**Task-Oriented Visual Servo System of Robot Arm for 3D Object based on Automatic Multilayers Networks Learning Approach**

**Ch. 1 Introduction**

Object recognition has been a well-studied topic in computer vision, and also widely applied in automatic industrial production line. Model-based recognition methods are commonly used in industrial application. Model-based methods are famous for efficiency and robustness. The performance is mainly related to the raw data which have to be captured manually. Hence, the system is hard to adapt with the changing of work pieces, manufacturing processing …etc. For 2D object recognition, users need to capture numerous raw data of target objects which are placed in specific poses. These cumbersome works would become much more complex while the methods are expanded to 3D object recognition.

In general industrial purposes, we do not really concern about all detail of entire object. Instead, the relation between input and target is the key point for completing the tasks like pick and place, work pieces arrangement, components insertion…etc. Take pick and place as an example, we would like to manipulate robot arm to pick work pieces on the conveyor and place them in specific pose.

For traditional recognition methods, the trivial and complex woks for building 3D models are inevitable, so there is necessity to develop a new automatic approach. The key point of our research is that we are concerned on the relation between the input and target rather than the entire 3D model. To solve this issue, we proposed an automatic system structure which can automatically infer the relation between input poses and corresponding target pose multilayer networks. Fig. 1 shows the flowchart of our approach. In measurement stage, camera 2 is used to replace manual labeling in other learning approach. Human certainly can give selected instance more precise label, but human input make labels become costly to derive due to labor cost. To build up an automatic system, the camera 2 play a role of human to judge whether the selected instance is correct or not. Labeling result become binary set, and can only use for training binary classifier [Ref.]

The binary classifier can only identify true or false of selected instances, which is far from solving our issue. The relation between inputs and targets has to rely on the extracted features of images. Inspired by human visual system, while people see an unknown object, we would intuitively associate this object with other object in our memory (Data base) which have similar feature or structure. Hence, we further proposed a special descriptor for object model and multilayers networks for learning phase. The descriptor is used to describe an object by the structure of feature points. There are many brilliant approaches with high quality and low computation cost [SLAM, SIFT, SURF, BRISK, FREAK]. All these methods are focus on building scale-invariant descriptors from sparse features. The sparse feature points are only extracted from some strong features in different scale, so the feature points are not uniform distribution on the object. For inferring the relationship, extracted uniform distributed feature would help finding obscure co-features in different viewpoints, so color features are much more suitable in this issue. Unfortunately, low-level features are easily affected by environment, and the uncertainty would reduce the robustness of descriptors.

To conquer the uncertainties of low-level feature, we proposed a descriptor which is constructed based on Markov Logic Network (MLN) [Ref]. MLN is an approach combines first-order logic and probabilistic graphical model. Probabilistic graphical models are famous for handling uncertainty, and flexible structure which can be refined through structure learning process [Structure learning ref.]. First-order logic enables to represent different level knowledge. In this paper, first-order logic not only can describe the structure of an object in different view-point (feature level), but relation between each viewpoints (object level), and the scenario of tasks (task level). These different level formula (or clause) can be quantified and integrated into multilayer networks through MLN. The most well-known multilayer networks are multilayer perceptron (MLP) [Ref.]. MLP is feed forward neural networks with multiple layers neurons. Although MLP had been well studied and enable to apply in wide variety of research field [Ref], MLP can only learn the relation between input and output with same labels as feature to feature, object to object…etc. Hence, we combine the concept of MLP with MLN, and tend to model multilayer networks which can integrate different level’s information for inferring the relationship.

In this paper, we desire to build up a task-oriented approach which can automatic learns the relation between input and target without any prior knowledge. The paper start with briefly overviewing of system designed and structure in section 2. The MLN-based descriptor for recognized object would be revealed in section 3. Section 4 introduces how to model the proposed multilayer networks, and active learning algorithm. Then, we compare the performance of proposed system with several different features and segmentation methods in section 5. Finally, the experimental review and conclusion would be discussed in final section.

