Affordance-map: Mapping Human Context in 3D Scenes Using Cost-Sensitive SVM and Virtual Human Models

Lasitha Piyathilaka[1] and Sarath Kodagoda[2]

***Abstract*— Robots are often required to operate in environments where humans are not present, but yet require the human context information for better human robot interaction. Even humans are present in the environment, detecting their presence in cluttered environments could be challenging. As a solution to this problem, this paper presents the concept of affordance-map which learns human context by looking at geometric features of the environment. Instead of observing real humans to learn human context, it uses virtual human models and their relationships with the environment to map hidden human affordances in 3D scenes. The affordance-map learning problem is formulated as a multi label classification problem that can be learned using cost-sensitive SVM. Experiments carried out in a real 3D scene dataset recorded promising results and proved the applicability of affordance-map for mapping human context.**

I. INTRODUCTION

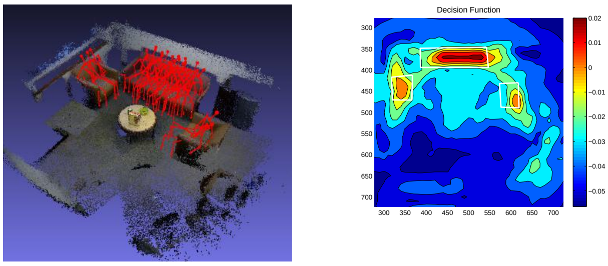
In robotics, learning human context often involves tracking humans to learn their motion patterns [1], human activity detection [2], [3] and modeling relationships between humans and their surroundings [4]. Almost all of these techniques require robots to detect and track humans for a considerable amount of time before it is being used to model human context. On the other hand, detecting, tracking and activity detection of humans in a cluttered environment are still largely open problems. Further, these tasks become more challenging and complicated when robots have to accomplish them while moving in a socially acceptable manner. Often these existing techniques require a considerable amount of re-engineering when they are introduced into new environments.

Introduced by Gibson in 1977 [5] affordance theory defines the word “affordance” as all action possibilities latent in the environment, objectively measurable and independent of the individual’s ability to recognize them [5], [6], [7]. Affordance theory argues that action possibilities are motivated by how environment is arranged. For example, chairs and sofas support the activity ‘sitting’ and they are physically designed to support that affordance which encourages the actor for sitting. Therefore, we believe this strong relationship between human context and environmental affordances could be used to learn human context even when humans are not observable. The rationale here is that, it is possible to learn environmental affordances by only looking at geometric features of the environment and this form the basis for the concept called affordane-map which is introduced in this paper. Affordance-map involves mapping possible human affordances in 3D scenes though virtual human models. This affordance-map learns the human context in a given environment without observing any real humans and bypasses challenges associated with human detection.

As robots use grid based maps for localization, path planning and obstacle detection, affordance-map could be used by a robot to improve the human robot interaction. Therefore, service robots operating in indoor environments could largely benefit from learning the hidden human context. For example, a domestic service robot could use the human context information embedded in an affordance-map to arrange objects in a human preferred manner before humans arrive from work. Or, it could use pose information of virtual humans models embedded in affordance-map to search and localize various objects. Even when humans are present in the environment, affordance-map could be used by a robot to carry out its task more efficiently. Firstly, it could use affordance-map to infer the possible human locations which could be used by the robot to efficiently detect humans. Secondly, it could use information from affordance-map to plan paths that minimize inferences to humans. Even affordance-map could provide strong priors for human activity detection when humans are partially observed or the views of them are completely obstructed.

II. MAPPING AFFORDANCES

Affordance-map is consisted of two sub-maps as shown in Fig. 1. First, it predicts virtual human skeleton models in locations that support the tested affordance as shown in Fig. 1(a). Then it outputs a confidence value for each predicted skeleton that indicates how likely the skeleton model is being supported by the surrounding environment as shown with a contour map in Fig. 1(b).



