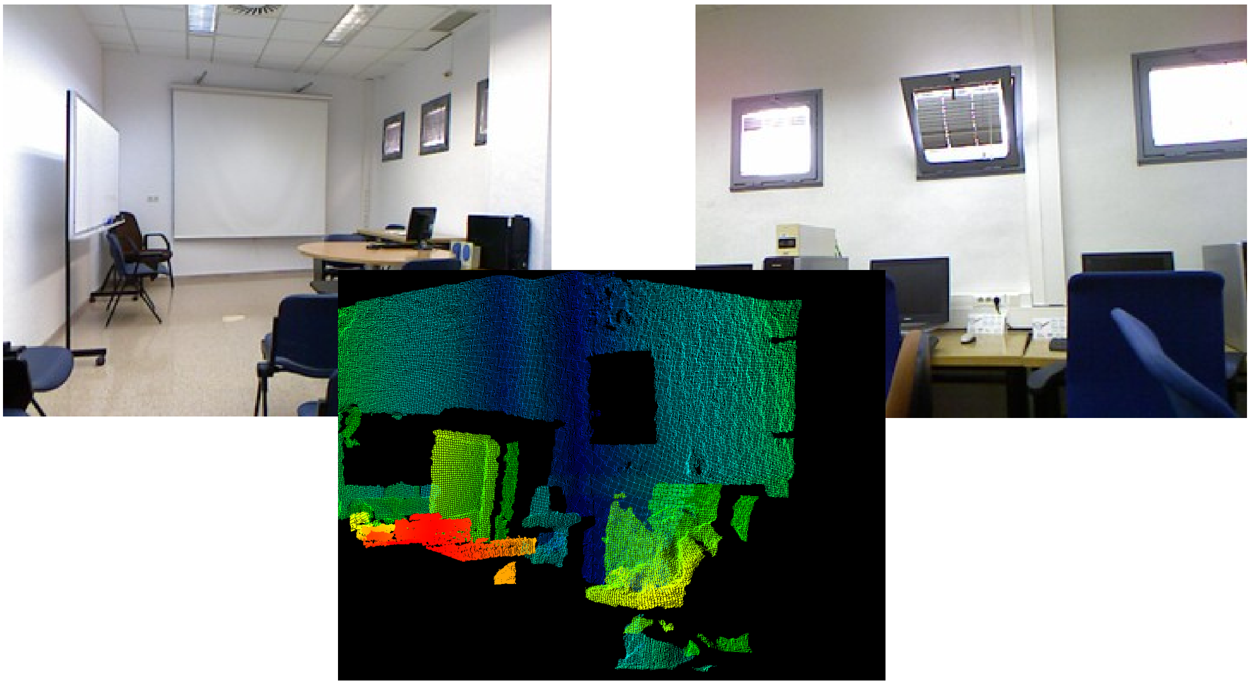
ViDRILO Toolbox: Guide of Use



UCLM
UNIVERSIDAD DE CASTILLA-LA MANCHA

Universitat d'Alacant
Universidad de Alicante

Index

Introduction

The ViDriLO dataset consists of five different sequences of annotated RGB-D images that can be downloaded from www.rovit.ua.es/dataset/vidrilo.

In order to process these sequences, a set of MATLAB tools is released in conjunction to the dataset.

Authorship

The Vidrilo toolbox and dataset has been developed by

- Jesus Martinez-Gomez.- Universidad de Castilla-La Mancha (Spain)
- Ismael Garcia-Varea.- Universidad de Castilla-La Mancha (Spain)
- Miguel Cazorla.- Universidad de Alicante (Spain)
- Vicente Morell.- Universidad de Alicante (Spain)

Files

The ViDriLo toolbox has the following .m files:

- runViDriLoClassifier
  - Description: trains a multi-class classifier for the room classification problem and 15 binary classifiers for the object recognition problem. These classifiers are trained using a specific sequence from the dataset and evaluated again another sequence. The classification model and features to be extracted from RGB-D images should be explicitly established in the call.
  - Params
    - Index of training sequence [1, 2, 3, 4 or 5]
    - Index of test sequence [1, 2, 3, 4 or 5]
    - Source of information
      - 0/'visual' → visual information (only perspective images), 1/'depth' → depth information (extracted from .pcd files), 2/'both' → visual and depth information
    - Type of classification model
      - 0/'SVM' → Support Vector Machine, 1/'kNN' → k Nearest Neighbor, 2/'RF' → Random Forest
    - Type of visual descriptor (when using 0 or 2 as source of information)
      - 0/'GIST' → GIST features: http://people.csail.mit.edu/torralba/code/spatialenvelope/
      - 1/'PHOG' → Pyramoid Histograms of Oriente Gradients: http://www.robots.ox.ac.uk/~vgg/research/caltech/phog.html