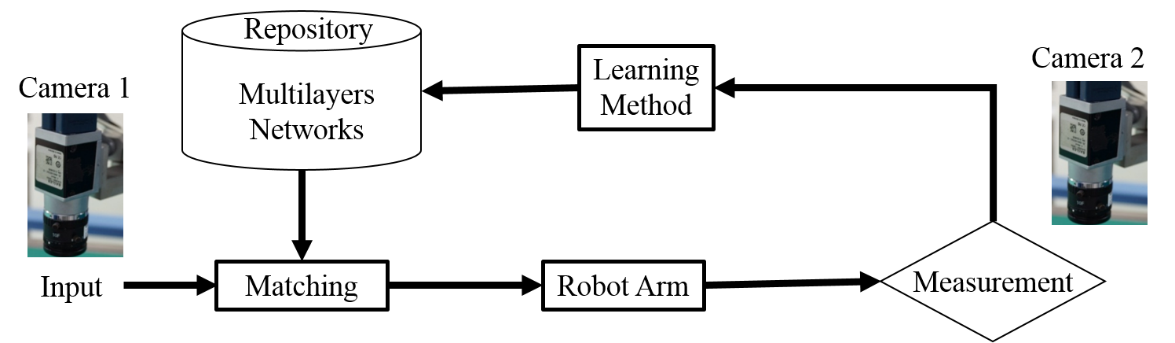


Fig. 1

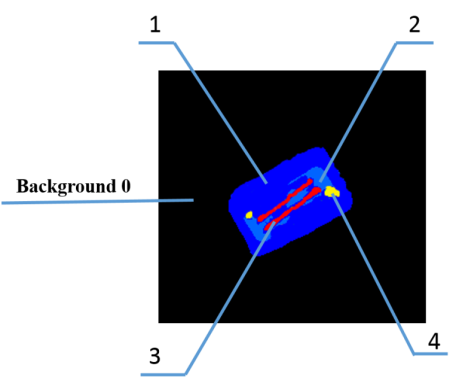
**Ch2. System architecture**

The main purpose of this paper is that desire to automatically derive the orientation relationship between input poses and target poses without 3-D model of objects, and guide robot arm to pick and complete assigned tasks. The target poses of work pieces have to be given by users. Users then only need to put the work pieces on the conveyor arbitrarily, and the relationship would be learned by proposed approach.

This system consists of three main parts: Firstly, camera 1 in fig. 1 capture all input work pieces with arbitrary poses, and classify the features of each single work piece through clustering or segmentation method [Ref.]. For each work piece, camera 1 would capture serial images while work pieces move into FOV as shown in fig. 2. The descriptors of every input work pieces would be constructed by these serial images. Hereafter, system would search matched descriptors from repository, and guide robot arm to pick the work pieces and rotate to target pose. Otherwise, the system would infer the most possible result through proposed multilayer networks, and guide robot arm to pick works pieces based on inferred results.



1. Serial captured frames



(b) Result of background subtracting and clustering

**Fig. 2** Preprocessing of input objects

For the second part of system, camera 2 plays a role as supervisor of learning. The results would be labeled as true or false. The descriptors with labels become the input of multilayer networks. The third part is learning approach of multilayer networks. The labeled data would be added to the networks after one cycle of system. The labels are used to refine the training result of multilayer networks. Based on these three parts, this system can automatically learn the relation model for robot arm to complete tasks without any manual operation.

The variables of our model can be classified into six types:

The relation between each variable is illustrated in fig. 3. is set of captured object 1 to **W** from camera 1. isset of captured serial frames of object n, which is composited by genuine and reference images as shown in fig. 2. is the genuine image, and is no. w reference frames. is constructed by based on proposed 2-D descriptor-constructing method in section 3. Descriptor would be classified to identified object () or unidentified object () based on Most Probable Explanation (**MPE**) [REF.]. Since a 3-D object is composited by multiple 2-D images, one object in our system is also defined by multiple 2-D descriptors. is the set of objects which mapping relation between descriptors and ideal pose had been constructed. is the set of descriptors of recognized object in , and is the set of descriptors of unidentified object in isthe set of rotation anglefor robot arm which represents the relation between input descriptor and target pose in Cartesian space, and can be inferred and trained from the descriptors, so set of descriptors is onto set of rotation angles of robot . The approached would be further discussed in section 4.

is unidentified objects which might be part of , and need more training to merge into . is also consisted by multiple descriptors. While input descriptor is classified into unidentified object as in fig. 3, the system would infer a automatically, and use camera 2 to identify the result. If the result is true, and would be merge into corresponding identified object in .Otherwise, would be wiped out of considerations to avoid the same mistake occur repeatedly.

