Affordance Mining: Forming Perception Through Action

Liam Ellis[1,2], Michael Felsberg[1], and Richard Bowden[2]

[1] CVL, Linköping University, Linköping, Sweden
[2] CVSSP, University of Surrey, Guildford, UK

Abstract. This work employs data mining algorithms to discover visual entities that are strongly associated to autonomously discovered modes of action, in an embodied agent. Mappings are learnt from these perceptual entities, onto the agents action space. In general, low dimensional action spaces are better suited to unsupervised learning than high dimensional percept spaces, allowing for structure to be discovered in the action space, and used to organise the perceptual space. Local feature configurations that are strongly associated to a particular 'type' of action (and not all other action types) are considered likely to be relevant in eliciting that action type. By learning mappings from these relevant features onto the action space, the system is able to respond in real time to novel visual stimuli. The proposed approach is demonstrated on an autonomous navigation task, and the system is shown to identify the relevant visual entities to the task and to generate appropriate responses.

1 Introduction

This paper proposes a method for discovering the visual features that are important to a vision system given a specific problem (e.g. a robotics tasks). This is achieved by first applying unsupervised learning in the problem output space (e.g. the agent's actions). The structure discovered in the output space is then used to organise the input space (e.g. the agent's perceptual representation), in order to form meaningful input representations. This organisation process is achieved by finding strong associations between modes of the output space and configurations of the input space. Association rule data mining algorithms are employed to efficiently find these associations.

This work is motivated by a desire for adaptive cognitive vision systems, that build their own visual representations based on experience and learn how to react to their environment, without the need for explicit definitions of representations or strategies by an engineer. Such emergent systems should be less 'brittle' than conventional hard-coded systems, and demonstrate increased robustness when faced with changes in the environment not envisaged by the engineer.

In natural cognitive systems, increased sensory complexity, along with the machinery used to interpret such complexity, is generally associated with an increasing ability to interact with and manipulate the environment, facilitated by increasing motor capabilities. It is straightforward to see that the complexity of interaction a system can demonstrate - its motor capabilities - is to a certain extent determined by the complexity of its perceptual system. It is, perhaps, less straightforward to see that the

complexity of a systems perceptual system, is determined by the complexity of the systems motor capabilities. However, this apparent cyclical causality, linking perceptual and motor capabilities is supported by a significant body of work in modern cognitive sciences, and has firm philosophical [1] and neurophysiological [2] foundations. In particular the theory of *embodiment*, a term used within psychology, philosophy, robotics and artificial intelligence, is based on the premise that the nature of the mind is determined by the embodiment of the cognitive agent [1] [3]. Related to this is *affordance* theory, that states that the world is perceived not only in terms of object shapes and spatial relationships but also in terms of object possibilities for action [4]. The work presented here demonstrates an embodied approach to constructing an affordance based representation of the world.

Data mining algorithms are useful for efficiently identifying correlations in large symbolic datasets. These methods have begun to be applied to vision tasks such as: identifying features which have high probability of lying on previously unseen instances of an object class [5], mining dense spatio-temporal features for multi-action recognition [6], and finding near duplicate images within a database of photographs [7]. These methods benefit from both the scalability and the efficiency of data mining methods. This work employs data mining algorithms to the novel domain of percept-action association mining. The mechanism of mining frequent and distinctive feature configurations employed here is most similar to that of Quack et al. [5], however, here the discovered configurations are used directly in an action generation process, rather than as a pre-processing step for identifying useful features for other classification techniques. Furthermore, whilst in [5] supervision is required to label the classes of objects that are learnt, in this work, classes of actions are obtained by an unsupervised learning approach.

The rest of this paper is organised as follows: In section 1.1 background to association rule mining is presented. In section 1.2 the robotic platform, training method and intended task are briefly detailed. Section 2 describes the central mechanism of action space clustering and how this identifies classes of actions and percept groupings. Section 3 presents a complete overview of the proposed system, identifying the key processing stages involved, which are presented in detail in sections 4 and 5. Section 4.1 details the approach used to encode visual information as feature configurations and section 4.2 presents the method for finding associations between classes of actions and these feature configurations. Section 5 details how mappings are learnt between associated percept and action data and how these mappings are exploited to generate responses to novel image data. Section 6 presents the experimental evaluation of the system and section 7 contains a discussion and conclusions.

1.1 Association rule mining

Association rule mining is the process of finding association rules in a database $D = \{t_1, t_2, ..., t_m\}$ of transactions, where each transaction is a set of items, and I is the set of all items[1]. An association rule is an implication of the form $X \Rightarrow Y$ where $X, Y \subseteq I$ and $X \cap Y = \emptyset$.