      - 2/'Histo' → Grayscale Histogram
    - Type of depth descriptor (when using 1 or 2 as source of information)
      - 0/'ESF' → Ensemble of Shape Functions: http://pointclouds.org/
      - 1/'Histo' → Depth Histogram
- featureExtraction
  - Description: extracts features from both training and test sequences. These features are then used as input for the subsequence training and classification stages. The extraction will depend on the type of descriptors previously selected.
    - GIST: default implementation, dimensionality 512.
    - PHOG: 2 levels and 30 bins, dimensionality 630.
    - Visual Histograms: 30 bins
    - ESF: default implementation, dimensionality 640
    - Depth Histograms: 30 bins
  - If the flag '`usePrecomputedFeatures`' has been set to '1', the features would be loaded from the configuration file instead of computed from the images.
  - Params: Configuration file
  - Result: FeaturesForTraining, FeaturesForTest
- visualizePointCloud
  - Description: visualizes the information encoded in a point cloud file for a specific frame in the dataset.
  - Params: sequenceIndex, frameIndex, stride, pointSize
- performLearningAndClassification
  - Description: trains a multi-class classifier (room classification) and 15 binary classifiers (object recognition) using the training features and evaluates such classifiers against the test features.
    - SVM Classification (libSVM implementation)
      - Default parameters: exponential chi-square kernel
    - kNN (Matlab implementation)
      - Default parameters: k=7
    - Random Forest (Matlab implementation)
      - Default parameters: 50 decision trees
  - Params: Configuration file, FeaturesForTraining, FeaturesForTest, Results
  - Result: Results
- evaluateResults

- Evaluates the results obtained both for room classification and object recognition. Generates three graphs: confusion matrix (room), precision/recall and ROC curve (object recognition), and generates the following metrics
    - Accuracy
    - Root Mean Squared Error (RMS)
    - Precision
    - Recall
    - F1-Score
    - Area Under the ROC curve (AUC)
    - Precision at recall levels for: 0.25, 0.50, 0.75
- Params: Configuration, Results

- evaluateExternalResults
    - Evaluates the results generated with any system and stored in a csv file with the following format:
        - A line for each test sequence with the format
            - R_i,O_{i1},,O_{i2},O_{i3},O_{i4},O_{i5},O_{i6},O_{i7},O_{i8},O_{i9},O_{i10},O_{i11},O_{i12},O_{i13},O_{i14},O_{i15}
            - where R_i is in the range [1-10]: 1-Corridor, 2-Hall, 3-ProfessorOffice, 4-StudentOffice, 5-TechnicalRoom, 6-Toilet, 7-SecretaryOffice,8-VideoConferenceRoom, 9-Warehouse and 10-ElevatorArea.
            - and O_{ij} should be 0 (the object j does not appear in the scene) or 1 (the object appears in the scene). The order for the objets is: Bench, Extinguisher, Computer, Table, Chair, Board, Printer, Bookshelf, Urinal, Sink, Hand-Dryer, Screen, Trash, Phone and Fridge.
    - Params: test of the index sequence (1...5) and the path to the .csv results file
- loadpcd (loads a binary/plain text point cloud file. Downloaded from: http://www.mathworks.com/matlabcentral/fileexchange/40382-matlab-to-point-cloud-library/content/loadpcd.m)
- showDatasetOverallStats (generates graphs with some statistics from the training and test sequences used in classification)
- setVidriloSequencesPath (established the absolute path to the folder where the sequences have been downloaded)

The configuration file (ConfigurationViDriLo.mat) is a structure with the following fields:

- labelClasses (labels of the rooms imaged in the dataset)
    - Type: array of strings – Values: ['corridor',...'hall']
- labelObjects (labels of the objects imaged in the dataset)
    - Type: array of strings – Values: ['bench',...'extinguisher']

- useDepth (1 if depth information is used for training/test or 0 otherwise)
    - Type: boolean – Value: 1/0
- useVisual (1 if visual information is used for training/test or 0 otherwise)
    - Type: boolean – Value: 1/0
- visualizeImageInfo (1 to visualize frame information while training sequences are loaded)
    - Type: boolean – Value: 1/0
- sequencesPath (Absolute path to the folder where sequences have been downloaded)
    - Type: string – Value: 'C:\ViDriLo\Sequences'
- Sequence1...5
    - visualPath (for each frame, the name of the visual image name)
        - Type: array of cells – Values ['sequence1visual1.png',...,'sequence5visual8412.png']
    - depthPath
        - Type: array of cells – Values ['sequence1depth1.pcd',...,'sequence5depth8412.pcd']
    - class (for each frame, the value of the room category)
        - Type: array of doubles (Nx1) – Values [1,1,1... 6,6]
    - objects (for each frame and object index, 1 if the object appears in the scene or 0 otherwise)
        - Type: array of doubles (Nx15) – Values [1 0 1 0 1 1 0 1 0 1 1 0 1 0 1,...,0 0 1 0 0 1 0 1 1 0 0 0 1 1 0]
    - name (name of the sequence)
        - Type: string – Values: 'Sequence1' or 'Sequence2'...or 'Sequence5'
    - path (relative path of the sequence)
        - Type: string – Values: 'Sequence1' or 'Sequence2'...or 'Sequence5'
- strideVisualization3D (used to visualize just a percentage of points when a point cloud file is plotted)
    - Type: int – Value: 250

Installation

The ViDriLo toolbox can be directly downloaded and unzipped in any desired folder of the operative system (let us call $ViDriLo_folder). The toolbox has a configuration file with the overall dataset configuration and a set of .m function files.