(a) Vitual Skeleton Map

(b) The contour map of affordance likelihoods

Fig. 1. The affordance-map of a living room

The affordance-map building process can be formulated as

[1] Lasitha Piyathilaka and [2] Sarath Kodagoda are with the Center of Autonomous Systems, University of Technology Sydney, Australia `sarath.kodagoda@uts.edu.au`

a supervised learning problem. Given a set of 3D pointcloud images $\{i_1, ..., i_k\} \subset I$ and their associated affordance-map A, the goal is to learn mapping $g : I \rightarrow A$ which can be later used to automatically build affordance-map in unseen pointclouds. As some locations of the map could have multiple affordances-labels, mapping $g : I \rightarrow A$ becomes a multi label classification problem. As affordance types are not mutually exclusive, one way to simplify this problem is to divide each 3D image space into four dimensional grid locations ((x,y,z) location of the skeleton model and θ orientation) and build an independent binary classifier for each affordance type. Each binary classifier predicts a binary label vector $\bar{\mathbf{y}}_k = (y_1, .y_i.., y_n)$ where $y_i \in \{+1, -1\}$ for the feature vector $\bar{\mathbf{x}}_k = (x_1, ..., x_n)$, $x_i \in \Re^n$ calculated for each grid location of the 3D scene. The label y_i becomes $+1$ if that location support the tested affordance and -1 if not.

A. SVM Classifier for Affordance-Detection

Support Vector Machines (SVMs) [8], [9], [10] is a popular machine learning technique, which has been successfully applied to many real-world application domains. The goal of the SVM learning algorithm is to find the optimal separating hyperplane which effectively separates instances into two classes. SVM algorithm is a discriminative binary classifier and it's decision function can be used to assign confidence values for its predictions. Therefore, SVM classifier is selected as the binary classifier for mapping affordances.

B. Feature selection

The proposed affordance-map predicts virtual skeleton models with likelihood values for each (x_i, y_i, θ_i) grid location of the 3D image. It is based on a binary classifier that predicts positive labels to the locations that support the tested affordance and negative labels to the locations that do not support the tested affordance. Therefore, the classifier's performance largely depends on the types of features used. These features should be highly informative such that the classifier would be able to predict class labels correctly. In this paper, a new set of features based on virtual human skeleton models is proposed for mapping affordances. These features directly model the relationship between the humans and the environment. Following sections describe different types of skeleton models and their associated features used by binary classifiers to build the affordance-map.

1) Virtual Human Skeleton Model: Instead of observing real humans in the environment, the proposed affordance mapping process uses virtual humans to model interaction between the environment and the human. Although many human poses could be observed in a given environment, a very few of them directly influence the context of the environment. For example, most frequently observed human pose in an office environment is sitting and working at an office desk. Therefore, if the locations of the office room that supports this affordance can be identified then human context of the environment can be inferred easily.

For the purpose of mapping affordances, human skeleton models are obtained from a human activity detection dataset [11]. These human skeleton models are captured using a depth camera from real humans while they perform different activities. The K-mean clustering is done to cluster all skeleton models to three clusters and the most frequently seen pose in the each cluster is shown in the Fig .2. Each of these skeleton models are associated with an affordance type that closely represents the type of activity that each skeleton model belongs.

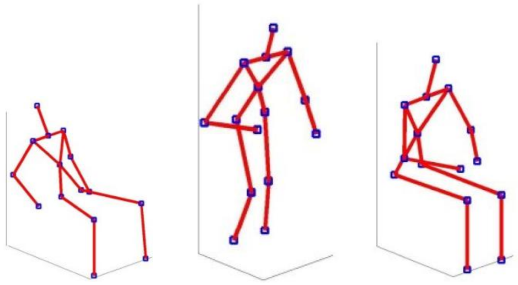

(a) Sitting-Relaxing (b) Standing-Working (c) Sitting-Working

Fig. 2. Types of affordances and their associated skeleton models

Each human model is a skeleton with 15 joint body positions in 3D. In order to obtain feature vectors, these virtual skeleton models need to be transformed to each (x,y,z,θ) pose of the map. This can be done by moving the 3D points of the human skeleton, H_l across given environment using the rigid body transformations of translation and rotation. Then each human skeleton model can be mapped to the coordinate system of the environment using (1), where $\mathbf{g}_k = (x_k, y_k, z_k, \theta_k)$ is the position and orientation of the skeleton's torso in the world coordinate system and $R_z(\theta_k)$ is the rotational matrix about z axis (vertically up). It is to be noted that only rotation about z axis is considered here.

$$H_w(\mathbf{g}_k) = [x_k, y_k, z_k]^T + R_z(\theta_k).H_l \tag{1}$$

2) Distance and Collision Features: The features model the relationship between the human skeletons and the environment. This relationship is modeled through two geometric features: distance features and collision features. Selection of these features are motivated by two facts. First is the proximity of objects for effective interactions and the second is to prevent collisions or intersects with occupied voxels of the environment.