Association rules are selected from the set of all possible rules based on constraints on measures of significance and interest. These constraints are thresholds on itemset *support* and rule *confidence*. The support, $supp(X)$, of an itemset X is defined as the

[1] The terminology *transactions* and *items* comes from the data mining literature, reflecting the subjects origins in market basket analysis applications

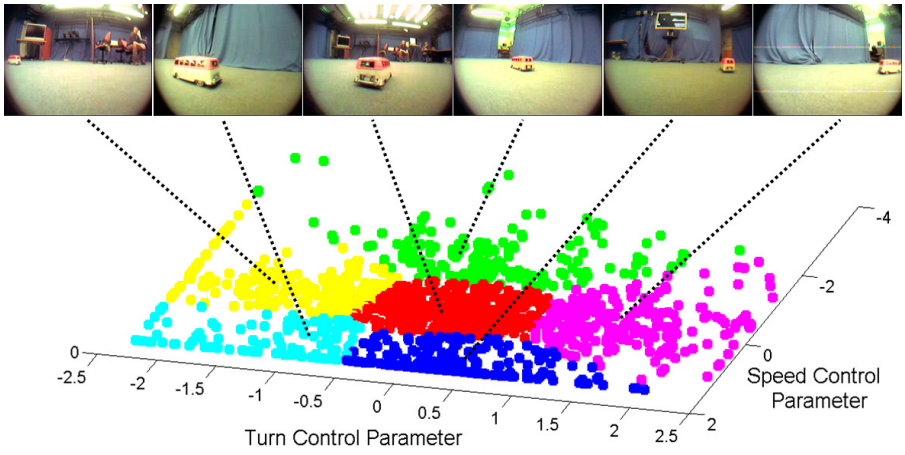


Fig. 1: *Action clustering*: Action clusters are formed along with sets of associated images.

proportion of transactions in the database which contain X. The confidence, $conf(X \Rightarrow Y)$ of a rule is defined:

$$conf(X \Rightarrow Y) = \frac{supp(X \cup Y)}{supp(X)} \tag{1}$$

The Apriori algorithm [8] employed in this work, exploits the anti-monotonicity of the support threshold constraint - that a subset of a frequent itemset must also be a frequent itemset - to efficiently mine association rules. This work uses an efficient existing implementation of the Apriori algorithm [9].

1.2 Robotic platform and training data collection

The robotic platform developed is a relatively inexpensive platform for the investigation of embodied artificial cognitive agents. Based on a standard Remote Control (RC) model car fitted with a wireless camera, the system allows a teacher to demonstrate the desired driving behaviour by viewing the images from the camera on a PC monitor and using a standard computer game steering wheel and foot pedal controller to navigate the car.[2]

The training process involves the teacher driving the agent in order to follow a lead vehicle. This collects a sequence of pairs of images and control parameters that implicitly capture the desired behaviour.

2 Action space clustering

Unsupervised learning techniques are often applied to percept spaces (e.g. image or feature space), but are prone to yielding ambiguous or erroneous results. This is often

[2] Details of robotic platform and collected data sets and code available here www.cvl.isy.liu.se/research/embodied-vehicle-navigation

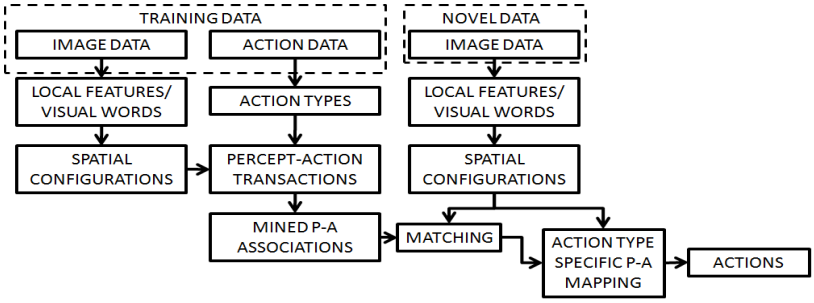


Fig. 2: *System overview*: Coupled percept and action data are represented as Percept-Action (P-A) transaction vectors by concatenating visual codeword configuration vectors and action-type labels. Data mining is then used to discover P-A associations that identify feature configurations that are associated to a particular action-type. Matching these association rules in training images then provides data for learning P-A mappings for each association rule, that map from feature configurations to actions. Matching the association rules in novel images then activates the associated P-A mappings, thus providing a mechanism for generating appropriate responses to novel image data.

due to assumptions about suitable distance metrics used to cluster the data. In general, action data (e.g. control signals) are of lower dimensionality than percept data, and related points in the action domain are generally more similar than related points in the percept domain [10]. This implies that the action space is more suited to unsupervised learning techniques. These observations lead to the proposition that the action space should drive the organisation of the percept space. This idea is strongly related to embodiment, and the Embodied Mind theory [1] [3].

