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

Ren C. Luo*, and Po-Yu Chuang†,
* †Electrical Engineering Department, National Taiwan University, Taipei, Taiwan 106

***Abstract*—Visual servo systems are widely applied on industrial robot arm recent years. A visual servo system could provide additional perception of robot arm, and make it more robust in different applications. In this paper, we propose an intelligent visual servo system which could automatically learn the model of unknown 3D objects, and self-supervised the results. Unlike traditional vision-based learning approaches, proposed learning system is task-oriented which means learning approaches only concern the relation between input and target rather then each individual part. To construct connections between input and target,the learning approach is established based on multi-layers networks. Multi-layers networks is used to model necessary knowledge in different domains. The experiment results show the feasibility of proposed structure for automatic learning, and multi-layers networks could effectively transfer knowledge between different domains.**

***Index Terms*—Visual servo system, Multi-layers network, Machine learning**

I. Introduction

ROBOT arm with visual servo system had been widely applied in automatic industrial production line recent years[1-4]. Visual servo system provides additional perception of environment, and make robot arm become more adaptive to complete various tasks. For vision system, model-based recognition methods are commonly used in industrial application. The performance is mainly related to the labeled data which is captured manually. Hence, the system is hard to adapt with adding new kinds of work pieces. For 2D object recognition, users need to capture numerous raw data of target objects in specific poses. These cumbersome works lift to much more complex level while the methods are expanded to 3D object recognition. These manual works not only increase the labor cost, but also point out the dilemma of present vision-based robot which is unable to automatically adapt to various assignments.

In general industrial purposes, we do not really concern about all details of entire 3D object. Instead, the relation between input and target is the key point for completing the tasks like pick and place,etc. To solve these issues, we propose an intelligent tasked-oriented visual servo based system which acquires ability of learning relational model of input and target automatically. In task-oriented view, system tends to learn the relation between input and target rather than delicate models for target 3D objects. Therefore, the state problem is that We only provide labeled data which face of 3D objects are desired to be placed on top by robot arm, but the other faces of 3D objects are unknown. The labeled data is considered as target face, and the input faces are arbitrary objects with random faces on top. The input of system is 2-D image data, and output is rotation angle of robot arm in Cartesian space which are different feature domains. Therefore, the traditional single layer model[HMM,SVM, GMM] which end in a linear or kernel classifier is not enough. We introduced a multilayer model to tackle our problems.

The learning of multilayer model achieve dramatically success recent years while deep learning architectures showed up. Hinton et al. [deep learning algorithm] proposed a hierarchical structure which hidden layers are formed by lower level feature to higher level, and had been successfully applied on different research fields[Deep learning application]. Comparing with traditional **ANN** model[Ref. of neural network], the deep learning method is aim to learn the representation of data in different level rather than produces classifiers through features in the same level. Enlightening by deep learning methods, hierarchical structure is applied to our model which is constructed by four layers: **Feature** (image based), **Descriptor**, **Object** and **RotationAngle** (for robot arm).

Through this model, the feature in different domains could be correlated through hierarchical structure, but the system still can not automatically learn the relation between input images and output rotation angle. Being a automatic learning system, the ability which could ”infer” latent edges between labeled and unlabeled data in is needed. Latent edge means two variables in different layers exist an edge in graph model if prior data is sufficient, but, in our case, system only have small amount of prior data. Hence, there are many latent edges which are waiting to be revealed through learning process.

The most challenge part of state problem is that the appearance of different faces of a single object might be huge different, so we design three modules to tackle automatic learning problem. Firstly, we design a probabilistic based image descriptor. Extracting scale- and rotation-invariant sparse feature is a pervasive topic in areas of computer vision. Although many brilliant methods[10-14] provide high quality performance by extracting sparse features, the sparse feature is not compact on inferring the relational model. The sparse feature only model strong features of observed face shows on top, but most of faces is unknown in our cases. we need a descriptor which can provide sufficient information for inferring latent edges, but still retain scale- and rotation-invariant. Proposed probabilistic based descriptor is established based

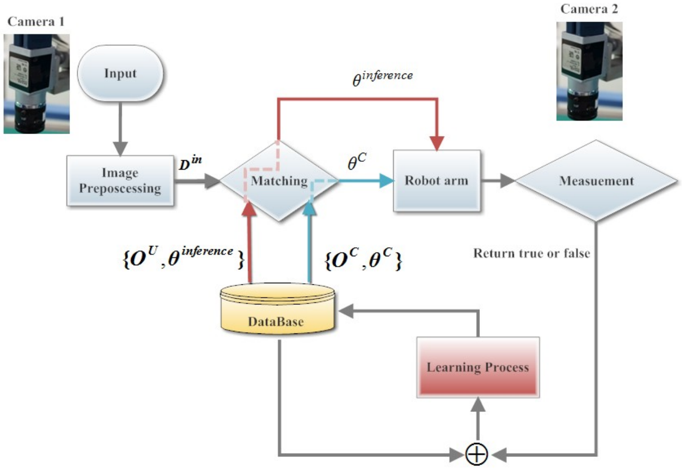


Fig. 1. System architecture

on the Markov Logic Network (MLN) [15]. MLN is an approach combines first-order logic and probabilistic graphical model. First-order logic enable compactly representing the neighborhood of feature points. Probabilistic graphical model can reveal latent edges by proper inference method, and also handle the uncertainty.

