An Online Visual Loop Closure Detection Method for Indoor Robotic Navigation

Can Erhan[a], Evangelos Sariyanidi[b], Onur Sencan[c], Hakan Temeltas[c]

[a]Istanbul Technical University Mechatronics Engineering Department, Maslak 34469, Istanbul, Turkey; [b]Queen Mary University of London School of Electronic Engineering and Computer Science, ADDRESS, London, England; [c]Istanbul Technical University Control and Automation Engineering Department, Maslak 34469, Istanbul, Turkey

ABSTRACT

Visual loop closure detection problem is an active area especially for the indoor environments, where the global position information is missing and the localization information is highly dependent on odometry sensors. In this paper, we propose an enhanced loop closure method based on unsupervised visual landmark extraction with saliency detection technique. In contradistinction to the previous methods, our approach uses additional depth information. Saliency regions are used to refer to the certain distinctive areas on the image patches. They are also suitable to represent locations in a sparse manner. In order to find out the similarity between two locations we use a straightforward function, that contains the detection confidences of the image and the landmark coordinates in 3D. Recognition of the previously visited and unvisited locations is also considered in the framework. Exemplary results and the practical implementation of the method are also given with the data gathered on the testbed with a depth camera (Kinect) mounted differential drive autonomous ground vehicle. Specifically, we adopt visual place recognition to close loops that is useful for the process of correctly identifying a previously visited location.

Keywords: Loop closure, depth map, zernike moments, computer vision

1. INTRODUCTION

In mobile robotics, autonomous navigation is an active research area especially for the indoor environments, where the global position information is missing and the localization information is highly dependent on odometry sensors. One of the major problems connected to robotic navigation is Simultaneous localization and mapping (SLAM) which still remains as an assertive problem in a lot of ways. Loop closing is defined as the correct identification of previously visited location in terms of SLAM. Information that is gathered from various data sources including LIDARs, RADARs, stereo and monocular cameras is highly utilized in loop closure detection. With the recent development of visual sensor technologies, the loop closure detection became a problem that is open for research in the field of computer vision.

In this paper, we present an enhanced loop closure method based on image-to-image matching with the additional depth information in indoor environments. Specifically, we adopt visual place recognition without using any metrical information such as rotation and exact location to correctly identify the previously visited location. This technique is quite simple because of its low computational complexity, and performs in real-time with high loop closing accuracy.

The technique we implement relies on discrete Complex Zernike Moments (CZMs) in 2-dimensional space. They are extracted using a set of complex polynomials which form a complete orthogonal radial basis functions defined on the unit disc. These moments are used to represent shape information inside the image within the

Further author information:
Can Erhan: E-mail: erhanc@itu.edu.tr
Evangelos Sariyanidi: E-mail: e.sariyanidi@eecs.qmul.ac.uk
Onur Sencan: E-mail: osencan@itu.edu.tr
Hakan Temeltas: E-mail: hakan.temeltas@itu.edu.tr

context of image processing. To make the image description more robust, the ZMs of the input image is calculated in local manner. In this study, the ZMs are extracted from both the greyscale and depth images separately and concatenated sequentially to create ultimate feature vector.

The locations are represented with the histograms of their images. Detecting loop closing events can be considered as a machine learning problem which number of classes expands continuously when a new unseen location comes. We utilized Nearest Neighbour (NN) algorithm to classify locations by which the distance metric is the regular L_1 aka. Manhattan distance that is one of the simplest distance metric available. In other words, the image which distance between the input image is minimum closes the loop, if the distance is lower than the predefined threshold value.

1.1 Literature Review

Visual loop closing techniques can be categorized into three main parts[1] : Image-to-image and image-to-map, map-to-map techniques. Most of the image-to-image loop closing approaches that rely on considering whole image[2,3] .

Loop closure estimations are cast by matching images. Most of image-to-image loop closing approaches that rely on salient image regions, utilize small, low-level features like[?] extracted from the whole image.[?,?,?,?] However, there is a technique that is used to extract relatively larger patches specific to environment that the robot navigates[?] on. Similar approaches are defined in other studies as well,[?,?] however these techniques do not define a complete loop closing framework.