For an embodied agent (e.g. all natural cognitive systems and the system proposed in this work), percept data is never obtained in isolation - it is always coupled to action data. This coupling is exploited in this work by clustering coupled percept-action exemplars, in the action space. This results in the formation of meaningful classes of action or 'action-types', as well as meaningful perceptual groups. The action data, $\{\mathbf{a}^1 ... \mathbf{a}^N\}$, with $\mathbf{a}^n = [a^n_{turn}, a^n_{speed}] \in \Re^2$, is clustered - using k-means clustering - into $k_{act} = 6$ clusters. Figure 1 illustrates the result of performing this action space clustering and examples of the associated images are shown. In order to obtain invariance to displacement, scale and rotation, the action data is whitened prior to clustering. The data is translated (by the mean sample value), scaled (each dimension by the associated eigen values of the sample covariance matrix) and rotated such that the features have zero mean, unit variance and the data axis coincide with the eigenvectors of the sample covariance matrix.

3 System Overview

An overview of the proposed approach is illustrated in figure 2. First an exemplar set, E, of training data of the form $E = \{(\mathbf{p}^1, \mathbf{a}^1), ..., (\mathbf{p}^N, \mathbf{a}^N)\}$, where $\{\mathbf{p}^1 ... \mathbf{p}^N\}$ is the set of images, and $\{\mathbf{a}^1 ... \mathbf{a}^N\}$ is the set of action vectors, is collected (details of this training

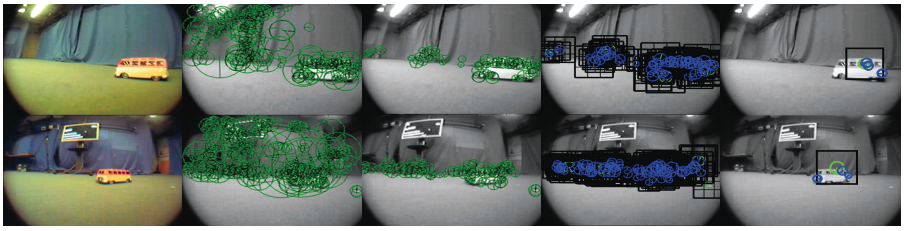
process are given below). Symbolic representations of both the actions, and percepts, are then formed. For the action data, k-means is applied directly to action vectors, resulting in k_{act} action-types, as detailed above. For image data, a visual codebook of SIFT features is built using k-means clustering, where the cluster centers make up the codewords. Spatial relationships between features are represented by encoding local feature configurations, as described in section 4.1. The visual information in each image is thus represented as a set of codeword configurations.

Links between the symbolic percept and action spaces are then obtained by performing data mining on a combined Percept-Action (P-A) representation, named P-A transactions. Each transaction represents an action-type coupled to a codeword configuration, where one item in each transaction represents the action-type, and the remaining items represent a visual codeword feature configuration, as detailed in section 4.2. The data mining algorithm then processes these transactions to produce P-A association rules.

The training data, and the mined association rules are then used to learn action-type specific P-A mappings, as in section 5.1. These mappings map from the continuous (un-quantised) pose of the image features associated to an action-type, onto the continuous action vectors belonging to that action-type. These mappings constitute affordances for the mined perceptual entities.

Still referring to figure 2, when presented with novel image data, the system constructs the visual codeword configurations as before. These configurations are matched to the mined association rules and the P-A mappings associated to the rules are applied to the features that form the matching configurations, in order to generate a response. This process of generating responses to novel image data is detailed in section 5.2.

4 Mining Percept-Action Associations



(a) Input image. (b) Sift descriptors. (c) Sift filtering. (d) Feature configurations. (e) Mined configuration.

Fig. 3: *P-A mining process*: Five stages of the feature mining process are illustrated. Sift descriptors are extracted from the input images. These are then filtered to remove features near the top of the image or that have overly large scales. Feature configurations are then assembled and those configurations that are associated to particular action-type are then discovered through data mining.

The proposed vision system is based on local feature descriptors. A Difference of Gaussian (DoG) detector is used to extract regions and the SIFT descriptor [11] is used to describe the regions. A prior is placed on the scale and location of the SIFT features used in the later stages of the process. This results in a filtering of the set of SIFT descriptors extracted from each image. Figures 3b and 3c illustrate this filtering stage. As the lead vehicle will always remain on the ground plain, and as features on the lead vehicle will have a limited scale in the images, features are rejected that appear too near the top of an image or have overly large scales.