Secondly, transfer information module is proposed for constructing latent edge. Transfer information module is realized by Self-taught Clustering algorithm [refer.]. Self-taught Clustering algorithm is a transfer learning method [6-9....need more] which is built for enhancing model through large amount of auxiliary unlabeled data. The input face can be considered auxiliary unlabeled data, and find co-cluster between priors face in the dataset. Hereafter, we further utilize the distribution of co-cluster to infer the possible rotation angle for robot arm, and robot arm would rotate the object from input face to output face by rotating inferred angle. Finally, the validation module is an eye-to-hand camera which used to validate the error between the output face and desired target face. Hereafter, the validation module feedback the error to the model to refined the model. Through these three module, proposed system can automatically learn the relation between input image and output rotation angle with only labeled the target face of each object.

In this paper, we start with briefly overview of system design and structure in section 2. The MLN-based descriptor for recognized object is described in section 3. Section 4 introduces how to model the proposed multilayer networks, build up and refine the structure of database for matching. Then, we compare the performance of proposed system with several different features and descriptors in section 5. Finally, the performance review and conclusion are presented in final section.

II. System architecture

The main purpose of this system is that desire to automatically derive the relationship between input face and target face of 3D assigned objects. The only prior knowledge are the target face. Input is arbitrary assigned object with random face on top, so it is very likely a unknown face of assigned object rather than prior target face. Therefore, system has to infer the correlation between input and existed prior target face in the pool. Proposed system is shown in Fig. 1. camera 1 captures images of all input objects with random faces on top, and construct MLN-based descriptor for each input. Then, system match the input with data in Database and output rotation angle for robot arm. After robot arm placing a input object, camera 2 would validate result, and feedback error for refining existed model. The system architecture in Fig. 1 is realized by a hierarchical-deep model in Fig. 3. The variables in the same layer are independent, and vertical adjacent two layers are full connected. The difference between classic **Deep Belief Networks**(**DBN**)[Ref.] is that proposed model exist two parallel parts as shown in Fig. 3. **Classified Descriptor**($\mathbf{D^C}$)**-Rotation angle**($\mathbf{\Theta^C}$) and **Unclassified Descriptor**($\mathbf{D^U}$)**-Inferred Rotation angle**($\mathbf{\Theta^U}$) have no connection between each other, but both have full connection with deepest layer **Object**($\mathbf{O^C}$) and first layer **Feature**($\mathbf{\Gamma^C}$). To handle tons of unknown data, the structure of connection will dynamically change with observed evidences. Sparse coding method is used to constructs edge in the model, most of connection is zero which is called latent edge in this paper. Latent edge might become non-zero while some new evidence has been discovered. For variable d^C_w in layer $\mathbf{D^C}$, the sparse coding result should be formulated as:

$$d^C_w = \sum_{i \in d^C_w} a_i \Gamma_i + \sum_{j \notin d^C_w} b_j \Gamma_j \tag{1}$$

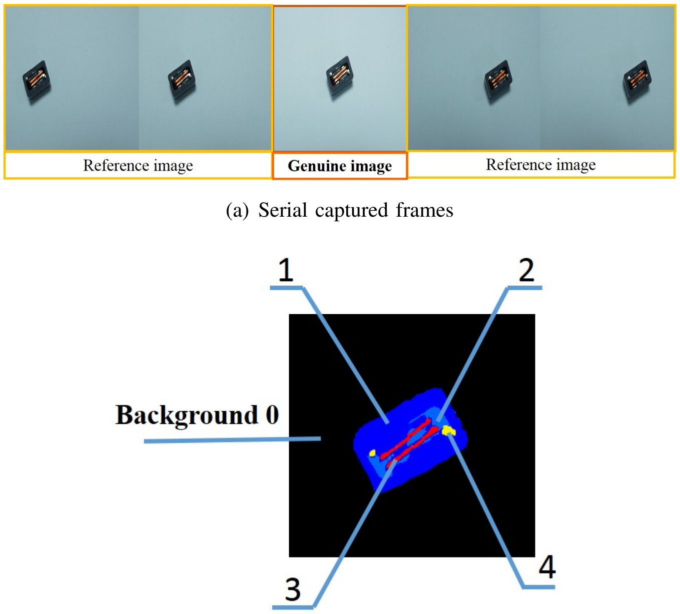
Although Eq.(1) can handle the problem of latent edge, it's impractical to sample all possible conditions whenever new evidence showing up. Therefore, proposed model separate descriptor layer into two parallel parts as:

$$d^C_w = \sum_{i \in d^C_w} a_i \Gamma_i \tag{2}$$

$$d^U_r = \sum_{j \in d^U_r} b_j \Gamma_j \tag{3}$$

d^C_w is considered a prior descriptor which the edge between $\mathbf{\Theta^C}$ and $\mathbf{O^C}$ had been established. Therefore, the left part of parallel layers can be considered as static model until there is a query classified to the $\mathbf{D^U}$. $\mathbf{D^U}$ layer is the set of descriptors which we haven't known that these descriptors are corespondent to which object. Therefore, we propose a inference method to infer the possible rotation angle, and camera 2 will checks the results. If inference success, variable d^U_r and $\theta^{inference}$ become new variables of **Classified Descriptor**($\mathbf{D^C}$)**-Rotation angle**($\mathbf{\Theta^C}$). Meanwhile, new edges of left part of parallel layers are established which can be considered latent edges become visible. By transfer variable between two parallel parts, variables in $\mathbf{D^C}$ will dynamically grow with number of inputs.