The Bag of Words[?] method has also been widely used to visual place recognition. Many notable techniques have been developed after this representation, since it is quite suitable to represent place images and match them on an appearance space. The well-known FAB-MAP method proposes a generative probabilistic model[?] which relies on BoW model, and a method to identify new places is proposed. The BoW model has also been utilized along with Bayesian Filtering[?] to reliably detect the loop closures. The BoW model has also used to eliminate the majority of candidates[?] and loop closing is decided after geometrically inconsistent matches are discarded using conditional random fields. More extensive information related visual loop closure techniques can be found on the relevant survey.[?]

2. METHODOLOGY

As it was mentioned earlier, the presented loop closing approach relies on calculating the Zernike Moments of partitioned image blocks across the input image in a local manner.

The whole image representation is constructed as follows. Firstly, the ZMs of each subimage that were obtained by partitioning the input image is calculated. After that, the calculated complex moments are coarsely quantized by keeping only the sign of the real and imaginary components and ignoring the rest of the information. Finally, the quantized binary data is coded as histograms that are concatenated to form the final feature vector.

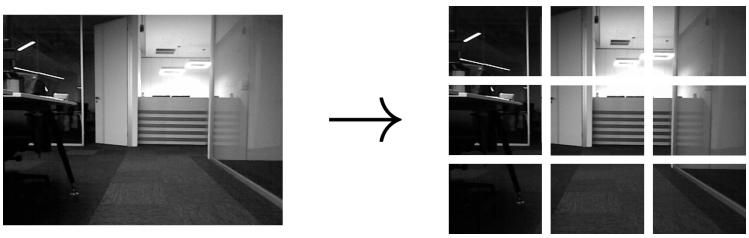


Figure 1: Image partitioned into blocks by k = 3

2.1 Extraction of Quantized Local Zernike Moments

The Complex Zernike Moments (ZM) of an image are used to represent the image on the 2-dimensional complex subspace. They are extracted using a set of complex polynomials which form a complete orthogonal radial basis functions defined on the unit disc. The coefficients in this complex subspace describe the holistic shape information within the image.

The Complex Zernike Moments of an image $f(i,j)$ are calculated as follows:

$$Z_{nm} = \frac{n+1}{\pi} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} f(i,j) V^{\star}(\rho_{ij}, \theta_{ij}) \Delta x_i \Delta y_j, \tag{1}$$

where x_i, y_j stand for the image coordinates, $\rho_{ij} = \sqrt{x_i^2 + y_i^2}$ and $\theta_{ij} = \tan^{-1} \frac{y_j}{x_i}$ are the polar coordinates of the image, n defines the order of the ZM and m stands for the number of repetitions. The constraint between n and m is stated that $|m| \leq n$, and $n - |m|$ must be even. Therefore, the number of components with respect to moment order n is calculated as follows:

$$K(n) = \begin{cases} \frac{n(n+2)}{4} & \text{if } n \text{ is even} \\ \frac{(n+1)^2}{4} & \text{if } n \text{ is odd.} \end{cases} \tag{2}$$

In order to extract ZMs in local manner, the input image is divided by $k \times k$-sized subimages in which k is a predefined partitioning parameter. If the image is not divided by k perfectly, the rest of the part is simply ignored. The set of images obtained from partitioning are flatten into a $k^2 \times 1$-sized single column in row-major order for more straightforward representation.

Consider that the input image $\mathbf{I}_{p \times q}$ is a combination of subimages I_{ij}:

$$\mathbf{I}_{p \times q} = \begin{bmatrix} I_{11} & \dots & I_{1Q} \\ \vdots & \ddots & \vdots \\ I_{P1} & \dots & I_{PQ} \end{bmatrix} \longrightarrow \begin{bmatrix} I_{11} \\ \vdots \\ I_{1Q} \\ \vdots \\ I_{P1} \\ \vdots \\ I_{PQ} \end{bmatrix} \tag{3}$$

where $P = \lfloor {}^{p}/_{k} \rfloor$ and $Q = \lfloor {}^{q}/_{k} \rfloor$. An exemplar input image partitioned into subimages is shown in Figure 1. The Z_{nm} of the input image can be considered as the transformation of the subimages in (3) individually. This transform is applied to subimages with all n and m values according to its order. Lastly, the transformed subimages contains a set of complex coefficients are collected into a new matrix such as:

$$\mathbf{Z}(\mathbf{I}_{p \times q}) = \begin{bmatrix} \mathbf{Z}(I_{11}) \\ \vdots \\ \mathbf{Z}(I_{1Q}) \\ \vdots \\ \mathbf{Z}(I_{P1}) \\ \vdots \\ \mathbf{Z}(I_{PQ}) \end{bmatrix} = \begin{bmatrix} Z_{00}(I_{11}) & \dots & Z_{nm}(I_{11}) \\ \vdots & & \vdots \\ Z_{00}(I_{1Q}) & \dots & Z_{nm}(I_{1Q}) \\ \vdots & & \vdots \\ Z_{00}(I_{P1}) & \dots & Z_{nm}(I_{P1}) \\ \vdots & & \vdots \\ Z_{00}(I_{PQ}) & \dots & Z_{nm}(I_{PQ}) \end{bmatrix} = \begin{bmatrix} a_{00}^{11} + \mathbf{i}b_{00}^{11} & \dots & a_{nm}^{11} + \mathbf{i}b_{nm}^{11} \\ \vdots & & \vdots \\ a_{00}^{1Q} + \mathbf{i}b_{00}^{1Q} & \dots & a_{nm}^{1Q} + \mathbf{i}b_{nm}^{1Q} \\ \vdots & & \vdots \\ a_{00}^{P1} + \mathbf{i}b_{00}^{P1} & \dots & a_{nm}^{P1} + \mathbf{i}b_{nm}^{P1} \\ \vdots & & \vdots \\ a_{00}^{PQ} + \mathbf{i}b_{00}^{PQ} & \dots & a_{nm}^{PQ} + \mathbf{i}b_{nm}^{PQ} \end{bmatrix}. \tag{4}$$

Therefore, $k^2 \times K(n)$-sized complex coefficient matrix is obtained. In order to facilitate the representation of the data and to reduce the size of the descriptor vector, a coarse quantization takes place at this step. First,

the complex numbers are decoupled as real and imaginary components, then the sign of each matrix element is taken to binarize the decoupled $k^2 \times 2K(n)$-sized matrix as follows:

$$\mathbf{Z}(\mathbf{I}_{p\times q}) \longrightarrow \begin{bmatrix} a_{00}^{11} & b_{00}^{11} & \dots & a_{nm}^{11} & b_{nm}^{11} \\ \vdots & \vdots & & \vdots & \vdots \\ a_{00}^{1Q} & b_{00}^{1Q} & \dots & a_{nm}^{1Q} & b_{nm}^{1Q} \\ \vdots & \vdots & & \vdots & \vdots \\ a_{00}^{P1} & b_{00}^{P1} & \dots & a_{nm}^{P1} & b_{nm}^{P1} \\ \vdots & \vdots & & \vdots & \vdots \\ a_{00}^{PQ} & b_{00}^{PQ} & \dots & a_{nm}^{PQ} & b_{nm}^{PQ} \end{bmatrix} \longrightarrow \begin{bmatrix} sgn(a_{00}^{11}) & sgn(b_{00}^{11}) & \dots & sgn(a_{nm}^{11}) & sgn(b_{nm}^{11}) \\ \vdots & \vdots & & \vdots & \vdots \\ sgn(a_{00}^{1Q}) & sgn(b_{00}^{1Q}) & \dots & sgn(a_{nm}^{1Q}) & sgn(b_{nm}^{1Q}) \\ \vdots & \vdots & & \vdots & \vdots \\ sgn(a_{00}^{P1}) & sgn(b_{00}^{P1}) & \dots & sgn(a_{nm}^{P1}) & sgn(b_{nm}^{P1}) \\ \vdots & \vdots & & \vdots & \vdots \\ sgn(a_{00}^{PQ}) & sgn(b_{00}^{PQ}) & \dots & sgn(a_{nm}^{PQ}) & sgn(b_{nm}^{PQ}) \end{bmatrix} \tag{5}$$

where $sgn(\bullet)$ is the signum function. According to (1), Z_{nm} with $m = 0$ is omitted since their imaginary components turn out to be constantly equal to zero.