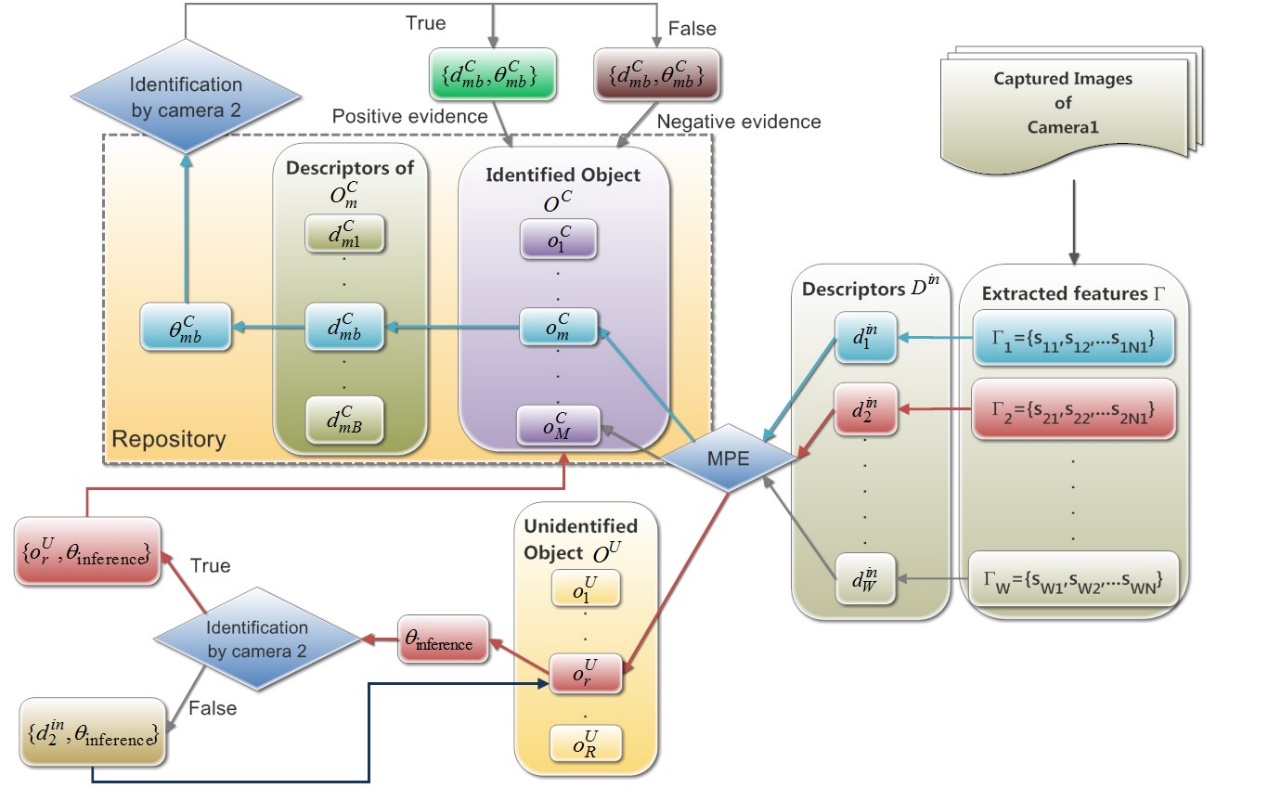


Fig. 3 Relation of variables in multi-layers networks

**Ch. 3 MLN-based Descriptor**

**3.1. The concept of constructing MLN-based descriptor**

Low-level features like RGB, HSV, edges are easily affected by environment. The segmented results of low-level feature might be different even two images of the same object, so the segmented results are hard to match with each other directly. Fig. 2(b) shows the background subtracted result of genuine image of fig. 2(a) by adaptive Gaussian mixture model [Ref.], and the object is clustered according to RGB features. RGB features are classified into 4 parts for each channel, so the max size of feature is 64 in this paper. **N** can be adjusted dependent on selected clustering method. Different classes of feature are labeled different number such as fig. 2(b), and the black part means subtracted background and is label 0. The other label number is between 1 and **N**.

Considering the uncertainties of clustering results of low-level features, probabilistic model is the best choice for conquering uncertainties. Although there are several attributes changed due to environment’s effect (e.g. size of clusters), most of geometric relations between each cluster are relative robustness. We desire to construct the probabilistic-based model in MLN by relations between clusters. Segmented results are used to proposed MLN-based descriptor. The main concept of MLN is that use weighted feature function to soft the hard constrains of first-order logic formulas. If a world violate one formula, it become less possible but not impossible. Take our case as example, when a clustering result of image capture by camera 1 have several clusters different from the result in repository, the MLN-based descriptor would only reduce the probability of candidate rather than wipe out of consideration. Hence, probability-based descriptor has more uncertainty tolerance than the other descriptors which are only modeled by the structure of key feature points.

Markov logic network L is a set of pairs (Fi,wi), where Fi is a feature function of first-order logic and wi is a weighting of correspond formula. The first-order logic formulas are converted to ***clause form*** (also known as ***Conjunctive Normal Form CNF***). Each node in L means one feature of each feature function Fi. The value of Fi is 1 if formula is true and 0 otherwise. The MLN aim to model the joint distribution of a set of variables . to are the serial captured image of object **k** in 2-D scene, and represents genuine images of .We desire to make probability of represent the descriptor of , because genuine image is the most representative image of 2-D face for a 3-D object in a serial images, and less overlap with other faces of the same object.