In Fig. 3, the extracted MLN-based descriptor of input d^in is derived by **MLN-based descriptor module**. Camera 1 captures serial images while input moves into FOV as shown in Fig. 2(a). First-logic formulas for MLN are derived from these



(a) Serial captured frames

(b) Result of background subtracting and clustering

Fig. 2. Preprocessing of input objects

images and constructed a MLN as image descriptor of the input. Variables of feature layer is a set of first-logic formulas which might be extending following the new feature showing up

from serial images which are captured by camera 1 while input moves into FOV as shown in Fig. 2(a).

This system is realized by three modules: Firstly, **Constructing MLN-based descriptor:**

Hereafter, the constructed descriptor becomes the input of **Transfer information module**. System search for matched descriptors in database ($\mathbf{D^C}$) by pseudo-log-likelihood. If the input doesn't match any descriptor in the data set, we apply Kullback-Leibler divergence(KL divergence)[Cover Thomas, 1991] to find min divergence and co-clusters between input d^in and $\mathbf{D^C}$. While d^C have min divergence with d^in, we assume d^in and d^C belong to the same object O^C. System will infer rotation angle ($\theta^{inference}$) based on the relation between co-cluster in Cartesian space, and guide robot arm to rotate input object by inferred angle.

Validation module is utilized to valid inferred result, and estimate error between desired result and observed result. If the observed result is just matched desired one, the latent edge between d^in, $\theta_{inference}$ and O^C will be established. Consequently, d^in is one of face belonging object O^C and target face d^C_t can be find by rotate d^in with angle $\theta_{inference}$. This method will be explain particularly in section XX.

The data transferring of data in three modules can be represented by hierarchical model shown in Fig. 3. variables of the model are represented by six types:

$$\{d^{in}, \mathbf{\Gamma}, \mathbf{D^C}, \mathbf{D^U}, \mathbf{O^C}, \mathbf{\Theta^C}\ \mathbf{\Theta^{inference}}\}$$

The relation between each variable is illustrated in fig. 3.

$\mathbf{D^{in}}$ is input descirptor constructed by $\mathbf{\Gamma}$ based on MLN model which is captured from camera 1. Γ is a set of image features 1 to K. Descriptor d^{in}_w would be classified to classified object($\mathbf{O^C}$) or unclassified object ($\mathbf{O^U}$) based on pseudo-log-likelihood. In fig.3, If d^{in}_1 is delivered to identified object set $\mathbf{O^C}$,

the system would further assign d^{in}_1 to the most possible belonging object $\mathbf{O^C_m}$. Since a 3-D is composed by multiple 2-D images, one object in our system is defined by multiple 2-D descriptors. $\mathbf{O^C_m}$ is a set of 2-D descriptors of object m. d^{in}_1 is considered as new input data of most possible descriptor couple d^C_{mb} in fig.3. Each descriptor couple includes the rotation angle information between input pose (based on 2-D descriptor) and target pose. θ^C_{mb} is the rotation angle which can make robot arm rotate object m from input to target pose. The result would be checked by camera 2. If it is true, the result is considered as positive evidence, and merged to the learning process. Otherwise, the result is negative evidence.

Similarly, d^{in}_2 is assigned to the unidentified set $\mathbf{O^U}$. Unidentified set $\mathbf{O^U}$ is a set of descriptors which hadn't identified the rotation angle of target prior pose. For unidentified $\mathbf{O^U}$, the system would infer the possible rotation angle $\theta_{inference}$ for input descriptor. The result would be supervised by camera 2. If result is true, the descriptor would be delivered to the identified object set $\mathbf{O^C}$ as a new discovered face of corresponding object. Otherwise, the descriptor would feedback to unidentified object set $\mathbf{O^U}$, and record the inference result to avoid the same fail predication. By doing so, the system is ensured to be converged, and all objects could be identified if the input is sufficient enough.

III. MLN-based Descriptor

A. The concept of constructing MLN-based descriptor

Being an automatically system, deriving more valuable information from raw data could help system deriving exact result, and adapt to numerous suspected input. Most of present image descriptors [10-14] are constructed based on strong extracted feature, because strong sparse feature points are consistent even in different environment. Although these kind of descriptors could efficiently and precisely match given object, the descriptors could not suit for cases which need to infer the relation between existing and unknown data. The sparse feature based descriptor would purge weak feature points of input face, but purged points might be valuable for inferring unknown input from existing data. Hence, the descriptor not only need to be robustness, but also provide sufficient information for constructing relationship. The normal distributed features are compact with our requirement which could provide all detail information of input image. Unfortunately, the normal distributed features such as RGB, HSV, edges, etc. are easily affected by environment. The segmented or clustering results might be different even two images belonging the same object, so the segmented results are hard to match with each other directly.

Considering the uncertainties of clustering results of normal distributed features, probabilistic model is the best choice for conquering uncertainties. Although there are several attributes