The next step is representing the binary matrix shown in (5) with histograms. Using the binary values for the overall image representation is not a practical solution. Describing binary data with histograms are more common solution to obtain the image representation. As the methodology described so far, $k^2 \times 2K(n)$-sized binary matrix is extracted. Each row of this matrix is linearly combined by weighting each of them as a power of 2. Hence the final feature vector $\mathbf{X}$ is obtained as follows:

$$\mathbf{X} = \begin{bmatrix} sgn(a_{00}^{11}) & sgn(b_{00}^{11}) & \dots & sgn(a_{nm}^{11}) & sgn(b_{nm}^{11}) \\ \vdots & \vdots & & \vdots & \vdots \\ sgn(a_{00}^{1Q}) & sgn(b_{00}^{1Q}) & \dots & sgn(a_{nm}^{1Q}) & sgn(b_{nm}^{1Q}) \\ \vdots & \vdots & & \vdots & \vdots \\ sgn(a_{00}^{P1}) & sgn(b_{00}^{P1}) & \dots & sgn(a_{nm}^{P1}) & sgn(b_{nm}^{P1}) \\ \vdots & \vdots & & \vdots & \vdots \\ sgn(a_{00}^{PQ}) & sgn(b_{00}^{PQ}) & \dots & sgn(a_{nm}^{PQ}) & sgn(b_{nm}^{PQ}) \end{bmatrix} \begin{bmatrix} 2^0 \\ 2^1 \\ \vdots \\ \vdots \\ \vdots \\ 2^{2K(n)-2} \\ 2^{2K(n)-1} \end{bmatrix} = \begin{bmatrix} x_{11} \\ \vdots \\ x_{1Q} \\ \vdots \\ x_{P1} \\ \vdots \\ x_{PQ} \end{bmatrix} \tag{6}$$

where x_i is an integer ranging between 0 and $2^{2K(n)-1} - 1$.

Finally, the input image $I_{p\times q}$ is described with a histogram of length of $2^{2K(n)-1}$.

3. EXPERIMENTS

The method implemented in this paper has been evaluated on three datasets which lighting conditions are different from each other.

tested with k = 5, n = 2 ZM parameters k x k is the number of regions the input image is divided n: the order of ZM tested in three different datasets by using image only, depth only and both image and depth. image and depth is created by concatenated with image and depth ZM feature vectors

3.1 Benchmark Datasets

In order to measure loop closing performance of the algorithm in different illumination levels, three different datasets are created in indoor environment by using Kinect sensor which can grab both RGB and depth images at the same time. The lighting conditions in the datasets are sequentially bright, dim and dark in which both depth and RGB images are captured. The exemplary images belong to different datasets are shown in Figure 2. In some cases, the line of sight of the Kinect is not enough to capture the depth information for $10m$ above distances, thus some of the depth images are decomposed.

Images are collected via Kinect mounted moving platform and approximately pointed in the direction of the displacement with average speed $0.25m/s$. Also, the rectangle-shaped trajectory of the three datasets is nearly

the same as the others to compare their performances. There are two loops that contains 1000 frames per loop for each dataset, one for discovering, the other for evaluating.

manually annotated, each frame is considered as a different location, similar frame sensitivity is approx 10 frame.