The first step of feature extraction involves modeling the environment. The 3D point clouds generated from RGBD SLAM algorithms [12] usually contain a large amount of 3D points, and searching for a particular affordance in this large feature space is computationally infeasible. Therefore, it is required to convert these dense point clouds into much lighter abstract representations without losing much data. This is achieved by modeling the environment by a 3D Distance Transform Map $DT(\mathbf{x})$ and a 3D Occupancy Map $OC(\mathbf{x})$, where $\mathbf{x}$ is any 3D position of the environment. The 3D Distance Transform (DT) is a shape representation that indicates the minimum distance from a point in the environment to the closet occupied voxel. In our approach, we calculated 3D Distance Transform by using the occupied voxels of 3D point clouds, OC. The distance transform map $DT(\mathbf{x})$ of the occupancy grid map OC can be generated using an unsigned distance function (2), that represents Euclidean distance from each location $\mathbf{x}$ of the environment to the nearest occupied voxel in $OC(\mathbf{x})$.

$$DT(\mathbf{x}) = \underset{O_j \in OC}{min} |O_j - \mathbf{x}| \tag{2}$$

The distance features are obtained by moving the human model across the voxels in the environment and calculating a distance measure for each and every skeleton points of the human model. Once the environment is modelled by (2), we can effectively calculate distance features of a human skeleton with location and orientation $\mathbf{g}_k = (x_k, y_k, z_k, \theta_k)$ by (3), where n is the number of 3D points in the skeleton.

$$[d_1, d_2, ...d_n] = DT(H_w(\mathbf{g}_k)) \tag{3}$$

Similarly, we can check for any collisions of a skeleton at any location and orientation, X_k by (4). In case of a collision c_i is assigned as 1 and 0 otherwise.

$$[c_1, c_2, ...c_n] = OC(H_w(\mathbf{g}_k)) \tag{4}$$

3) Normal Features: The other set of features used for affordance detection is normal features. These normal features represents vertical and horizontal planes of the environments. The selection of these features is motivated by the fact that most of the affordances are supported by vertical and horizontal planes. For example 'sitting' affordance is supported by a horizontal plane under the lap of a sitting skeleton and the spine of the skeleton is supported by a vertical plane.

The normals features for the affordance detection are calculated as follows. First, a 1m x1m x 1m cubic volume is considered from the torso position of a skeleton model at location, (x, y, z, θ). Then it is voxelised into 10cm x 10cm x 10cm voxels. Finally surface normal values of points in each grid cell are averaged to find the normal features for each voxel.

III. Experimental Setup

This section explains the experimental setup used for mapping affordances in 3D pointclouds.

A. Dataset

Both the SVM and Flexible Naive Bayes classifier discussed in previous sections are supervised learning algorithms. Therefore, they need to be trained first before used for building an affordance-map in an unseen environment and require a number of 3D scenes to learn parameters of the classification models. The recent advancements in RGBD Simultaneous Localization and Map Building (SLAM) algorithms ([12]) allows us to build highly informative dense 3D scenes by stitching 3D point clouds frames acquired from low cost 3D depth cameras. A dataset is created for affordance mapping by capturing a number of dense 3D scenes using a ASUS Xition depth camera and a 3D map building software [12]. The final dataset is consisted of 12 high quality 3D scenes captured in office and domestic environments.

TABLE I

Summary of the Class Imbalance in the Dataset

Affordance	**# Positive Examples**	**# Negative Examples**	**Imabalance Ratio**
Sitting-Relaxing	2457	1285326	1:523
Sitting-Working	213	2570439	1:12067
Standing-Working	391	1284935	1:3286

B. Search space

The affordance map is a concatenation of classifier's outputs for each (x, y, z, θ) grid location of the 3D scene. Therefore, the first step of the affordance mapping involves voxelising the input 3D images into grid locations. This voxelising is done by dividing the the input image into 10 cm x 10 cm x 10 cm grids. The rotation θ of the each skeleton model is evaluated in 0.1 rad resolution at each grid level. In order to limit the search space, the grid search along the z axis can be restricted as the human skeletons are always located close to the ground plane.

C. Ground Truth Labels

The proposed affordance-map building process is a supervised learning problem, which requires ground truth labels in order to learn parameters of the classifier. Therefore, all possible locations in 3D images that support the tested affordances are manually labeled for each affordance type.