The 128-dimensional SIFT feature descriptors are clustered to form a visual word vocabulary, using k-means clustering. Additionally, the scale and orientation of the features are clustered to form 'scale words' and 'orientation words'. Meaning that each SIFT feature can be described using three discrete labels - descriptor, scale and orientation words - and the continuous horizontal and vertical position. For clustering the descriptor, $k_{desc} = 50$, for scale and orientation, $k_{scale} = 5, k_{orient} = 5$.

4.1 Feature configurations

Figure 4 illustrates the method used to encode the spatial configuration of the extracted SIFT features. A similar scheme was introduced in [5]. For every feature in an image (after filtering) a 3-by-3 grid is placed on the image, centered on the feature, and scaled proportionally to the feature scale. Any neighbouring features that fall into a tile of the grid are encoded as part of that feature configuration, the encoding reflects which tile the feature is in i.e. it's spatial relation, and the visual, scale and orientation words representing the feature. A sparse vector representation is employed for which the non-zero indices encode the configuration and the values store the feature index in the image, so that the continuous feature pose may be recalled for the P-A mappings. The feature configuration vector contains the indices of the non-zero elements of the sparse vector, and is used to represent the visual information in the data mining process.

Examples of feature configurations for two of the training images are shown in figure 5. As can be seen, some of the feature configurations lie on or partially on the target vehicle, whilst many lie on the background. The full set of configurations for an image (as illustrated in figure 3d) will contain considerable redundancy, where each local pairwise spacial relationship will be encoded a number of times within multiple feature configurations.

4.2 Percept-Action transaction database

A Percept-Action (P-A) transaction represents a feature configuration coupled to the associated action-type. The action-type being the cluster label assigned to the action parameters that are associated to the image from which the feature configuration is extracted.

The set of items is $I = \{\alpha_1, ..., \alpha_k, R_1, ...R_l\}$, where $\{\alpha_1, ..., \alpha_k\}$ are the $k = 6$ action-type items and $\{R_1, ...R_l\}$ are the $l = 540$ (9 tiles, 50 visual, 5 orientation and 5 scale words) unique spatial relationships that form the feature configurations. Each transaction vector is the concatenation of the action-type item with the items from the feature configuration vector, as illustrated in figure 6. Therefore each transaction contains a subset of I with one item always drawn from $\{\alpha_1, ..., \alpha_k\}$.

The transaction database $D = \{t_1, t_2, ..., t_m\}$ is assembled, as in figure 6, by collecting together all P-A transactions drawn from all training data, $E = \{(\mathbf{p}^1, \mathbf{a}^1), ..., (\mathbf{p}^N, \mathbf{a}^N)\}$.

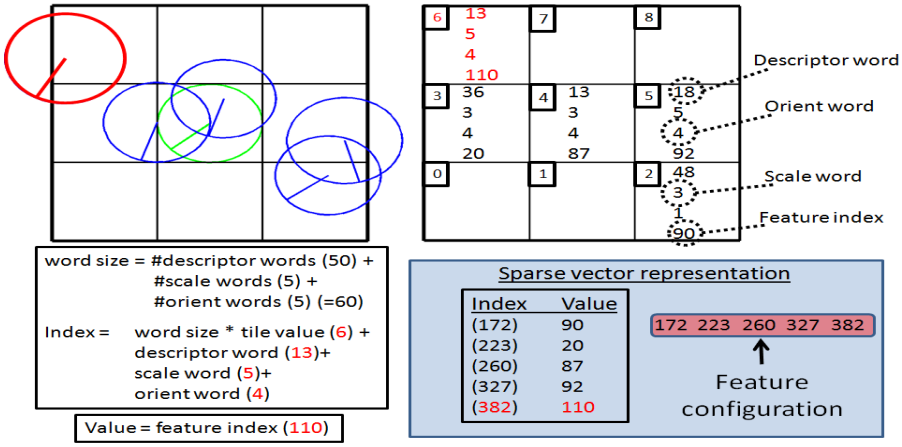


Fig. 4: *Encoding configurations*: This figure illustrates how a configuration of features is encoded in a sparse vector representation, and how this sparse vector representation is used to build the feature configuration vectors used by the mining algorithm. The top left of the figure shows a configuration of features found around the central (green) feature. The top right of the figure illustrates how the feature configuration is represented as a configuration of visual codewords at quantised relative locations, scales and orientations. The bottom left part of the figure details how a particular feature (marked in red in the top left) is encoded in the sparse vector representation. The bottom right of the figure shows the sparse vector representation of the configuration. Also shown is the feature configuration vector that forms the percept part of the transaction vectors used in the data mining. The values of the non-zero indices of the sparse vectors are the feature indices that identify the feature in the image, these are used when mapping from feature pose to action parameters. Note that the center feature (green) is not represented.