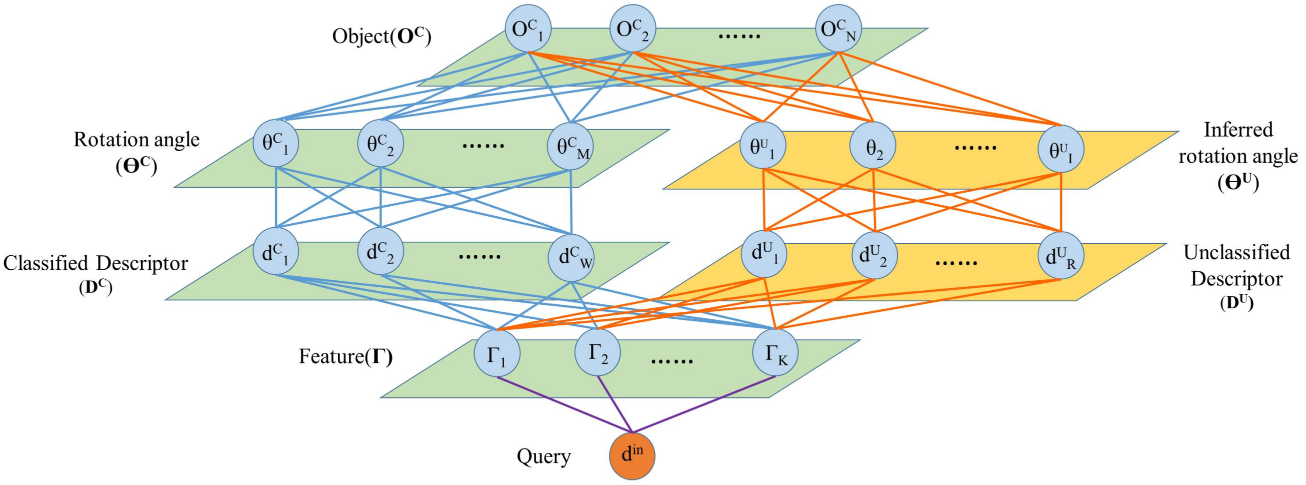


Fig. 3. Hierarchical-deep model for self-taught system

changed due to environment 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 modelled by the structure of key feature points.

Markov logic network L is a set of pairs (F_i, W_i), where F_i is a feature function of first-order logic and W_i is weighting of corresponding formula. The first-order logic formulas converted to **clauseform** (also known as **Conjuctive Nornal Form CNF**) Each node in L means one feature of each feature function F_i. The value of F_i is 1 if formula is true and 0 otherwise. The MLN aim to model the joint distribution of a set of variables $\mathbf{\Gamma_w} = \{\mathbf{\Gamma_{wg}}, \mathbf{\Gamma_{w1}}, \mathbf{\Gamma_{w2}}, \ldots, \mathbf{\Gamma_{wN_w}}\}$ in fig.3. The genuine image Γ_{wg} is the image which center of object is closest to center of camera, so genuine image could be considered as most representative 2-D face of a 3-D object in serial frames. The probability distribution of $\Gamma_w g$ over possible world Γ_w specified by MLN is given by:

$$P(\mathbf{D^{in}} = \mathbf{d^{in}_g}) = \mathbf{P}(\mathbf{\Gamma_w} = \mathbf{\Gamma_{wg}}) = \frac{\mathbf{1}}{\mathbf{Z}}\mathbf{exp}(\sum_{\mathbf{i=1}}^{\mathbf{F}} \mathbf{w_i n_i}(\mathbf{\Gamma_{wg}}))$$

$$\mathbf{Z} = \sum_{\mathbf{\Gamma_{wn}} \in \mathbf{\Gamma_w}} \mathbf{exp}(\sum_{\mathbf{i=1}} \mathbf{w_i n_i}(\mathbf{\Gamma_{wn}})) \qquad (4)$$

Where $\mathbf{F}$ is number of formulas in Γ_{wn} and $n_i(\Gamma_w g)$ is number of grounding of true grounding of $\mathbf{F_i}$ in Γ_{wg}.

For proposed MLN-based descriptor, the first order logic are consisted by conjunction form of predicates. Predicate $ne(s_j, s_{neighbor})$ is used to represent the conjunctive neighbours of each clusters. We consider cluster s_j as key atom and sample conjunctive neighbour $s_{neighbour}$ to derive predicate. Each center atom acquire one formula. Therefore, if an object is segmented to **J** kinds of segmented features, it descriptor would would be constructed by **J** firs-order formulas.

Table I. Example of predicates and first-order logic formulas

Key Atom	1	2	3	4
Predicates	ne(1,2)	ne(2,1)	ne(3,1)	ne(4,1)
	ne(1,3)	ne(2,3)	ne(3,2)	ne(4,2)
	ne(1,4)	ne(2,4)		
	ne(1,0)			
Formulas	ne(1,2)∩ne(1,3) ∩ne(1,4)∩ne(1,0)	ne(2,1)∩ne(2,3) ∩ne(2,4)	ne(3,1) ∩ne(3,2)	ne(4,1) ∩ne(4,2)

Based on this concept, the MLN-based descriptor could be constructed by following process: The background of input image are subtracted through MOG [21](supported by open source library OpenCV), and each isolated object is clustered according to RGB features. RGB features are classified into 4 parts for each channel through K-means clustering [22], so the max size N of feature S is 64 in this paper. N can be adjusted dependent on selected clustering method. Fig. 2(b) shows the background subtracted result of genuine image of fig. 2(a), and different classes of feature are labelled different number such as fig. 2(b). The black part means subtracted background and is label 0. The other label number is between 1 and N. Then, taking fig. 2(b) as example, the object is segmented to 4 kinds of features, and predicates and first-logic formulas are shown in table 1.

The complexity of formulas constructing process depend on the number of different kinds of segmented. From table 1, there are some equivalent predicates (e.g. $ne(1, 2)$ and $ne(2, 1)$) which might be repeated sampled. To reduce the complexity, we use dynamic programming algorithm in algorithm 1 to enhance efficiency. The algorithm can define all equivalent predicates in one iteration, and avoid repeated sampling.