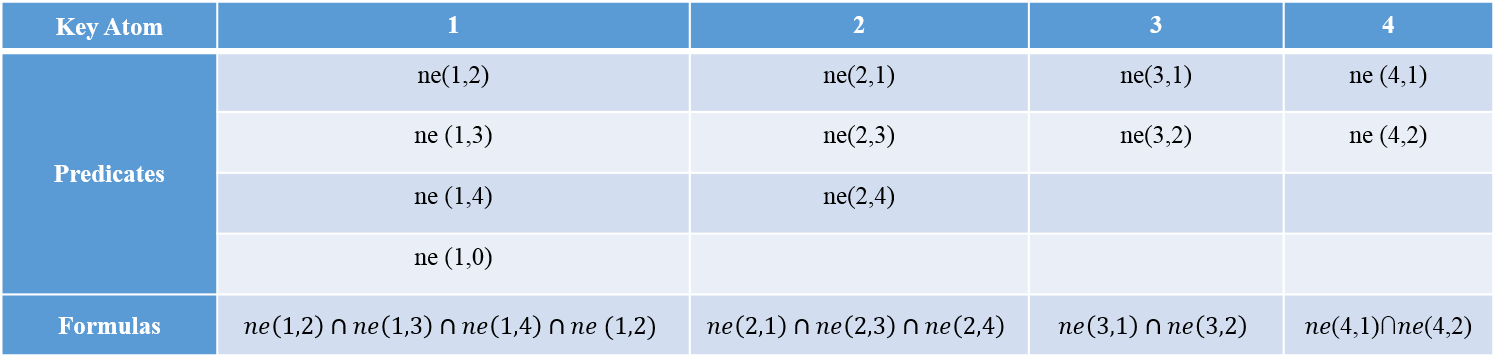
The probability distribution of over possible world specified by MLN is given by:

(1)

Where **F** is the number of formulas in the and is the number of true groundings of Fi in .

The first-logic formulas are consisted by conjunction form of predicates. Predicate ***ne(***sj,sneighbor***)*** is used to represent the conjunctive neighbors of each clusters. We consider cluster as key atom and sample conjunctive neighbors to derive predicates. One center atom would derive one formulas. Therefore, if an object is segmented to **J** kinds of segmented features, its descriptor would be constructed by **J** first-order formulas. Take fig. 2(b) as an example,

Table 1



The complexity of formulas constructing processes depend on the number of different kinds of segmented features in an object. From table 1, it’s seen some equivalent predicates (e.g. **ne(1,2)** and **ne(2,1)**) are repeated sampled. To reduce the complexity, we use dynamic programing algorithm in table 2 to enhance efficiency. The algorithm can define all equivalent predicatesin one iteration, and avoid repeated sampling.

Table 2 Algorithm for constructing MLN\_based descriptor

|  |
| --- |
| **Function** MLN\_Descriptor(*S\_img , S , S’ , S\**)  **Input**: *S\_img*, an image with segmented result  the contour point *p* of *ns*th cluster with label *l\_cs*  **Output**: nneighbor[*n* , *l\_c* , *l\_n*], the neighbor with label *l\_n* of *n*th center with label *l\_c*  Random(Point in *S\_img*)  **repeat**      **Endif**    **Endfor**  **Endfor**  **Endfor**  Random()  **Until**  **Return** *neighbor*[*n* , *l\_c* , *l\_n*] |

**3.2. Inference and Weight learning of MLN-based descriptor**

The weight of MLN-based descriptor is learning by maximizing the pseudo-log-likelihood. Since each descriptor can be considered as a close world, we only need to consider the atoms which derive from captured serial frames. Comparing with uniform sampling approach, maximizing pseudo-log-likelihood is more efficient, because log pseudo-likelihood only need to considered relational data. The pseudo-log-likelihood of eq. (1) could be written as:

is the set of first-order logic formulas of , and is the lth ground truth value of . Instead of sampling entire predicates in , we more concern on the strongest formulas in serial images. For every members of including the same predicates with , the set of formulas which include common predicates is considered as Markov blankets. Fig. 4 demonstrates the construction of Markov blanket. We set formulas in is composted by predicate ***ne4***, ***ne6*** and ***neZ***, so we only sample the other members of which include the formulas which are also composited by ***ne4***, ***ne6*** or ***neZ***.The set of these formulas is considered as . The gradient of the pseudo-log-likelihood can be written as:

]

Where is the number of true grounding of ith formula while set , and similar for . Pseudo-log-likelihood can learn MLN weights efficient while combine with the L-BFGS optimizer [REF.]