In the experiments carried out in section 6, the total number of transactions in the database, $m = 88810$. This database is then processed using the Apriori [9] data mining algorithm, in order to find frequent and discriminative feature configurations for each action-type.

4.3 Mining P-A association rules

Association rule mining is employed to mine the P-A transaction database, in order to discover feature configurations that frequently co-occur with a particular action-type, and not all other action-types. The algorithm finds subsets of items from the transaction vectors that are frequent and discriminative to a given action-type. The Apriori algorithm is run once for each action-type, where it searches for rules including that action-type, and treats all other action-types as negative examples.

For the experiments carried out here, the support threshold $T_{Supp} = 0.02$ and confidence threshold $T_{Conf} = 99$ are used for all action-types and are selected by experimentation. These values are chosen as they provide an appropriate size set of

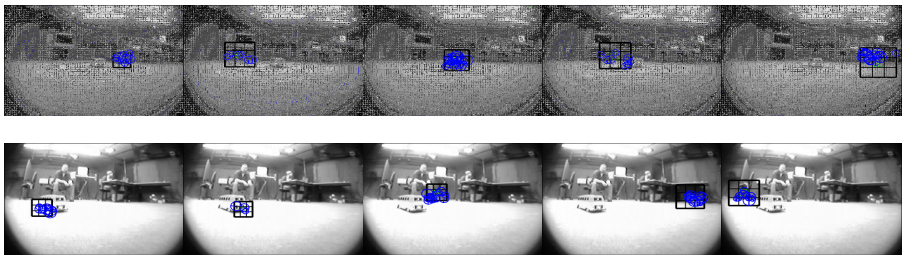


Fig. 5: *Feature configurations*: Five examples of feature configurations for two frames are shown. Some of the configurations contain features on the target, some contain features only from the background.

rules to allow for real time rule matching in novel images (as detailed below in section 5.2). Between 400 and 500 rules are found for each action-type. The rules contain between 3 and 10 items (including the action-type item). An example of such a rule would be $\{\textit{slow-left} \rightarrow 114, 188, 295\}$, meaning that a particular configuration of three features has been associated with actions of the type 'slow-left'.

For the mining, the feature configurations are represented using the indices of the non-negative elements of the sparse vector representation, as illustrated in figures 4 and 6. However, when matching configurations found in an image to association rules, the sparse vector representation is used. The dot product is used to efficiently match rules to configurations found in an image. Examples of the mined association rules for each action-type are illustrated in figure 7.

5 Affordance Based Representation

This section details how the proposed system builds an affordance based representation of the world, and how this representation is used to generate responses to novel percept data. This is achieved by attaching learnt mappings to each mined association rule. These map from the pose (horizontal and vertical position, scale and orientation) of the features in rules onto actions. Linear regression is used to learn linear mappings from pose space to action space.

5.1 Learning action-type specific P-A mappings

A linear percept-to-action (P-A) mapping, $\mathbf{H}_{P-A}$, is learnt for each association rule (mined configuration). $\mathbf{H}_{P-A}$ maps from $(C * 4)$-dimensional feature pose space, to 2-dimensional action space, $\Re^{C*4} \rightarrow \Re^2$, where C is the number of features that make up the rule. A bias term is included in the linear model. An action, $\mathbf{a}$, is computed from a $(C * 4)$-dimensional pose vector, $\hat{\mathbf{p}}$, as in equation 2.

$$\mathbf{a} = \mathbf{H}_{P-A}\hat{\mathbf{p}} + b \tag{2}$$

In order to learn each $\mathbf{H}_{P-A}$, N training examples of $\{\mathbf{a}_i, \hat{\mathbf{p}}_i\}$ pairs, $(i \in [1, N])$ are required. The training set for each $\mathbf{H}_{P-A}$ is obtained by matching rules to configurations found in the training images. Whenever a configuration found in a training