Algorithm 1 Algorithm for sampling neighbours of key atoms

```
Function Sampling($S\_img, S, S', S^*$)
Input :
$S\_img$, segmented input image
$S[ns, p, l]$, the contour point $p$ of $ns^{th}$ cluster with label $l$
$S'$, contour pixel of each cluster
$S^*[ns, p^*, l^*]$, neighbour pixel $p^*$ of $S.p$ with label $l^*$
Output :
$neighbour[n, ln, la]$, the neighbour with label $ln$ of $n^{th}$ key atom with label $la$
Random(Point in $S\_img$)
for k← 1 to number of cluster do
    $n \leftarrow k$
    $S' \leftarrow \emptyset$
    for i←1 to size of contour do
        $S' \leftarrow$ contour pixel of label $ln$
        $S^*.p^* \leftarrow$ neighbour of $S'$ with different label
        if $S^*.label$ is changed then
            $n \leftarrow n + 1$
            $neighbour[n, ln, la] \leftarrow$
                    $neighbour[n, S^*.l^*, S.l]$
        end if
        for $j \leftarrow$ $k$ to $n - 1$ do
            $neighbour[j, ln, la] \leftarrow$
                    $neighbour[j, ln, S^*.l^*]$
        end for
    end for
    $S[ns, p, l] \leftarrow S[k, S', S.l]$
    $S^* \leftarrow S^* - S$
end for

Random ($S^*$)
Until $S^* = \emptyset$
Return $neighbour[n, ln, la]$
```

B. Inference and Weight learning of MLN-based descriptor

The weights of MLN-based descriptor is learned by maximizing the pseudo-log-likelihood. Since each descriptor can be considered as a closed 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 pseudo-log likelihood only need to considered relational data. The pseudo-log - likelihood of eq.(1) can be written as:

$$\log P_w^*(\mathbf{\Gamma_k} = \Gamma_{kg}) = \sum_{i=1}^{L} \log P_w(\mathbf{F_{kg}} = f_{kgl} | \mathbf{MB_{\Gamma_k}(f_{kgl})}) \tag{5}$$

$\mathbf{F_{kg}}$ is a set if first-order logic formulas Γ_{kg}, and f_{kgl} is l^{th} ground truth value of $\mathbf{F_{kg}}$. Instead sampling all predicates $\mathbf{\Gamma_k}$, the strongest formulas in serial images should be more concerned. For every members of $\mathbf{\Gamma_k}$ including the same predicates with f_{fgl}, the the set of formulas which include common predicates is considered as Markov blankets $\mathbf{MB_{\Gamma_k}(f_{kgl})}$. Fig.4 demonstrates the construction of Markov blanket. We set formula f_{kgl} in Γ_kg is composted by predicate $\mathbf{ne_4}$,$\mathbf{ne_6}$ and $\mathbf{ne_Z}$, so, based on the concept, sampling approach only need to sample the other members of $\mathbf{\Gamma_k}$ which are also composted by $\mathbf{ne_4}$,$\mathbf{ne_6}$ and $\mathbf{ne_Z}$. The set of sampled formula is considered as $\mathbf{MB_{\Gamma_k}(f_{kgl})}$. Hence, in the case fig.4, Γ_{k1},Γ_{k2}, Γ_{k3},Γ_{kN-2}, and Γ_{kN} would be sampled.

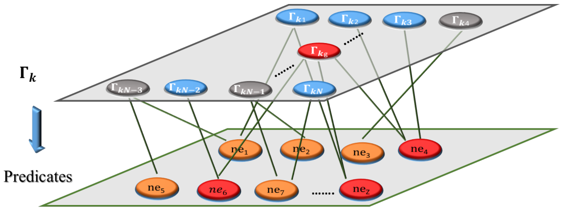


Fig. 4. Example of sampling Markov blanket

Hereafter, the MLN weights are learned generatively by maximizing the pseudo-log-likelihood of Markov blanket. The gradient of the pseudo-log-likelihood with respect to the weights is:

$$\frac{\partial}{\partial w_i} \log P_w^*(\mathbf{\Gamma_k} = \Gamma_{kg}) =$$

$$\sum_{l=1}^{L} \{n_i(\Gamma_{kg}) - P_w(\mathbf{F_{kg}} = 0 | \mathbf{MB_{\Gamma_k}}(f_{kgl}))n_i(f_{kgl} = 0)$$
$$- P_w(\mathbf{F_{kg}} = 1 | \mathbf{MB_{\Gamma_k}}(f_{kgl}))n_i(f_{kgl} = 1)\} \tag{6}$$

Where $n_i(f_{kgl} = 0)$ is the number of true grounding of i^{th} formula while set $\mathbf{F_{kg}} = 0$, and similar for $n_i(f_{kgl} = 1)$. The learning of pseudo-log-likelihood in our approach are further boosted by the L-BFGS optimizer [24], to make entire process become more efficiency.