IV. Soft Margin Optimization with Class Imbalance

The Table. I summarises the number of positive and negative examples found in the dataset for each affordance type. It is clear form this table data that all of the affordance types found in the dataset have a high number of negative examples than the number of positive examples. The sitting-working affordance has recorded the highest class imbalance. This is desirable as the sitting surface of an office chair is very small with compared to the rest of the room. Any affordance detection algorithm should be able to effectively handle these type of class imbalances in order to effectively build affordance map in a large room. However, a few researchers have reported that SVM classifier tend to produce sub-optimum results when trained on highly imbalanced datasets [13], [14].

To understand the impact of the class imbalance on affordance detection, we carried out few experiments and experimental results are shown below. First the dataset is divided into two subsets: training set and validation set. Then SVM classifiers are trained for different class imbalances. These learnt classifiers are tested on unseen 3D images and F1-scores are recorded for each setting. The average F1-scores for each affordance type in different class imbalances are shown in Fig. 3.

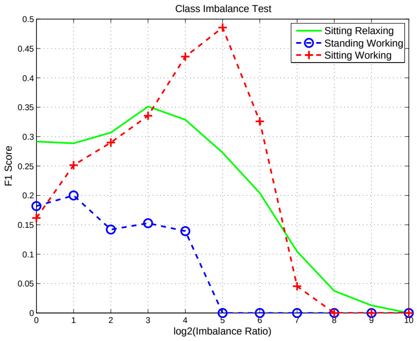


Fig. 3. The effect of class imbalance on the SVM classifier

According to the test results, it is clear that all three affordances have reported low F1- scores initially (when the positive and negative classes are equally balanced) and increased slightly before gradually dropping down again. This behaviour is clearly visible with the sitting-working affordace and less visible with the standing-working affordance. When more negative examples are added for training, the performances of the SVM classifiers have gradually increased as can be seen with increasing F1-scores. This is predicted because more negative examples mean the SVM classifier can discriminate the difference between the positive examples and negative examples more effectively. However, the performances of the SVM classifiers have degraded when the class imbalances become more and more extreme. When the class imbalances are too high all F1-scores have fallen to zero. That means the classifier have predicted all locations of the room with negative labels. Overall, standing-working affordance has recorded low F1-scores.

This unexpected behaviour is due to the movement of the separating hyperplane of the SVM classifier toward the majority class and it can be be understood by analysing the SVM soft-margin optimization problem.

$$\begin{aligned} &\arg\min_{(\mathbf{w},b)}\{\frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i\} \\ &y_i(\mathbf{w}\cdot\Phi(\mathbf{x_i}) - b) \geq 1-\xi_i \\ &s.t \quad \forall i = 1,\ldots,n \end{aligned} \tag{5}$$

The objective function of the standard SVM classifier shown in (5) has two terms. The first part tries to maximize the margin while the second part attempts to minimize the penalty associated with the misclassification. The regularization parameter C, which is a constant, balances the tradeoff between these two terms and can also be considered as the defining factor of the misclassification cost. As both the positive and negative examples are assigned with the same misclassification cost (C), the penalty term is minimized when the misclassification cost of the training samples become minimized. However, for an imbalance dataset with a higher number of negative examples there could be more negative examples even near the class boundary where the ideal hyperplane is passing through. This influences the optimization problem, because in order to minimize the total misclassification cost, the separating hyperplane could shift towards the minority class. This undesirable shift can cause more false negative examples by impacting the performances of the minority class. However, the total misclassification cost and the total error would remain low due to the higher number of true negative examples. In cases like affordance detection, where class imbalance is extreme, the SVM could easily produce a skewed hyperplane which would classify all examples as negative.

A. Cost Sensitive SVM

As it explained in the previous section, the main reason for the inability of soft margin SVM to learn separating hyperplane accurately when dataset is imbalanced would be that it assigns equal cost values for both the minority and majority class misclassification in the penalty term. This would create a biased model with a hyperplane that is skewed towards the minority class. The cost sensitive SVM model proposed in [15] has been designed to mitigate this issues by assigning different penalty terms for the two classes. By assuming that the positive class is the minority class, misclassification cost of C^+ can be assigned for the positive class and C^- cost can be assigned for the negative class in the objective function of the soft margin SVM optimization equation as shown in (6).