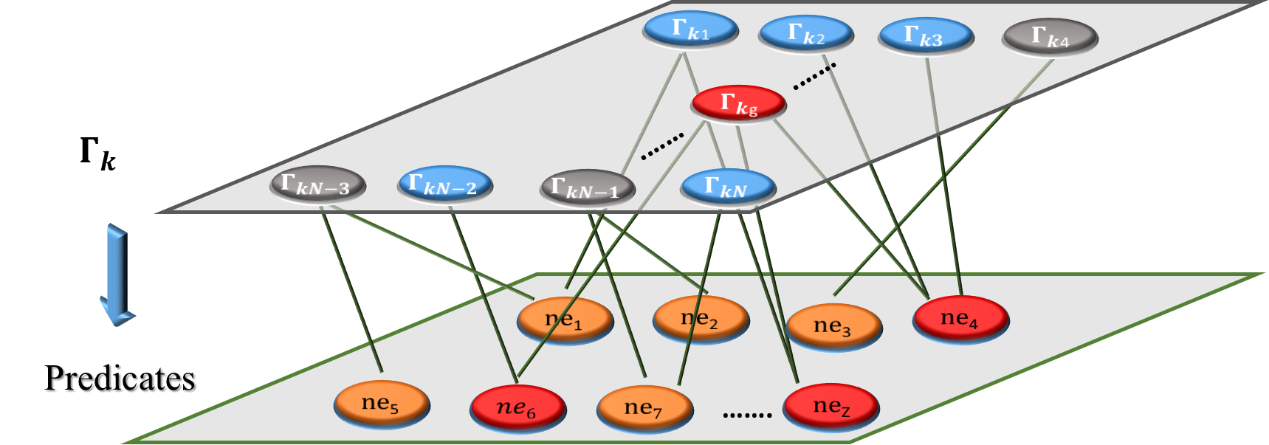


Fig. 4

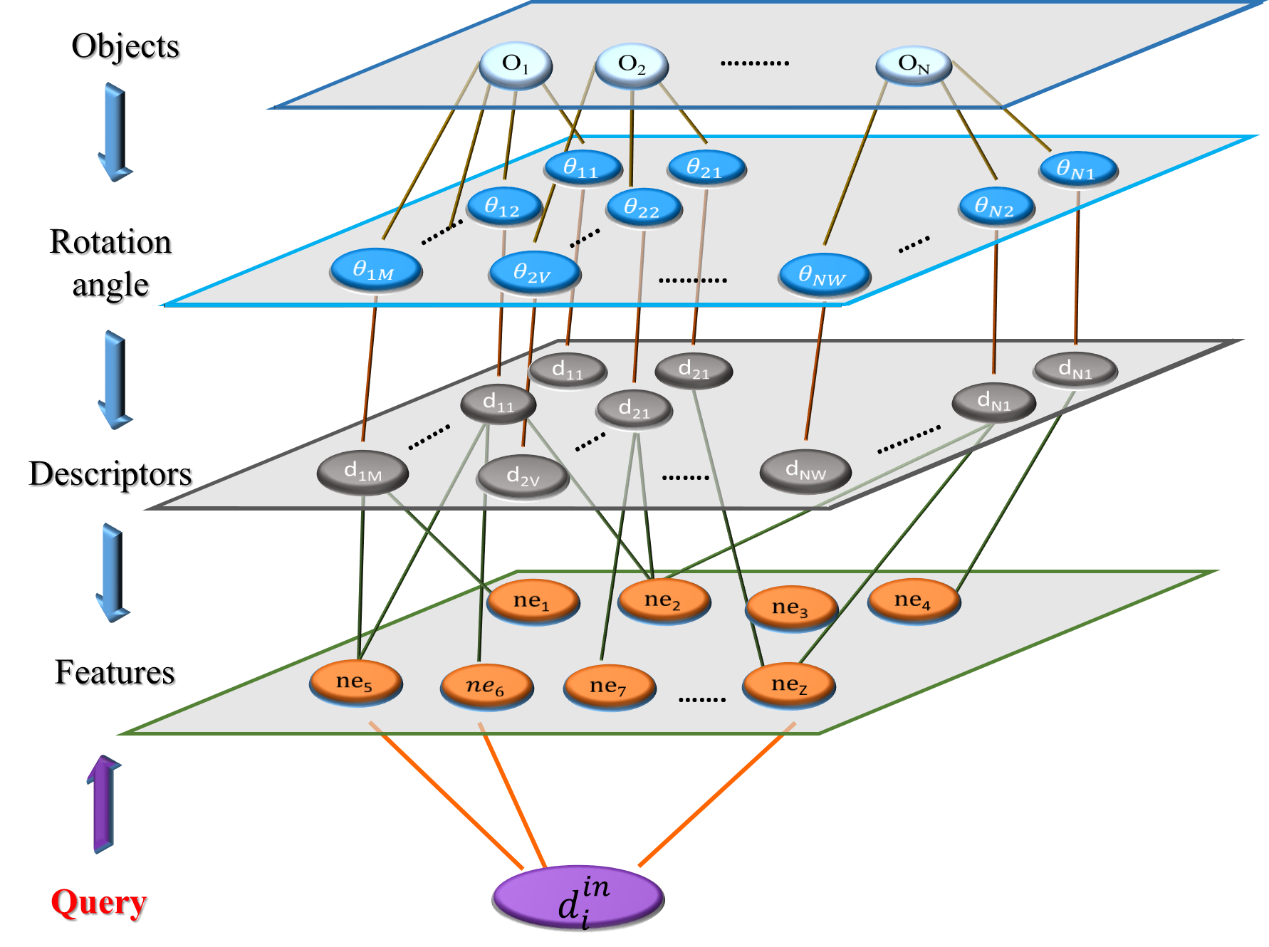


Fig. 5

**3.3. Matching of MLN-based descriptors**

For constructed descriptor , system would search for the matching descriptor in the repository, and further classify to set of identified or unidentified object. The structure of multilayer networks in the repository can be illustrated as fig.5. The objects in the repository are composited by multiple rotation angles and descriptors. A descriptors is the combinations of predicates, we can use the same concept of inference in section 3.2. The descriptors in the repository which have common predicates with query would be considered as evidence, and use maximum likelihood to derive the matching result. The pseudo-log-likelihood of descriptors matching could be formulated as:

represents the set of descriptors which are belong object , and include common predicates with . In our approach, we only consider one object at a time, and the descriptors in the object which have common predicates are the Markov Blanket of , and calculate likelihood function for all possible objects. The input descriptor would be matched according to the result of maximum likelihood. is the Markov Blanket which has maximum likelihood of . If likelihood of a input descriptor is lower than a threshold for every candidates, the descriptor would become a new unidentified object and save in the repository. Reminding that the descriptors in unidentified object class are not abandoned, but need more information to merge into identified object class.

**Ch. 4** **Inference and Learning of Mapping between MLN-based Descriptor and Rotation Angle**

MLN-based descriptors are matched according to the neighborhoods of clusters, so the descriptor is scale and pose invariant. To make robot arm place input objects to corresponding target pose, the relation between descriptor and rotation angle have to be constructed. We assume different faces of same object have to include some common predicates, and these common predicates can be used to infer the relation between input and target pose. Set of rotation angle is composited by ,,andwhich represent roll, pitch and yaw angle of 3-DOF end effector of robot arm. Initially, is unknown due to proposed system without any prior knowledge, so can be only predict by the common predicates between descriptors. For two matched descriptor, the common predicates has significant possibility to be the same parts of object, so the relation between common predicates in Cartesian space can be used to predicate possible rotation angle , and make inferred results more reliable and accurate. We use the mass center of each cluster represent the position of each cluster in Cartesian space, and the center of images is origin of coordinate. Firstly, we sample the center atoms of common predicates between input and target descriptor, and compare the difference of position between each center atom. The difference are represent by vector which is formulated as:

Where is the vector of center atom c. and are the position vector of center atom c of target descriptor and input descriptor. Therefore, for every matching pair, there would be one and only one vector for each center atom of common predicate. According to the MLN-based descriptors, the formula with higher weighting means more reliable. Reminding the example in table 2, every predicates of a formula belong to the same center atom, so the weight of each formulas could be further considered as the reliability of each center atom. Hence, we choice center atom which is included in the formula with the highest weight firstly, and transfer to rotation angle for robot arm. The result would be checked by camera 2. While the number of inputs grows, the results would become the set of vector which is the set of vector of descriptor b in object m. The set of vector is used to build up a Markov network model which could refine the predicating result based on the historical results which are identified by camera 2. Since the uncertainty of probabilistic descriptor that the matched input descriptor for same target descriptor might not be totally same one, we would like to build up a mapping model which can predict ideal result depend on different input. There are two factors have to be concerned: (1). the distribution of historical vector . (2). likelihood of input and target descriptor. Hence, the mapping model could be formulated by joint distribution:

Where is a set of likelihood of input and target descriptor. is partition function of model, and is the likelihood of input and target descriptor in time t. f(\*) is feature function which is 1 while \* is true and 0 otherwise, and is weight of feature function. This model represents the mapping between set of input descriptors and the same target descriptor. While derive new input descriptor, the predicated result can be derived by:

**Ch. 5 Experimental Results**

The proposed system is evaluated by chosen 20 kinds of work piece for computer motherboard inserting task, and implement on 6-DOF robot arm with 3-DOF end effector. Target poses of chosen work pieces are shown on fig. 6, and classified into the three class: WP1, WP2 and WP3. WP1 is the work pieces which is monotonous, and small size, so the features on image are insufficient. Insufficient feature would increase the uncertainty of constructed descriptor even for scale space based approach (e.g. SIFT or SURF), or proposed probabilistic based descriptor. The work pieces in this class are used to prove the ability of improvable of proposed approach. The proposed system can raised the accuracy after several iteration even for poor inputs.