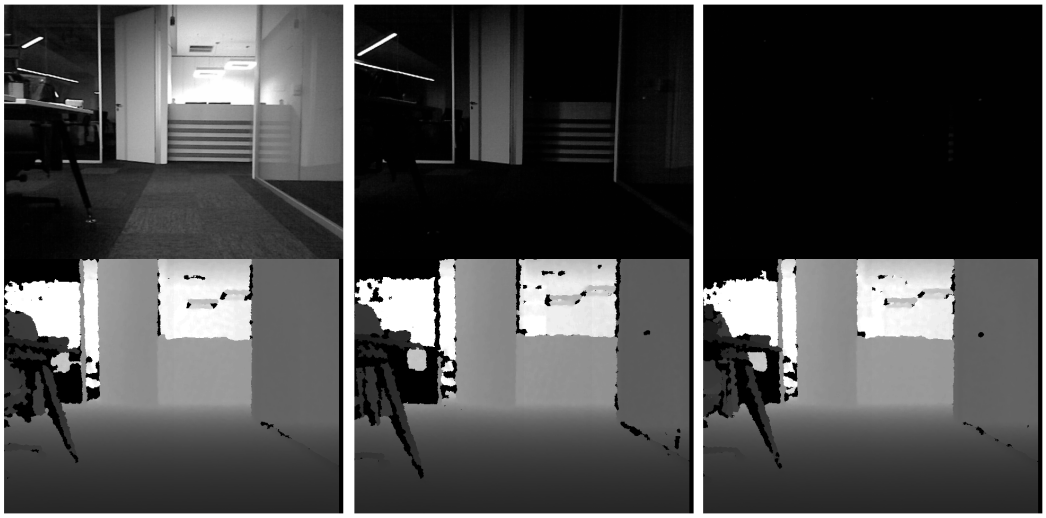
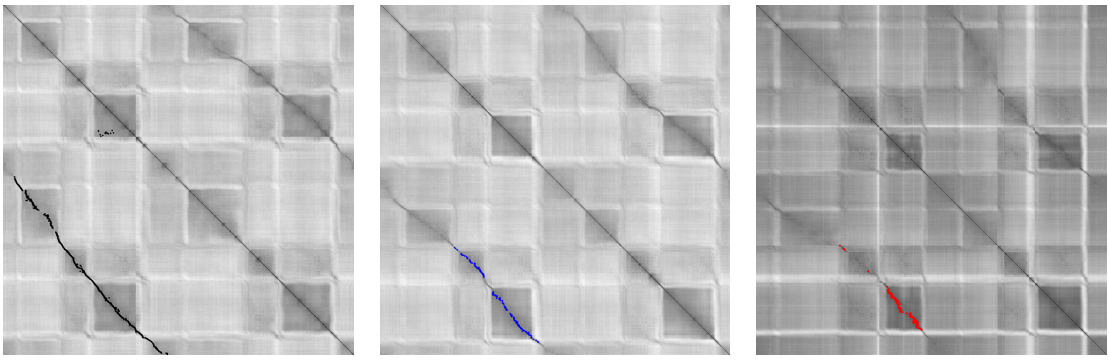


Figure 2: Dataset examples: (right) bright, (middle) dim, (left) dark

4. RESULTS

4.1 Detection Performance

4.2 Real-Time Performance



(a) Bright distances (b) Dim distances (c) Dark distances

Figure 3: Distance matrixes

5. CONCLUSION AND FUTURE WORK

ACKNOWLEDGMENTS

REFERENCES

[1] B. Williams, M. Cummins, J. Neira, P. Newman, I. Reid, and J. Tardós, "A comparison of loop closing techniques in monocular slam," *Robotics and Autonomous Systems* , 2009.