The toolbox makes use of the external libSVM library. This library needs to be properly configured by opening Matlab and then opening the $ViDriLo_folder/libsvm/matlab folder. Once we are placed in such folder, we have to execute the make.m function to compile the library.

The loadpcd.m file has been downloaded from the following project https://code.google.com/p/matlab-toolboxes-robotics-vision/. Please visit such project for any doubt or problems, as well as for updates.

The PHOG code has been downloaded from the following page: http://people.csail.mit.edu/torralba/code/spatialenvelope/

The GIST code has been download from the following page: http://www.robots.ox.ac.uk/~vgg/research/caltech/phog.html

In order to compute ESF features, the Point Cloud Library should be previously installed (http://pointclouds.org/). Once it has been installed, open a terminal and move to $ViDriLo_folder/pcl/extractESFVidrilo. Then, you can find a source file to be compiled by typing:

- cmake ./src/
- make

This would generate a esf_extraction executable file that should be moved to $ViDriLo_folder/pcl/

The sequences of the dataset can be downloaded in any folder of the operative system (let us call $ViDriLo_sequences_folder). This folder should present five different folders named Sequence1,...,Sequence5. Each of these folders needs to have a folder named visualInformation/ with the visual images and an additional one named depthInformation/ with the point cloud files.

The $ViDriLo_sequences_folder has to be **explicitly established** in the Configuration file. This can be done using the setVidriloSequencesPath function as follows:

- setVidriloSequencesPath('/home/user/ViDriLo/Sequences') in Linux or
- setVidriloSequencesPath('C:/ViDriLo/Sequences') in Windows

Using precomputed features

The toolbox is released with an Configuration file where all the descriptors have been previously pre-computed. This allows users to train and evaluate any classification system without the need of downloading the RGB-D sequences and computing the descriptors from these images.

In order to use the precomputed features, download the 'ConfigurationVidriloPrecomputedFeatures.mat' file from http://www.rovit.ua.es/dataset/vidrilo/downloads.html in $ViDriLo_folder and rename it as 'ConfigurationVidrilo.mat' to replace the original configuration file.

Functionality

Classification

The main functionality included in the toolbox is the generation of a multimodal semantic localization system for both room classification and object recognition problems.

This functionality is provided by calling the runVidriloClassifier function, which internally invokes other functions (as showDataSetStats, featureExtraction, performLearningAndClassification or evaluateResults).

The runVidriloClassifier function accepts up to 6 values as parameters. The two firsts are the indexes of the sequences used for training and test respectively. The third parameter encodes the type of information used for training: visual information (0), depth information (1) or both (2). The fourth parameter selects the classification model: SVM (0), kNN (1) and Random Forest (2). The fifth parameter selects the visual descriptor: GIST (0), PHOG (1) or Grayscale Histogram (2) and the sixth the depth one: ESF (0) or Depth Histogram (1).

Once the runVidriloClassifier function has been invoked, the following steps are carried out:

1. Visualize information about the objects and rooms included in both the training and the test sequences (Figure 1).

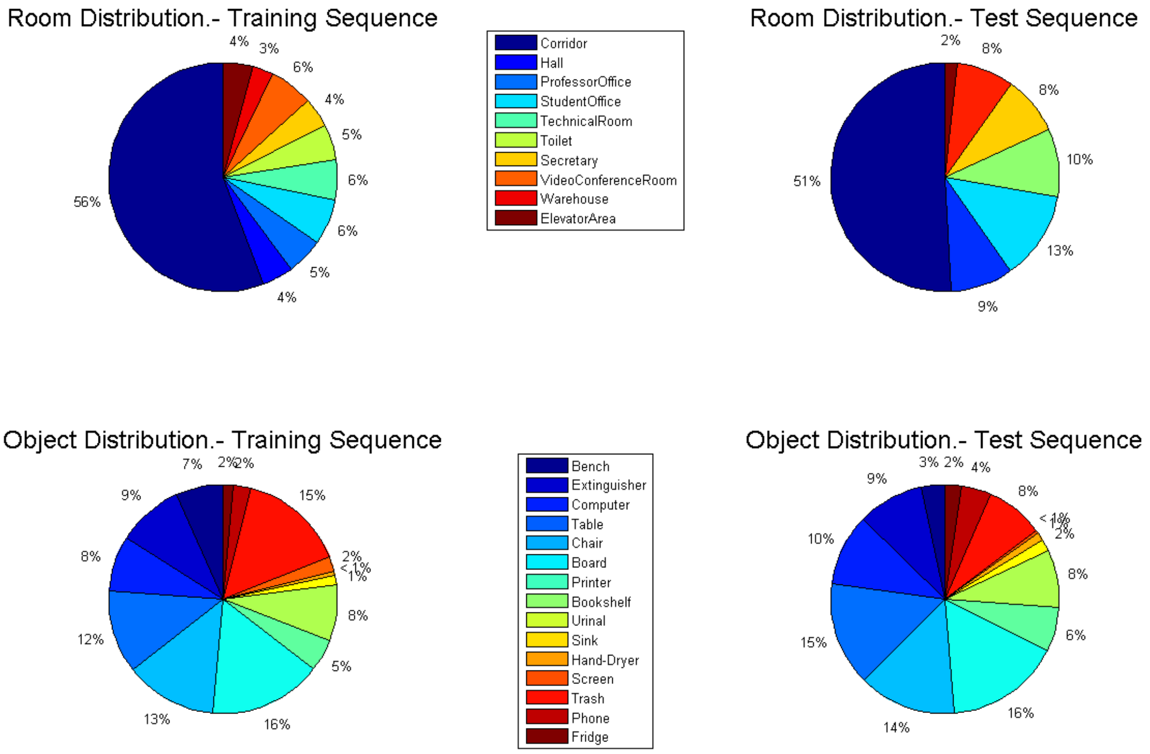


Figure 1: Training and Test Sequences Rooms and Objects Distribution

2. Extract features[1]. Each visual image and/or point cloud file (depending on the third parameter) from the training or test dataset sequence is processed to extract a valid descriptor. The type of descriptor will depend of the 5th and 6th parameters.
    1. Before extracting any descriptor, it is checked in the filesystem if such feature has been previously computed. In such case, the feature is loaded instead of extracted.
    2. Once a feature has been extracted, it is saved in a folder (created automatically) that encodes the type of features, the sequence and the frame number.

       Example: $ViDriLo_folder/VidriloData/features/visual/histo/sequence1visual1.mat
    3. If the visualizeImageInfo option in the configuration file is 1, we visualize a depth and colour representation of each frame when it is loaded (Figure 2).

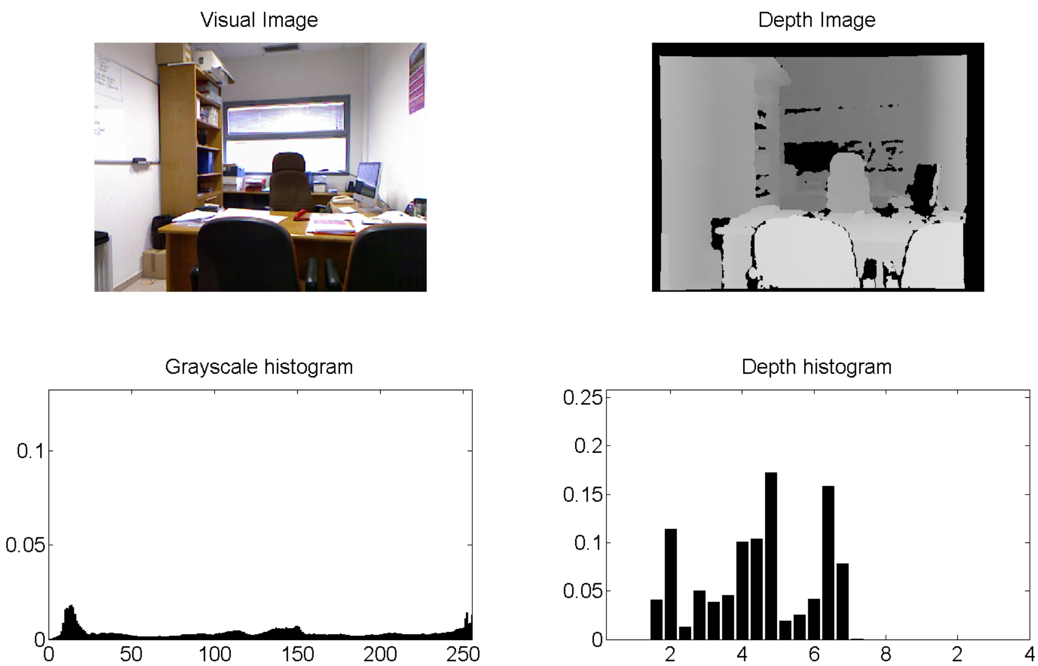


Figure 2: Frame Information: Visual and Depth images, and grayscale and distance histograms

3. Learning and Classification.
    1. Using the extracted training features and the frame room categories, we train a multi-class classifier depending on the user selection:
        - SVM using the expChi-kernel and default libSVM parameters.
        - KNN with k=7
        - Random Forest with 50 decision trees
    2. Using the extracted training features and the frame object presences/lacks, we train 15 binary classifiers depending on the user selection:

1 Features will be automatically obtained from the configuration file (not computed) when using the FLAG Configuration.usePrecomputedFeatures=1 and a Configuration file with such features.