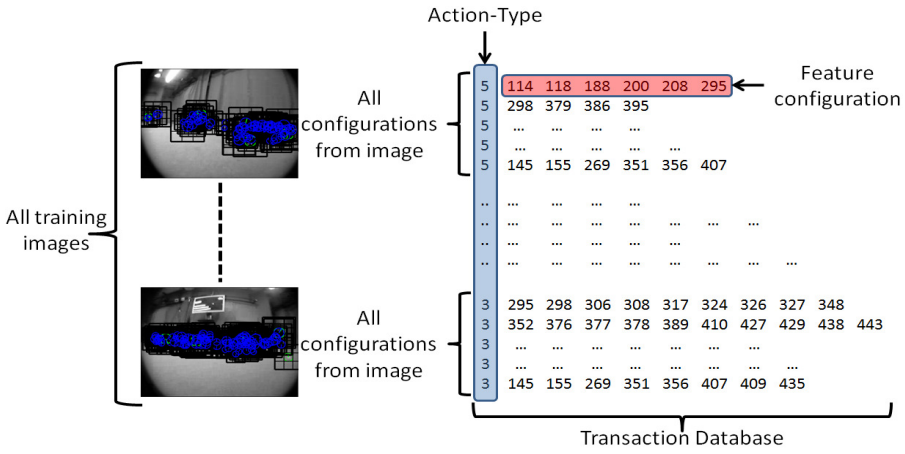


Fig. 6: *Transaction database*: Each transaction is the concatenation of an action-type label (obtained by k-means clustering the action parameters) with a feature configuration (the indices of the non-zero elements of the sparse vector representation). The transaction database is the collection of all transactions from all training images.

image is matched to a rule, the pose parameters of the features that make up that configuration form a new pose vector $\hat{\mathbf{p}}$. The value of the non-negative elements of the sparse vector provide the index to the matched configurations constituent features.

For each rule, all the matched configuration pose vectors, $\hat{\mathbf{p}}$, and the associated action vectors, $\mathbf{a}$, are stacked into the training matrices, $\mathbf{P}$ and $\mathbf{A}$ respectively. To learn the bias for the linear model an additional column of 1s is added to the end of $\mathbf{P}$, giving: $\mathbf{P}' = (\mathbf{P}, [1])$, where $[1]$ denotes a column vector of N rows. Using least squares, $\mathbf{H}_{P-A}$ can now be obtained as follows:

$$\mathbf{H}_{P-A} = \mathbf{A}\mathbf{P}'^{+} = \mathbf{A}\mathbf{P}'^{T}(\mathbf{P}'\mathbf{P}'^{T})^{-1} \tag{3}$$

Where $\mathbf{P}'^{+}$ is the pseudo inverse of $\mathbf{P}'$.

5.2 Responding to novel data

A new input image is processed to generate a set of visual codeword feature configurations as detailed above. Configurations are then compared to all the mined action-type specific configurations (rules). Matching a configuration to a mined rule is achieved by computing the dot product of the two sparse vector representations. If the number of non-zero elements in the dot product is equal to the number of non-zero elements in the sparse vector representation of the association rule, then the rule is matched. If a match is found then an action prediction is made as in equation 2 using the $\mathbf{H}_{P-A}$ associated to the matched rule. Once all found configurations have been compared to all rules, the output action is computed as the median of all action predictions.

To speed up the generation of actions, only configurations within a search range of the previous target location are compared to the rules. The search range is proportional

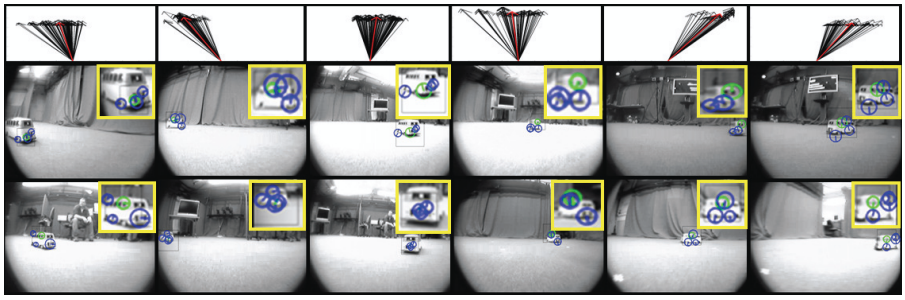


Fig. 7: *Association rules*: Training vectors for six action-types (from left to right on top row: 'slow-left', 'fast-left', 'slow-straight', 'fast-straight', 'fast-right', 'slow-right') are shown along with examples of associated configuration rules mined for each type. In general, if the lead vehicle is to the left/center/right, then the associated action is left/center/right. However sometimes the pose of the lead vehicle, rather than the position is used to associate to the action-type (e.g. far right on middle row, and second from right bottom row).

to median grid size of the configurations matched in the previous frame, and is centered at the median position of the previously matched configurations.

6 Evaluation

The two objectives of this paper - to discover the visual entities important to the task and to generate appropriate responses to novel data - are evaluated. This is achieved by using ground truth data for the target vehicle position. This data is obtained by learning (in a supervised manner) a detector for the lead vehicle. The detector is a Waldboost detector [12] trained on hand labeled examples - sufficient examples are used in training to provide a detector that achieves very high accuracy on the test dataset. The position of the lead vehicle is then used to evaluate how well the mined configurations relate to the lead vehicle. Additionally, the ground truth data is used as input to a supervised method for action generation, to compare to the proposed unsupervised approach.