$$\begin{aligned} &\arg\min_{(\mathbf{w},b)}\{\frac{1}{2}\|\mathbf{w}\|^2 + C^+\sum_{i|y_i=+1}^{n}\xi_i + C^-\sum_{i|y_i=-1}^{n}\xi_i\} \\ &y_i(\mathbf{w}\cdot\Phi(\mathbf{x_i}) - b) \geq 1-\xi_i \\ &s.t \quad \forall i = 1,\ldots,n \end{aligned} \tag{6}$$

The effect of the imbalanced data is minimized by assigning higher misclassification cost for the positive class (i.e.,$C^+ > C^-$) which eventually would fix the bias of separating hyperplane. Therefore the optimization would try to balance the total positive and total negative cost values and the separating hyperplane would not skew towards the positive class examples. The dual Lagrangian form of this modified objective function can be represented as follows:

$$\begin{aligned} &\max_{\alpha_i}\{\sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i,j}\alpha_i\alpha_j y_i y_j k(\mathbf{x}_i\cdot\mathbf{x}_j)\} \\ &s.t. \quad \sum_{i=1}^{n} y_i\alpha_i = 0 \\ &0 \leq \alpha_i^+ \leq C^+, \quad 0 \leq \alpha_i^- \leq C^- \quad i = 1,\ldots,n \end{aligned} \tag{7}$$

where α_i^+ and α_i^- represent the Lagrangian multipliers of positive and negative examples, respectively. The cost values C^+ and C^- can be chosen by the rule of thumb method proposed in [15].

$$\frac{C^+}{C^-} = \frac{Number \quad of \quad negative \quad examples}{Number \quad of \quad positive \quad examples} \tag{8}$$

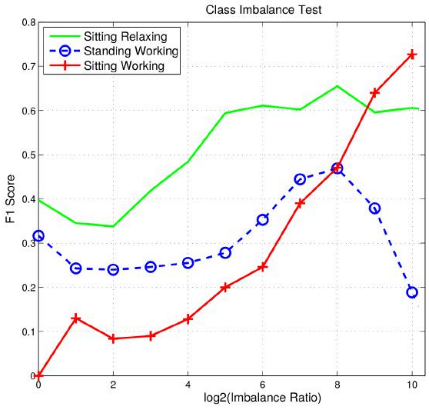


Fig. 4. SVM learning with different cost models

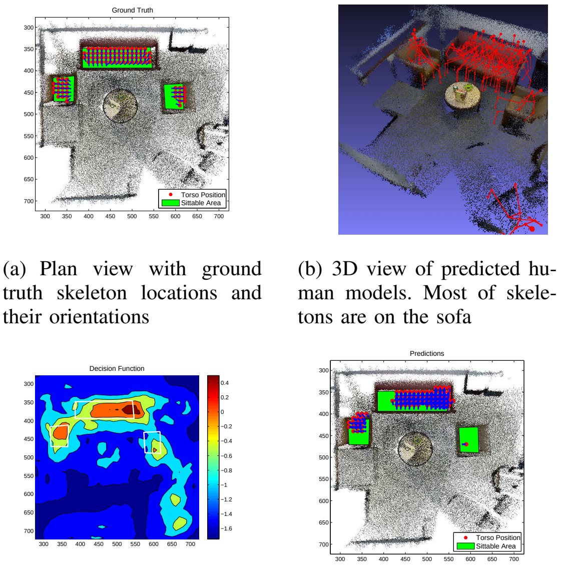
V. Results and Discussion

To evaluate the performances of the SVM classifier, a series of experiments are carried out. First, the data set is divided into three subsets : training set, validation set and testing set. The training set is used to train the model parameters and validation set is used to fine tune the input parameters. Finally, trained classifiers are tested on the testing set. This process reduces the possibility of over-fitting the classifiers.

The behavior of the cost-sensitive SVM classifier in different class imbalances is shown in Fig. 4. Initially F1-scores of all three affordances have gradually increased or tried to stabilize as more and more negative examples are added to the SVM trainer. A steep increase is visible in the sitting-working affordance and has resisted extreme class imbalances well. The sitting-relaxing affordance type has also resisted well for the class imbalance problem. However, F1-score of the standing-working affordance a have declined when the class imbalance become too extreme. Overall, cost sensitive SVM has shown steady resistance to class imbalance problem and recorded better results than the conventional SVM.

The qualitative analysis is done by selecting an 3D image for each affordance from the dataset and comparing prediction results. Both 2D affordance map and 3D skeleton map are used for this purpose.

