearest neighbors in c++/python," https://github.com/spotify/annoy, accessed: 2021-03-01.