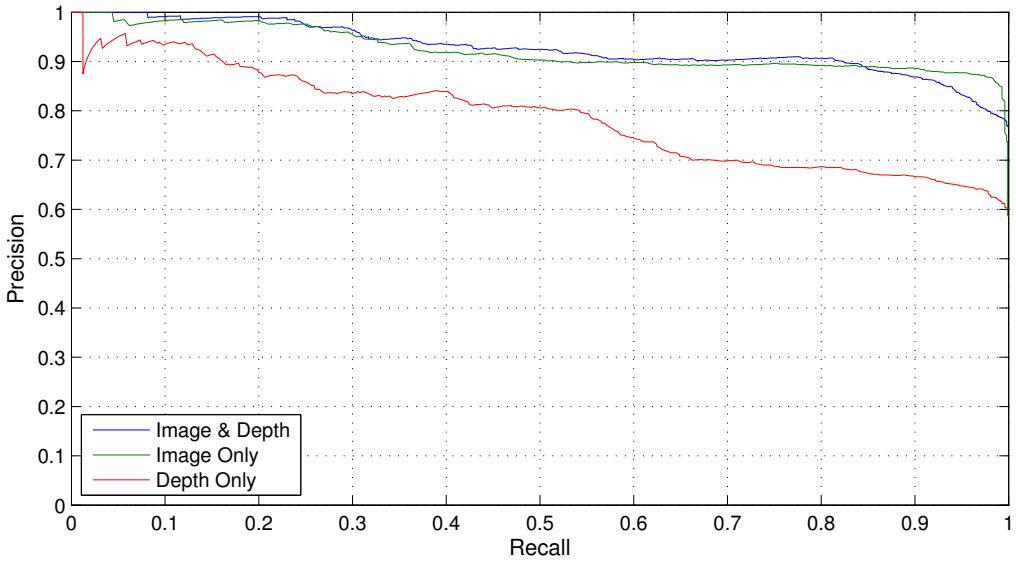


Figure 4: Bright dataset results

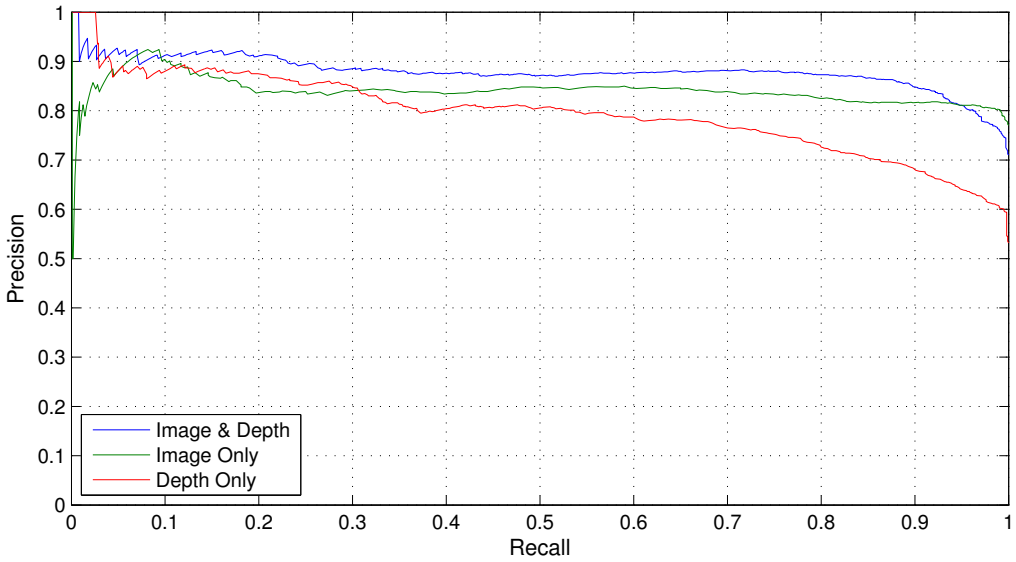


Figure 5: Dim dataset results

[2] C. Cadena, D. Ga andlvez Lo andpez, F. Ramos, J. Tardo ands, and J. Neira, "Robust place recognition with stereo cameras," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pp. 5182 –5189, Oct. 2010.

[3] M. Cummins and P. Newman, "FAB-MAP: Probabilistic Localization and Mapping in the Space of Appearance," *The International Journal of Robotics Research* **27**(6), pp. 647–665, 2008.

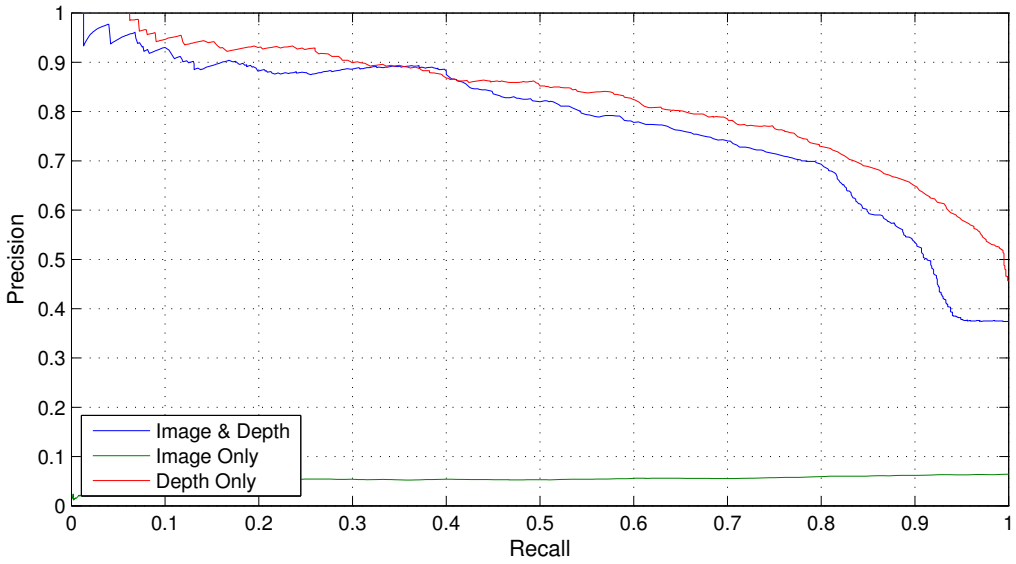


Figure 6: Dark dataset results