   - SVM Using the exponential chi-square kernel and default libSVM parameters
   - KNN with k=7
   - Random Forest with 50 decision trees

3. All the classifiers are used to process the test features, obtaining for each frame the room category and the list of objects present in the scene.

4. Evaluation of Results. Using the obtained results and the ground truth data (obtained from the configuration file) we evaluate the overall performance of the system. E.g. runVidriloClassifier(2,1,'visual','knn','gist')

   1. For the room classification sub-problem, we generate a confusion matrix graph (Figure 3) and computed the following metrics:

Figure 3: Confusion Matrix Results for Room Classification

Confusion Matrix

	CR	HA	PO	SO	TR	TO	SE	VC	WH	EA
CR	1206	14	6	20	8	44	3	6	8	18
HA	58	5	0	7	4	25	2	0	2	0
PO	38	4	29	24	5	11	3	3	5	2
SO	33	0	2	103	3	2	2	7	3	0
TR	24	0	1	24	73	6	2	2	2	2
TO	38	0	7	3	3	67	0	0	1	2
SE	49	1	0	1	5	2	21	0	9	10
VC	25	0	2	33	3	19	0	41	21	5
WH	6	2	2	11	3	4	1	1	36	4
EA	32	20	1	3	0	1	1	1	2	39

```
### ROOMS CLASSIFICATION - DETAILED RESULTS BY ROOM ###
```

Room	TP Rate	FP Rate	Precision	Recall	F1-Score	ROC Area
CR	0.90473	0.28693	0.79920	0.90473	0.84870	0.80890
HA	0.04854	0.01794	0.10870	0.04854	0.06711	0.51530
PO	0.23387	0.00927	0.58000	0.23387	0.33333	0.61230
SO	0.66452	0.05640	0.44978	0.66452	0.53646	0.80406
TR	0.53676	0.01509	0.68224	0.53676	0.60082	0.76084
TO	0.55372	0.05026	0.37017	0.55372	0.44371	0.75173
SE	0.21429	0.00611	0.60000	0.21429	0.31579	0.60409
VC	0.27517	0.00893	0.67213	0.27517	0.39048	0.63312
WH	0.51429	0.02285	0.40449	0.51429	0.45283	0.74572
EA	0.39000	0.01879	0.47561	0.39000	0.42857	0.68561
W.Avg:	0.67811	0.17068	0.66579	0.67811	0.65374	0.75371

```
### ROOMS CLASSIFICATION - OVERALL RESULTS ###

### ROOMS: WELL CLASSIFIED: 1620.
### ROOMS: BAD CLASSIFIED: 769.
### Accuracy: 67.81.
### Root Mean Squared Error: 0.25373
```

2. For the object recognition sub-problem, we generate a graph where the precision/recall metric is shown for each one of the 15 objects (Figure 4), another graph with the ROC curves (Figure 5), and the following metrics

```
### OBJECT RECOGNITION - DETAILED RESULTS BY OBJECT ###
```

Object	TP Rate	FP Rate	Precision	Recall	F1-Score	ROC Area
Ben	0.29921	0.03747	0.48718	0.29921	0.37073	0.63087
Ext	0.50000	0.15513	0.35772	0.50000	0.41706	0.67243
Com	0.67340	0.04015	0.70423	0.67340	0.68847	0.81662
Tab	0.70615	0.04359	0.78481	0.70615	0.74341	0.83128
Cha	0.61741	0.03799	0.80902	0.61741	0.70034	0.78971
Boa	0.46269	0.12486	0.55578	0.46269	0.50498	0.66891
Pri	0.41143	0.02800	0.53731	0.41143	0.46602	0.69171
Boo	0.67687	0.04964	0.65677	0.67687	0.66667	0.81361
Uri	0.55556	0.03169	0.28846	0.55556	0.37975	0.76193
Sin	0.31579	0.00380	0.40000	0.31579	0.35294	0.65600
Han	0.00000	0.00000	0.00000	0.00000	0.00000	0.50000
Scr	0.15854	0.01040	0.35135	0.15854	0.21849	0.57407
Tra	0.26011	0.08901	0.47742	0.26011	0.33675	0.58555
Pho	0.08889	0.01000	0.25806	0.08889	0.13223	0.53944
Fri	0.66667	0.01329	0.55072	0.66667	0.60317	0.82669
W.Avg:	0.49219	0.06978	0.58309	0.49219	0.52470	0.71121

```
### OBJECT RECOGNITION - OVERALL RESULTS ###

### OBJECTS: TOTAL NUMBER OF OBJECTS WELL DETECTED: 1860 ###
### OBJECTS: TOTAL NUMBER OF OBJECTS BAD DETECTED: 1349 ###
### OBJECTS: TOTAL NUMBER OF OBJECTS NOT DETECTED: 1919 ###

### Root Mean Squared Error: 0.30199.
### Average Precision: 0.58.
### Average Recall: 0.49.
### Average F1 score: 0.53.
### Average Area Under Curve (ROC): 0.69.
```

```
### Average Precision at 0.25 Recall Level: 0.20.
### Average Precision at 0.50 Recall Level: 0.38.
### Average Precision at 0.75 Recall Level: 0.48.
```