C. Matching of MLN-based descriptors

For each constructed input descriptor d_k^{in}, system would search for the matching descriptor in the database, and further arrange it to the proper set of identified($\mathbf{O^C}$) or unidentified object ($\mathbf{O^U}$) as shown in fig.3. The elaborate structure of multilayer networks in the databased can be illustrated as fig.5. The objects in the repository are composited by multiple rotation angles and descriptors. Each descriptor except the descriptor of target pose has one and only one corresponding rotation angle to guide robot are rotate object to the target pose. Since a descriptors is the combinations of predicates, the matching of descriptors can use the same concept of inference in section 3.2. The descriptors in the database 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:

$$\begin{aligned} L(\mathbf{D^{in}} = d_k^{in} | \mathbf{B} = \mathbf{d_m}) &= P(\mathbf{B} = \mathbf{d_m} | \mathbf{D^{in}} = d_k^{in}) \\ &= \textstyle\prod_{\eta} P(\mathbf{d_m} = d_{m\eta}, \mathbf{D^{in}} = d_k^{in}) \end{aligned} \tag{7}$$

$$\hat{d_k^{in}}_{MLE} = \mathbf{argmax}_{B=d_m} \hat{l}(\mathbf{D^{in}} = d_k^{in} | \mathbf{B} = \mathbf{d_m^*}) \tag{8}$$

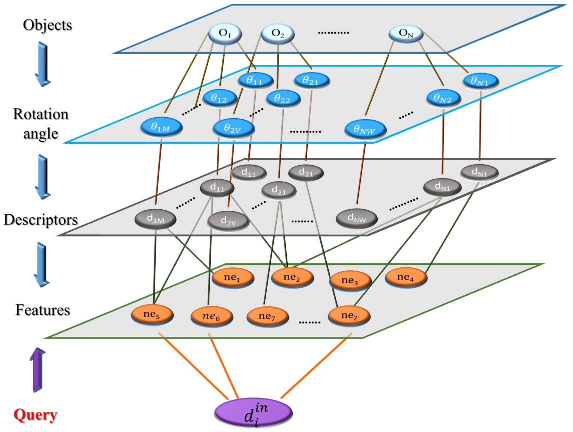


Fig. 5. Example of sampling Markov blanket

$\mathbf{d_m}$ represents the set of descriptor which are belonging object m, and acquire common predicates with d_k^{in}. The system would only consider one input object at a time.If descriptors in the sampled object have common predicates with the query descriptor, the descriptors would be considered as Markov blanket of d_k^{in}, and calculate likelihood function for all possible objects. The query descriptor would be matched the descriptors in the database depend on the result of maximum likelihood. $\mathbf{d_m^*}$ is the Markov blanket which has maximum likelihood of $d_k^i n$.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.

IV. Inference and Learning of Mapping between MLN-based Descriptor and Rotation Angle

Since MLN-based descriptors are matched according to the neighborhoods of clusters, 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, and make knowledge could be transferred between different domains in proposed multilayer network. We assume different faces of same object include at least one common predicates, and the common predicates can be used to infer the relation between input and target pose. Set of rotation angle Θ is composited by θ_R, θ_P, and θ_Y which represent roll, pitch and yaw angle of 3-DOF end effector of robot arm. Θ is unknown at first, because there is no prior knowledge of rotation angle for proposed system as mention before. Θ 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 reliable and accurate in several iteration. The mass center of each cluster is considered 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 differences are represent by vector which is formulated as:

$$\vec{V_c} = \vec{V_c^T} - \vec{V_c^{in}} \tag{9}$$

Where $\vec{V_c}$ is the vector of key atom c. $\vec{V_c^T}$ and $\vec{V_c^{in}}$ are the position vector of key atom c of target and input descriptor. According to the MLN-based descriptors, the formula with higher weighting means more reliable. Reminding the example in table 1, every predicates of a formula belong to the same key atom, so the weight of each formulas could be further considered as the reliability of each center atom. Hence, we choice key 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 $\boldsymbol{\vec{V}_{mb}}$ which means the set of vector of descriptor b in object m. The set of vector $\boldsymbol{\vec{V}_{mb}}$ 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 transfer function which can predict ideal rotation angle depend on different input descriptors. There are two factors have to be concerned: (1) the distribution of historical vector $\boldsymbol{\vec{V}_{mb}}$. (2) likelihood of input and target descriptor. Hence, the transfer function could be formulated by joint distribution:

$$P(L(d_{kT}^{in}|d_m^T), \mathbf{\vec{V}_{mb}}) =$$

$$\frac{1}{Z_V} exp(\textstyle\sum_k \sum_b \lambda_t \mathbf{F}\{L(d_k^{in}|d_m^T) = l(d_k^{in}t|d_{mt}^T), \mathbf{\vec{V}_{mb}} = \vec{v}_b\}) \tag{10}$$

Where $L(d_{kT}^{in}|d_m^T)$ is a set of likelihood of input descriptor k and target descriptor during a period of time T. $l(d_k^{in}t|d_{mt}^T)$ is the likelihood at time t. $\mathbf{F}\{*\}$ is feature function which is 1 while $*$ is true, and 0 otherwise. λ_t is weight if feature function. This transfer function represents the mapping between set of input descriptors and the same target descriptor d_m^T. While derive a new input descriptor, the predicated result can be derived by:

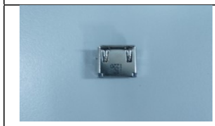
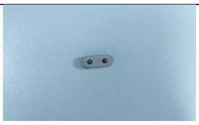
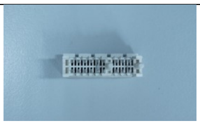
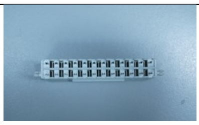
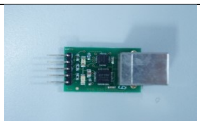
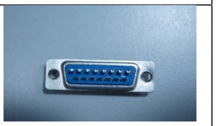
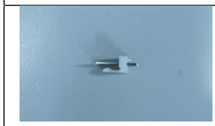
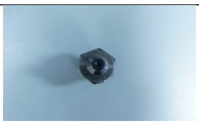
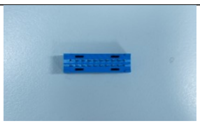
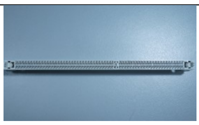
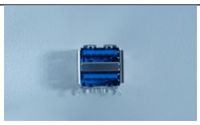
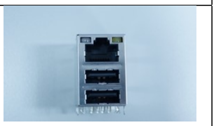
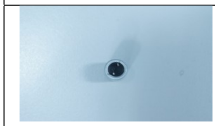
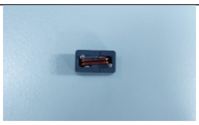
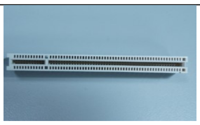
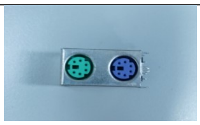
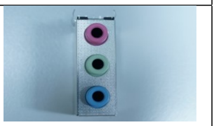
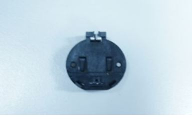
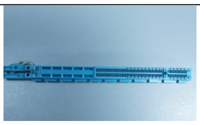
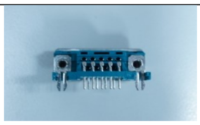
$$\mathbf{argMax} P(\vec{V^*}_{\vec{V}-\vec{V}_{fail}}) =$$

$$P(\vec{V}^*|l(d_t^{in}|d_m^T), L(d_{t-1}^{in}|d_m^T), \mathbf{\tilde{V}_{mb}}) \tag{11}$$

V. Experiments

The inputs of proposed system are serial images of each object, so most of open source databases can not be applied for proposed system, and also not suit for purposes in this paper. Therefore, the experiments are implemented through our own dataset. The testing dataset is constructed by twenty different kinds of chosen work pieces as shown in table 2. The experiment is implemented based on several assumptions: The input objects are not occluded, and not adjacent with each other. The input objects are placed on conveyor with random

Table II. Three classes of testing work pieces for experiments

WP1		**WP2**		**WP3**	
WP1.1	WP1.5	WP 2.1	WP 2.5	WP3.1	WP3.5
					
WP1.2	WP1.6	WP 2.2	WP 2.6	WP3.2	WP3.6
					
WP1.3	WP1.7	WP 2.3		WP3.3	WP3.7
					
WP1.4		WP 2.4		WP3.4	
					

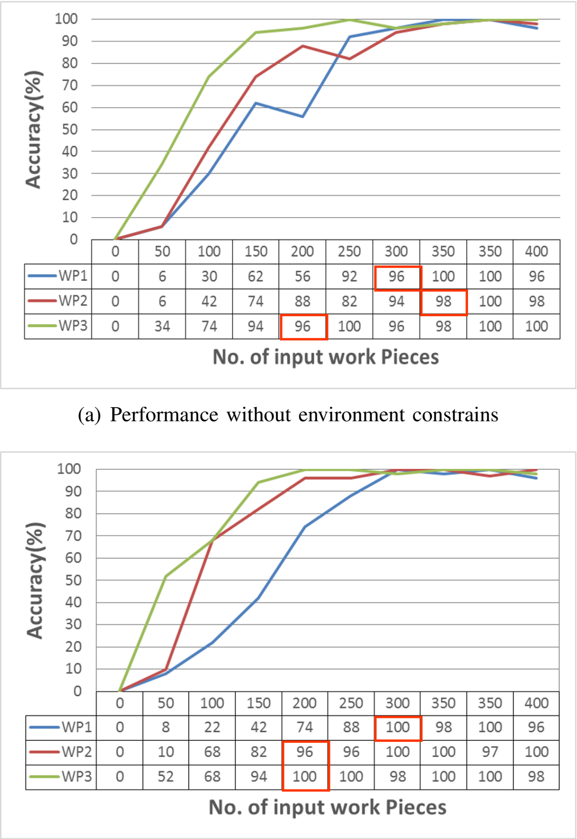
poses, and we assume the probability of every faces showing on top is uniform distribution.

The testing work pieces are classified into three classes in table 2. For class $\mathbf{WP1}$, the work pieces are featureless and small, so it's hard to construct robustness descriptor even building relational model for entire model. For class $\mathbf{WP2}$, all work pieces acquire similar shapes or size, so, for general method, this kind of object is easily mismatch in the matching process. The work pieces in $\mathbf{WP3}$ are matched group of this experiment. The work pieces acquire sufficient feature for descriptor, and have plenty of information for identifying and constructing relational model. In the first stage of experiment, we would like to compare the performance of proposed system between different classes in different environments. The results of different classes are shown in fig.6. In fig.6(b), the environment lighting is controlled by on-axis lighting source, so the information of object are more complete and distinct than images with lighting control in fig.6(a). The accuracy of recognized result is average of 100 times repeatedly testing.

The system is considered convergence while accuracy is over 95%, and stop learning approach. If the accuracy is under 95% again, the learning approach would be re-excuted. Comparing the results, in both cases, class $\mathbf{WP3}$ could be convergent with least input sample, and convergent time of class $\mathbf{WP2}$ is slowest. The results shows the efficiency of learning could be slightly improved by environment constrain, but the accuracy is not effected, and always over 95% after learning approach stopped. Similarly, twenty kinds of work pieces are included in learning stage in the same time, and the results are shown in fig. 7. The accuracy of each test is also the average of 100 times repeatedly testing. The results show that system need more inputs to convergent while more kinds of objects are included in learning stage. The performance is also slightly improved while environment is lighting controlled. In brief, these two experiments verify proposed system is competent to learn the relational model automatically. Although the learning rate would be dragged by the kinds of input objects, the learning rate still can be convergent by reasonable number of inputs. The result shows the system can be convergent by less than 1000 sample pieces with random poses. Furthermore, the accuracy of recognition is stable once learning approach completing, and would not be under threshold(95%) again unless adding new kind input.