Fig. 5 shows the the predictions results of the affordance type 'Sitting-Relaxing'. The test room is a living room with a sofa set in the middle and it is marked with green rectangles. In the plan view of the room, red color circles represents torso position of skeleton models and corresponding blue lines denote their orientation. It is clear from these results the affordance-map has predicted most of its skeletons with high confidence values on the sofa set.



(a) Plan view with ground truth skeleton locations and their orientations

(b) 3D view of predicted human models. Most of skeletons are on the sofa

(c) SVM decision function. High confidence values can be seen on the Sofa set

(d) Predicted skeleton locations and their orientations

Fig. 5. Test results for the Sitting-Relaxing Affordance. Best viewed in colour

The 3D view and 2D plan view of the office space used to test the 'sitting-working' affordance is shown in the Fig. 6. It has four office chairs as shown with green rectangle areas. The affordance-map has placed a number of sitting-working human models on office chairs with high confidence scores. It has failed to place any skeleton on the chairs to the top-left of the plan view of the room. However, this chair is not oriented towards the table, and therefore could argue that it is not supporting sitting-working affordance.

The Fig. 7 shows the room used to analyze the performances of the 'standing-working affordance'. It is a lab space with two workbenches and green color rectangle areas indicate workbenches of the lab space. The affordance-map has placed correctly oriented skeleton models near one of the work benches but has failed to predict any skeletons near other workbench. However, the decision function has placed high confidence scores near both workbenches.

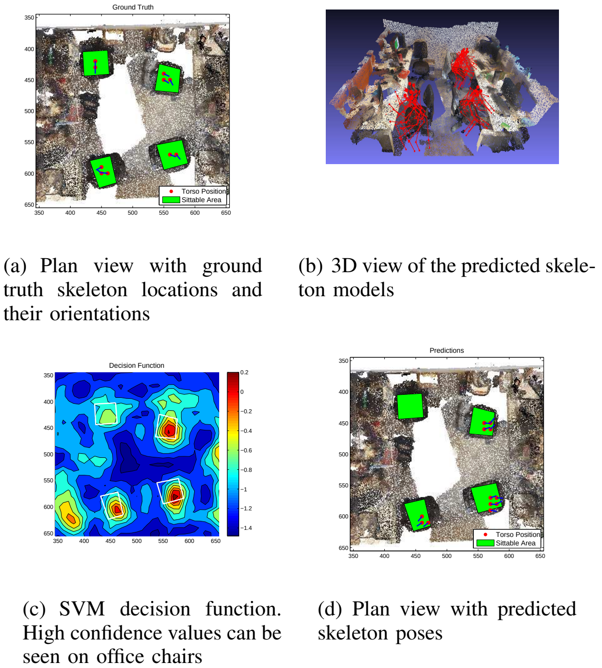
The k-fold cross validation is carried out to report the performances of the two classifiers. In k-fold cross validation, the dataset is divided into k folds and one of them is selected as the testing set. Then the affordances are trained on rest of the $k-1$ folds and finally tested on the left-out set. The k-fold cross validation results for the affordance detection is shown in table II. The sitting-relaxing affordance type has recorded the highest F1-score and standing-working affordance has recorded the lowest F1-score. Therefore it seems that learning the standing-working affordance type is difficult than other two affordance type.

TABLE II

K-fold cross validation results of affordance detection

Affordance Type	Precesion	Recall	F1-score
Sitting-relaxing	0.65	0.68	0.65
Sitting-working	0.50	0.55	0.52
Standing-working	0.15	0.42	0.21

According to the table II the sitting-relaxing and sitting-working affordance types have recorded acceptable results.



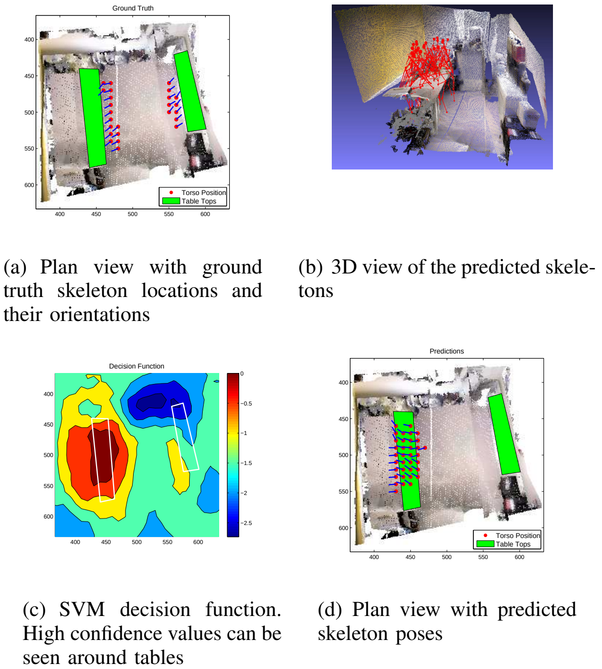
(a) Plan view with ground truth skeleton locations and their orientations