Figure 4: Precision Recall Graph for the object recognition problem

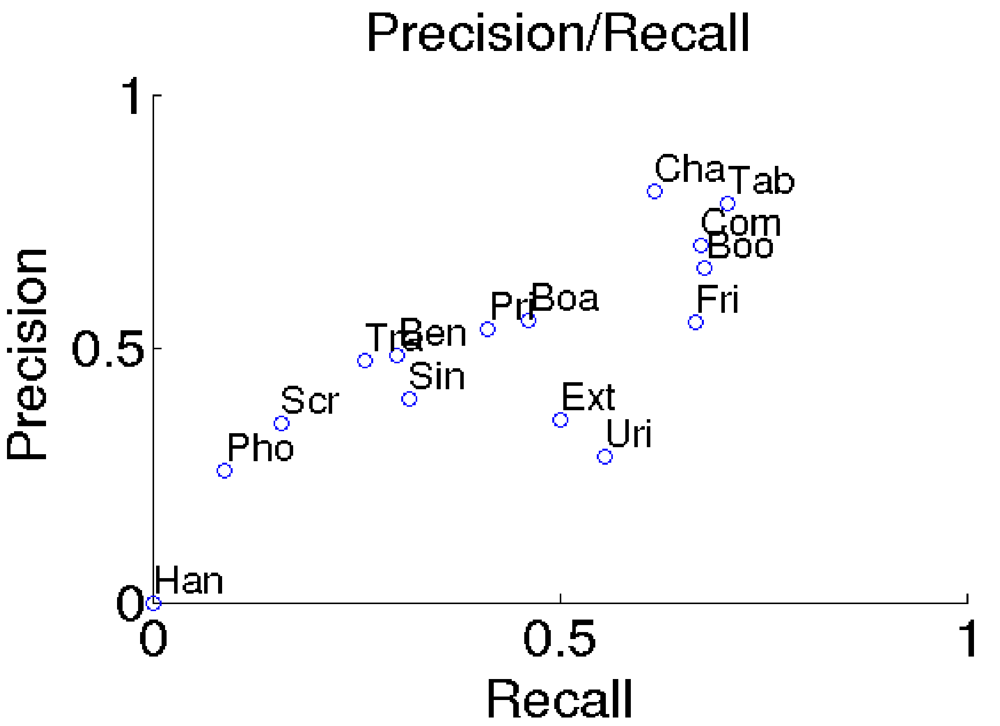
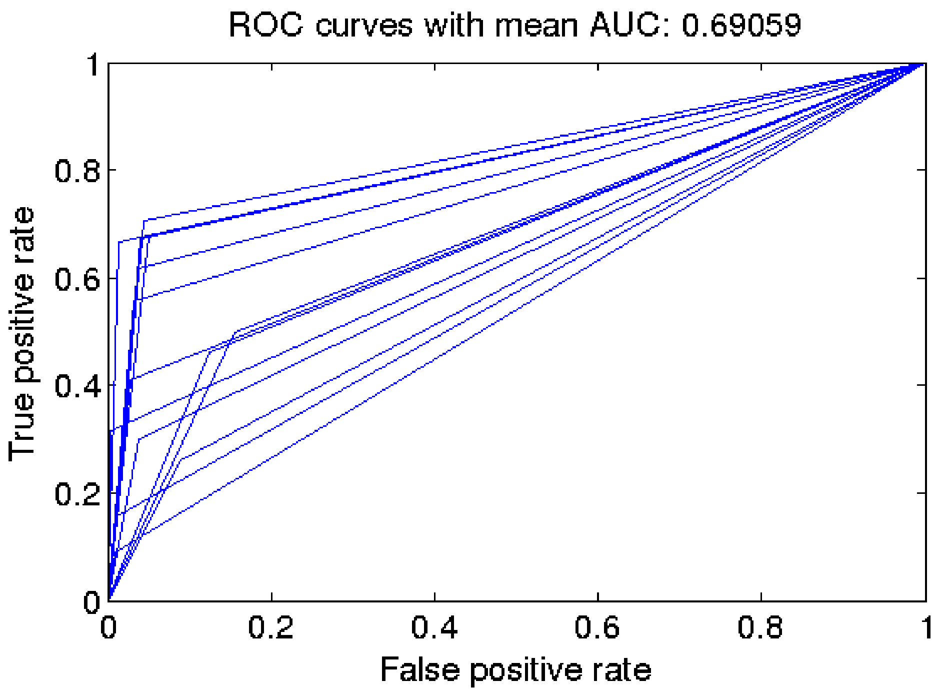


Figure 5: Roc Curves for Object Recognition



Visualization

Point Cloud Files

Another functionality included in the dataset is the visualization of point cloud files as Matlab figure. In order to achieve this goal, we take advantage of the loadpcd.m function (obtained from http://www.mathworks.com/matlabcentral/fileexchange/40382-matlab-to-point-cloud-library/content/loadpcd.m) and the Matlab scatter3 function.