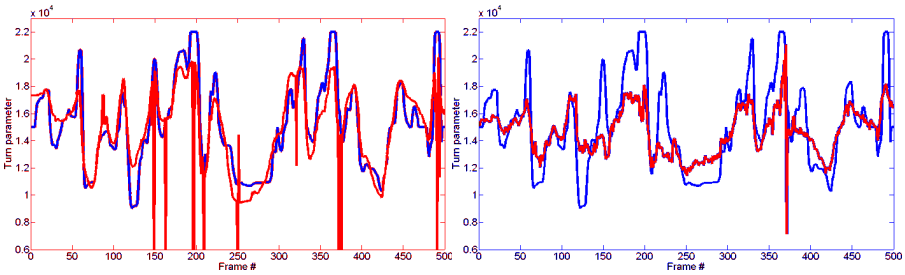
Table 1: Hit/miss ratio for mined configurations lying on the lead vehicle.

Action class	slow-left	fast-left	slow-straight	fast-straight	fast-right	slow-right
Hit/Miss ratio	0.95	0.78	0.83	0.74	0.92	0.87

Figure 7 shows examples of mined configurations that lie on the object of interest, the lead vehicle. Indeed the majority of mined configurations do lie on the lead vehicle, implying that the proposed method has discovered the important visual entities. To quantitatively evaluate this, the hit/miss ratio is measured across a test set of unseen

data. A hit is defined as when at least 50% of the features that make up a configuration lie within the bounding box obtained from the detector. Table 1 shows the hit/miss ratio for each action-type.

The action generation mechanism is evaluated by comparing the actions generated by the system on unseen test data with actions generated by a supervised approach. The supervised approach maps from the ground truth target pose to the action parameters using a single linear regression model, the same as in the proposed approach. In figure 8 it can be seen that the signals generated by both the approaches approximately follow the expected signals.



(a) Supervised method (b) Proposed method

Fig. 8: *Generated action signals*: The generated 'turn-control' action signals (red) are shown for the proposed method and a supervised method, along with the expected action signal (blue).

Comparing the action signals generated by the supervised and proposed (unsupervised) approaches (figure 8), it can be seen that both methods approximately reproduce the control signal provided by the teacher. Note that the high accuracy of the supervised approach in parts of the signal, reflects the strongly linear relationship between target pose and action signals.

The large peaks in the signal generated by the supervised approach correspond to false detections. Although there are false detections (incorrect configuration matches) in the proposed system, these generally have a minimal effect on the output as the output is the median of a number of predictions, therefore these irregularities in the action signals are generally avoided.

Certain parts of the signal generated by the proposed approach do not exactly follow the expected signal (for example from frame 100 to 150). This is in some cases be due to the fact that the expected signal, provided by the teacher, includes instances of oversteer and compensation, and is therefore not necessarily superior to the generated signal.

Figures 9 and 10 demonstrate the approach at imitating the desired behaviour. In figure 9 the target is placed at three stationary positions and the agent is shown to generate actions that drive toward the target. In figure 10 the lead vehicle is driven around and the agent is shown demonstrating the desired behaviour - following the lead vehicle.

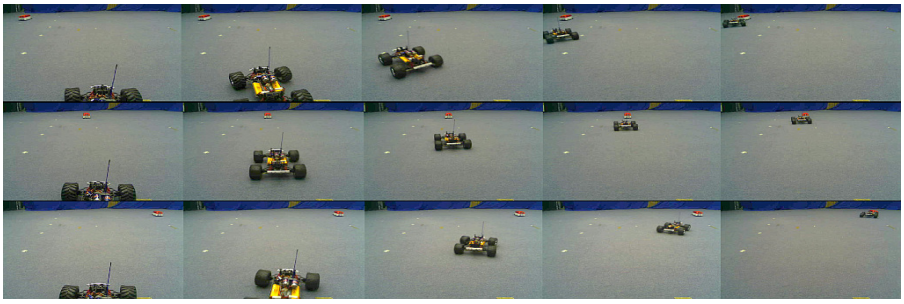


Fig. 9: *Action generation results*: The agent is shown to demonstrate the appropriate actions, by driving (to left - top, straight - middle, to right - bottom) toward the target and then coming to stop.