For class WP2, work pieces are similar shape and easy to cause mismatching. Since proposed descriptor is scale and pose invariant, this class is used to exam the ability of recognition for similar objects. Class WP3 is the reference for our experiment. The works pieces are ideal for our system which have sufficient features, distinct shapes, and easy to be identified. The results of WP3 are base line of our experimental results, and the results of WP1 and WP2 would be further discuss based on the base line. The system is realize by two computers: computer 1 take responsible for controlling robot arm, capturing and preprocessing of images from camera 1 and camera 2. Captured images would be background subtracted firstly based on the MOG function in ***OpenCV 2.42*** [Ref], and cluster RGB features by approach on section 3.1. Preprocessed results are transferred to computer 2 by Ethernet socket. Proposed inference and learning approach of multi-layer networks is realized on computer 2 based on ***Ubuntu 12.04*** with open source library ***Alchemy*** ***2.0***[Ref.] and ***Dlib machine learning*** [Ref.]

The inputs of experiments are work pieces which arbitrary separate on FOV of camera 1 with arbitrary poses, and the number of work pieces in each frame is non-constant. Fig. 7 shows the experimental results for each class. Fig. 7 (a) is the result with lighting control which brightness and contrast ratio are ideal for each frame, and fig. 7 (b) is result without lighting control. The accuracy values of these two figures are average value of 20 times repeat experiments, and checked every 50 input pieces. The system is considered convergence while accuracy of matching results are higher than 95%, and stop learning process. Comparing fig. 7(a) and fig. 7(b), the results of WP3 in both two conditions are the first convergent class while number of input work pieces are 200. WP1 is convergent in 300 inputs in both two conditions. WP2 is convergent in 200 inputs with lighting control and convergent in 350 pieces without lighting control. Based on these results, the environment factor doesn’t make significant effect for proposed system.

Furthermore, we are curious about the relation between numbers of different kinds work pieces and system convergent time. Fig. 8 shows the result while put all 20 kinds work pieces in the training set. Intuitively, more work pieces would increase the probability of mismatching, and put off the convergence. According to the result, the system with lighting control is convergent around 500 to 700 inputs, and convergent in 800 to 900 in five tests. The results shows that number of different kinds of work pieces would put off convergent time and the environment control would slightly improve the convergent time, but doesn’t affect the accuracy.