Taking into account that point cloud files include information from 307200 different points (640x480), the complete visualization of the cloud would result in very complex task. Therefore, we only visualize a percentage of these points using a parameter included in the configuration file: strideVisualization3D. Namely, only one out of 'strideVisualization3D' points are visualized. The size of the points in the image are automatically adjusted to avoid showing large spaces between them (See Figures 6, 7 and 8).

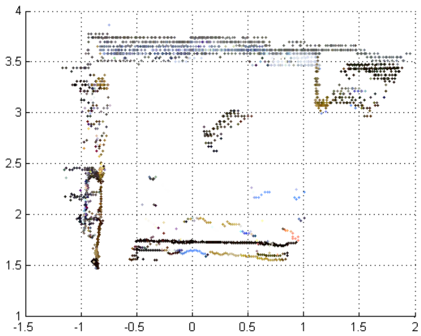


Figure 6: Stride = 100

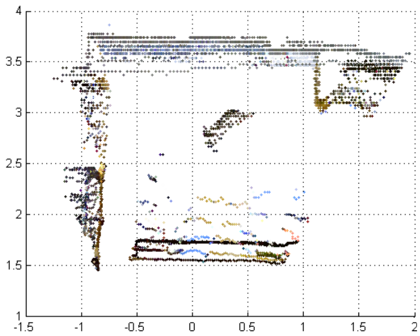


Figure 7: Stride = 50

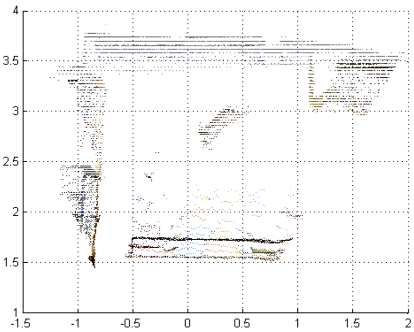


Figure 8: Stride = 30

Once the figure has been create, the user can interact by rotating the scene.

Figure 9: Visual Image

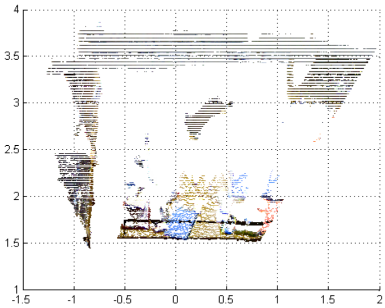


Figure 10: Top View

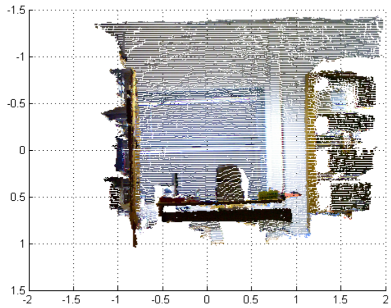


Figure 11: Front View

Dataset Overall Statistics

We can also visualize the following statistics from the complete dataset:

- Rooms Distribution (Figure 12)
- Objects Distribution (Figure 12)