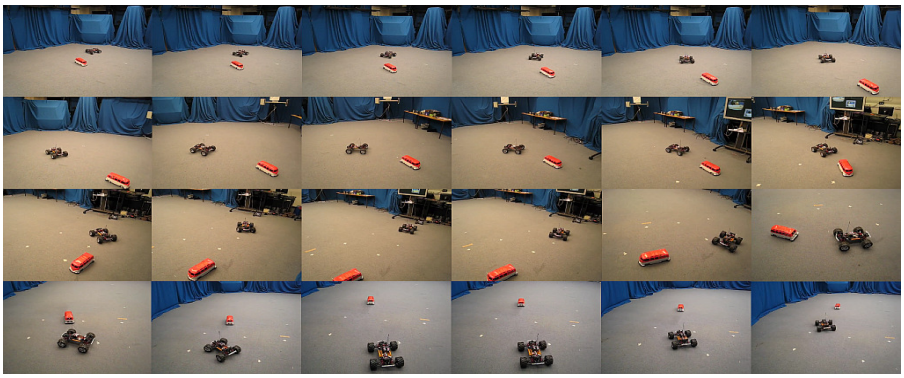


Fig. 10: *Behaviour imitation*: The behaviour demonstrated by example is replicated by the agent, as it follows the lead vehicle.

7 Discussion

This work presents a method for discovering the visual entities that are important to a given autonomous navigation task and utilising these perceptual representations to imitate the behaviour that is demonstrated by the teacher. The system requires no explicit definition of behaviour, uses no prior model of the objects of interest to the task and no supervision, other than the provision of input-output exemplars in the form of images and actions i.e. recorded experiences that exhibit the desired behaviour.

Partitioning the training exemplars using similarity of actions provides a means of organising the perceptual space of the agent in a way that is relevant to the problem domain. This allows for the discovery of perceptual representations that are specific to a particular class of actions. These representations are discovered using efficient association rule mining techniques. The representations are built on a spatially encoded visual word representation. The results shown in figure 7 and table 1 confirm that the visual entities discovered do in fact relate to the object in the scene that is important to the task.

By attaching action generation models (linear percept-to-action mappings) to each discovered visual entity, the system builds an affordance based representation of the world. This novel representation directly couples percepts to actions, resulting in a

system that is able to respond to novel percepts in real time. The results presented in figures 8, 9 and 10 demonstrate that this novel affordance based representation generates the type of actions expected and allows the system to imitate the behaviour demonstrated by the teacher, when presented with new situations. This is achieved with no explicit definition of the behaviour.

Choosing $k_{act} = 6$ ensures that there is sufficient inter and intra class variance of visual information whilst also ensuring sufficient exemplars for learning the visual representations and mappings for each action-type. Larger k_{act} reduces the number of training examples for both the configuration mining and mapping learning. Smaller k_{act} increases within class variation and reduces the discriminative power of the mined configurations. Clearly the selection of k_{act} will impact on the quality of both the mined configurations and the generated actions. Future work will investigate the effect of this parameter on system performance, and investigate the use of mode seeking and other clustering algorithms for action space clustering.

Acknowledgement. This research has received funding from the EC's 7th Framework Programme (FP7/2007-2013), grant agreements 21578 (DIPLECS) and 247947 (GARNICS).

References

1. Lakoff, G., Johnson, M.: Philosophy in the Flesh : The Embodied Mind and Its Challenge to Western Thought. Basic Books (1999)
2. Garbarini, F., Adenzato, M.: At the root of embodied cognition: Cognitive science meets neurophysiology. Brain and Cognition **56** (2004) 100–106
3. Brooks, R.A.: Intelligence without reason. In Myopoulos, J., Reiter, R., eds.: Proceedings of the 12th International Joint Conference on Artificial Intelligence (IJCAI-91), Morgan Kaufmann publishers Inc.: San Mateo, CA, USA (1991) 569–595
4. Gibson, J.J. In: The Theory of Affordances. Lawrence Erlbaum (1977)
5. Efficient Mining of Frequent and Distinctive Feature Configurations. In: Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on. (2007)
6. Gilbert, A., Illingworth, J., Bowden, R.: Fast realistic multi-action recognition using mined dense spatio-temporal features. In: Proc. Int. Conference Computer Vision (ICCV09). (2009)
7. Chum, O., Philbin, J., Sivic, J., Isard, M., Zisserman, A.: Total recall: Automatic query expansion with a generative feature model for object retrieval. In: Proceedings of the 11th International Conference on Computer Vision, Rio de Janeiro, Brazil. (2007)
8. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules in large databases. In: VLDB '94: Proceedings of the 20th International Conference on Very Large Data Bases, San Francisco, CA, USA, Morgan Kaufmann Publishers Inc. (1994) 487–499
9. Borgelt, C.: Efficient implementations of apriori and eclat (2003)
10. Granlund, G.H.: The complexity of vision. Signal Processing **74** (1999) 101–126 Invited paper.
11. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision **60** (2004) 91–110

12. Sochman, J., Matas, J.: Waldboost " learning for time constrained sequential detection. In: CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2, Washington, DC, USA, IEEE Computer Society (2005) 150–156