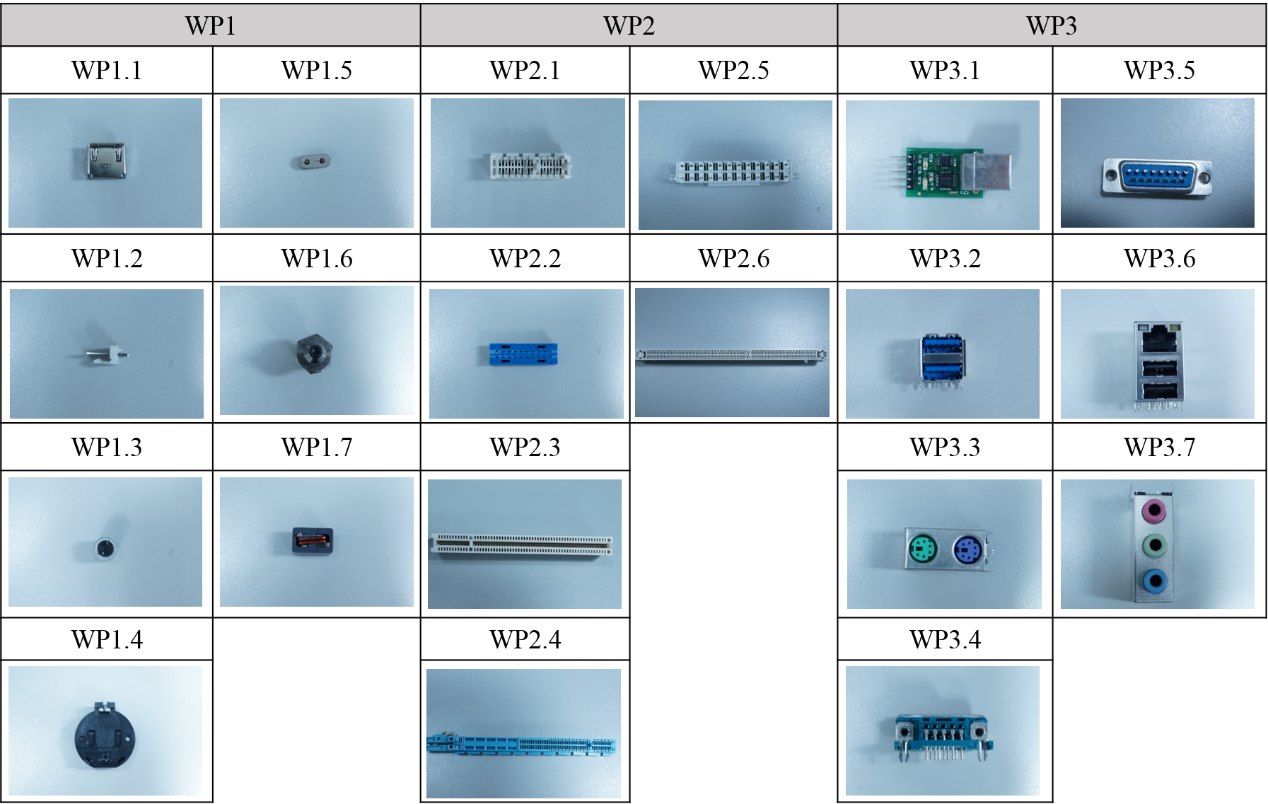


Fig. 6

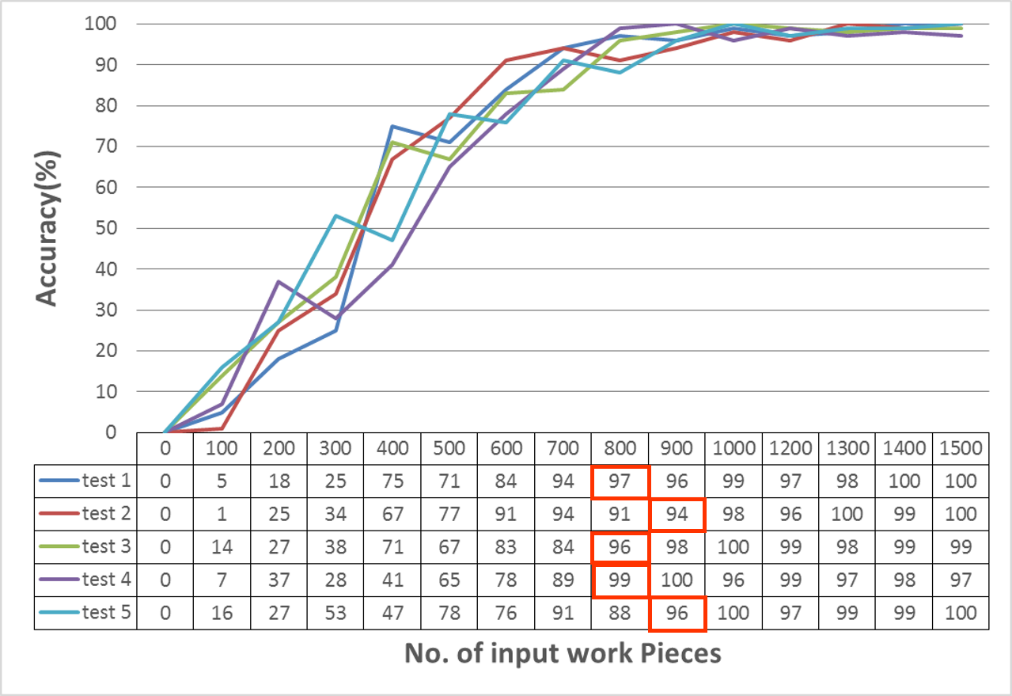
 

(a). (b).

Fig.7



(a).



(b).

Fig. 8

**Ch. 6 Discussions and Conclusions**

The experimental results approve that proposed approaches can conquer the effect of environment by more iterations. The inputs with insufficient feature (WP1) and similar outward (WP2) can be also distinguished and recognized accurately by constructed multilayers networks. MLN-based descriptors are not only easily derived from low-level feature, but can conquer the uncertainty of environment. The probabilistic model is improvable even for unreliable inputs, and optimized system through continuous inputs.

For a task-oriented system, we would like to focus on how to finished the task rather than gather all detail information of system. The key feature of this paper is multilayers architecture which can connect variables in different learning field by networks, and further learn the relation between input and target. The inferred results which had been checked by camera 2 are able to influence all variables in the networks. The MLN-based descriptor and rotation angle can be refined simultaneously through mapping model. The mapping model builds up the relationship between different variables, and re-estimates the registration of each variable in different field based on the captured results of camera 2. Although the experiments in this paper only concern on the result of pick and 3D object, we believe this approach could implement on more different tasks for visual-servo based robot arm like assembling, manufacturing …etc.