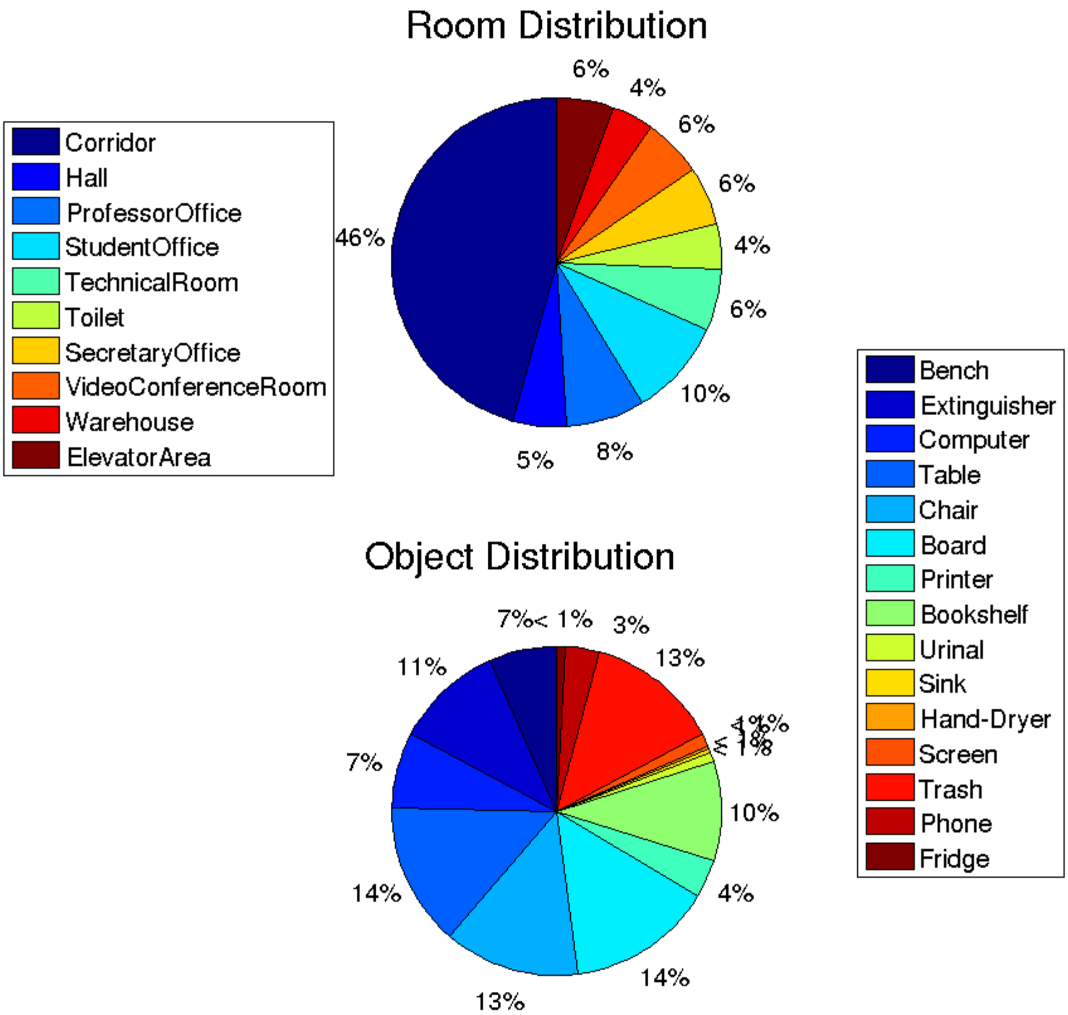


Figure 12: Overall distribution for rooms and objects appearances

- Probability of being placed at room *r* if the scene contains the object *o* (Figure 13)
- *Probability of containing the object o if the scene belongs to the room category r* (Figure 14)

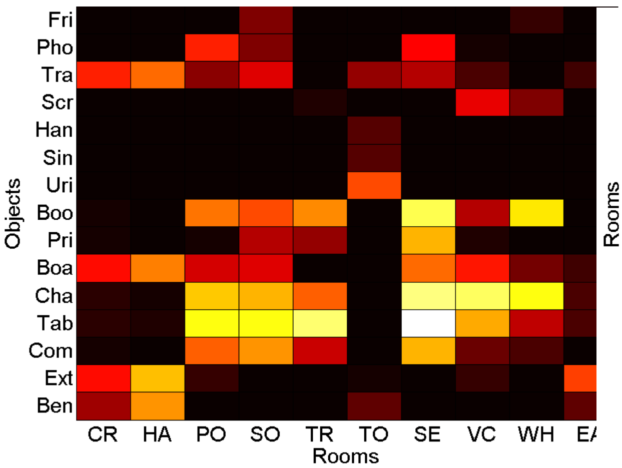


Figure 13: Probability of object appearance given the room

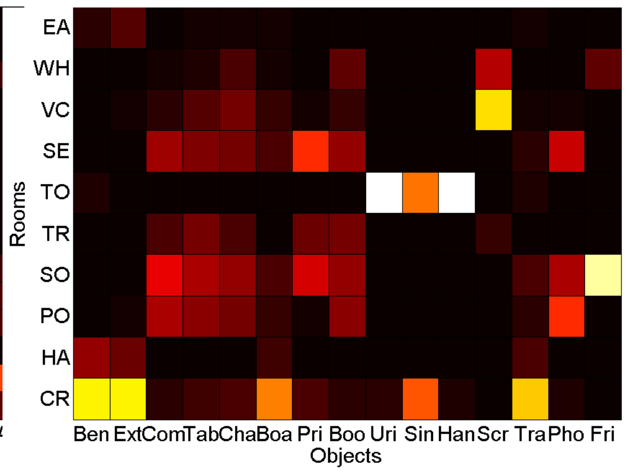


Figure 14: Probability of room given the object appearance

This statistics are very useful from a semantic point of view, due to they allow to visualize the string relationships between rooms and objects, as well as the prior probabilities.

Citation

If you want to cite the Vidrilo dataset, please include the following reference:

J. Martinez-Gomez, I. Garcia-Varea, M. Cazorla and V. Morell.- “ViDRILO: The Visual and Depth Robot Indoor Localization with Objects information Dataset”

License

The Vidrilo toolbox is distributed under GPL license.

Contact

For any doubt related to the dataset or the toolbox, please contact:

- Jesus Martinez-Gomez: jesus.martinez@uclm.es

For any doubt related to the website or the download section, please contact:

- Miguel Cazorla: miguel.cazorla@ua.es