(b) 3D view of the predicted skeleton models

(c) SVM decision function. High confidence values can be seen on office chairs

(d) Plan view with predicted skeleton poses

Fig. 6. Prediction results of the Sitting-Working Affordance. Best viewed in colour



(a) Plan view with ground truth skeleton locations and their orientations

(b) 3D view of the predicted skeletons

(c) SVM decision function. High confidence values can be seen around tables

(d) Plan view with predicted skeleton poses

Fig. 7. Test results of the Standing-Working Affordance. Best viewed in colour

The performances on the standing-working affordance type is moderate. This could be due to the high intra-class variations associated with standing-working affordance type.

VI. Conclusions and Future Work

This paper presented the affordance-map which involves mapping hidden affordances in 3D scenes using geometric features. This problem was formulated as a multi label classification problem with a binary classifier for each affordance-type. The cost-sensitive SVM classifier was proposed as the binary classifier for mapping affordances and it showed acceptable results for the class imbalance problem. The affordance mapping algorithm was tested on real 3D scenes and recorded promising results. However, the performance of the standing-working affordance type was realatively low with compared to other two affordance types. Therefore, our future work involves improving affordance mapping in standing-working affordance type using Structured output SVM algorithm.

References

[1] S. Sehestedt, S. Kodagoda, and G. Dissanayake, “Robot path planning in a social context,” in *Robotics Automation and Mechatronics (RAM), 2010 IEEE Conference on*. IEEE, 2010, pp. 206–211.

[2] L. Piyathilaka and S. Kodagoda, “Human activity recognition for domestic robots,” in *Field and Service Robotics Conference*. Springer, 2013, pp. 567–572.

[3] L. Piyathilaka and S. Kodaagoda, “Gaussian mixture based hmm for human daily activity recognition using 3d skeleton features,” in *Industrial Electronics and Applications (ICIEA), 2013 8th IEEE Conference on*. IEEE, 2013, pp. 567–572.

[4] Y. Jiang, H. Koppula, and A. Saxena, “Hallucinated humans as the hidden context for labeling 3d scenes,” in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*. IEEE, 2013, pp. 2993–3000.

[5] E. J. Gibson and A. D. Pick, *An ecological approach to perceptual learning and development*. Oxford University Press, 2000.

[6] H. Grabner, J. Gall, and L. V. Gool, “What makes a chair a chair?” in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 1529–1536.

[7] L. Piyathilaka and S. Kodagoda, “Active visual object search using affordance-map in real world : A human-centric approach,” in *ICARCV, The 13th International Conference on Control, Automation, Robotics and Vision*. IEEE, 2014.

[8] C. Cortes and V. Vapnik, “Support-vector networks,” *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[9] T. Joachims, “Making large scale svm learning practical,” Universität Dortmund, Tech. Rep., 1999.

[10] J. A. Suykens and J. Vandewalle, “Least squares support vector machine classifiers,” *Neural processing letters*, vol. 9, no. 3, pp. 293–300, 1999.

[11] J. Sung, C. Ponce, B. Selman, and A. Saxena, “Unstructured human activity detection from rgbd images,” in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, 2012, pp. 842–849.

[12] I. Dryanovski, R. G. Valenti, and J. Xiao, “Fast visual odometry and mapping from rgb-d data,” in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 2305–2310.

[13] Y. Tang, Y.-Q. Zhang, N. V. Chawla, and S. Krasser, “Svms modeling for highly imbalanced classification,” *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 39, no. 1, pp. 281–288, 2009.

[14] R. Akbani, S. Kwek, and N. Japkowicz, “Applying support vector machines to imbalanced datasets,” in *Machine Learning: ECML 2004*. Springer, 2004, pp. 39–50.

[15] T. Joachims, “Transductive inference for text classification using support vector machines,” in *ICML*, vol. 99, 1999, pp. 200–209.