The experiment in fig. 6 and 7 testified the performance of proposed system could automatically learn the relational model of variant kinds of objects. Then, we would like to compare the performance of proposed system with other advanced approaches. Since none of similar systems could handle this issue in our survey, so the comparisons would be done by dividing our system into two parts. One is 2-D descriptors for each face of objects, and the other is machine learning approach for learning relational model.

For the descriptor part, four kinds of other descriptors are chosen to compare with proposed system. B-SIFT[25]and Edge-SIFT[26] are modified versions of SIFT approach which enhanced the accuracy of feature point registration. BRISK[13] descriptor is constructed based on binary robust invariant scalable key points, and Zernike Moment (ZM)[11] phase-based descriptor is a moment-based descriptor which use the phase information of signal. All of these descriptors are representative methods in relative field recent years, and had been testified by plenty of researchers. To compare the robustness and accuracy, the performance is testified by two conditions. One is relationship of each faces is prior of system, and the descriptors only provide information for object matching. The experiments are implemented by the same learning approach which proposed in previous section.The other is no prior for learning approach that information of descriptor need



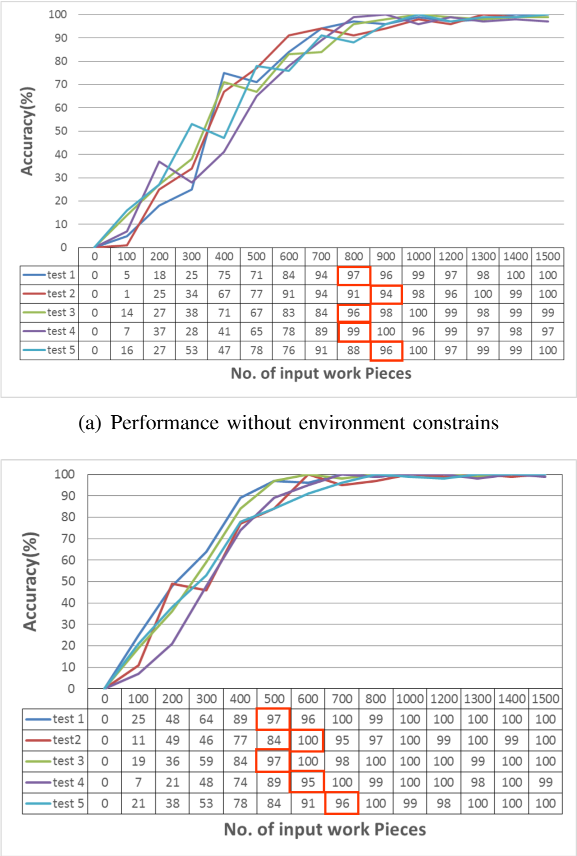
	0	50	100	150	200	250	300	350	350	400
WP1	0	6	30	62	56	92	96	100	100	96
WP2	0	6	42	74	88	82	94	98	100	98
WP3	0	34	74	94	96	100	96	98	100	100

(a) Performance without environment constrains

	0	50	100	150	200	250	300	350	350	400
WP1	0	8	22	42	74	88	100	98	100	96
WP2	0	10	68	82	96	96	100	100	97	100
WP3	0	52	68	94	100	100	98	100	100	98

(b) Performance with environment constrains

Fig. 6. Experimental Results of different classes in different environment constrains



	0	100	200	300	400	500	600	700	800	900	1000	1200	1300	1400	1500
test 1	0	5	18	25	75	71	84	94	97	96	99	97	98	100	100
test 2	0	1	25	34	67	77	91	94	91	94	98	96	100	99	100
test 3	0	14	27	38	71	67	83	84	96	98	100	99	98	99	99
test 4	0	7	37	28	41	65	78	89	99	100	96	99	97	98	97
test 5	0	16	27	53	47	78	76	91	88	96	100	97	99	99	100

(a) Performance without environment constrains

	0	100	200	300	400	500	600	700	800	1000	1200	1300	1400	1500
test 1	0	25	48	64	89	97	96	100	99	100	100	100	100	100
test2	0	11	49	46	77	84	100	95	97	100	99	100	99	100
test 3	0	19	36	59	84	97	100	98	100	100	100	99	100	100
test 4	0	7	21	48	74	89	95	100	99	100	100	98	100	99
test 5	0	21	38	53	78	84	91	96	100	99	98	100	100	100

(b) Performance with environment constrains

Fig. 7. Experimental Results of all work pieces in different environment constrains

to use for inferring the relational model. The ZM descriptor have the best performance in the condition without prior, but accuracies of descriptors are close. In condition without prior, the MLN-based descriptor acquire best performance which testified MLN-based descriptor is suited for automatic learning system.

Hereafter, the performance of different learning methods should be further discussed. The learning approach in proposed system need to learn the distribution of different domains, so regular learning approaches is hard to applied on proposed structure directly. The transfer learning methods are famous for handling cross domains problem, so the other three kinds of transfer learning approaches: Locally Weighted Ensemble approach(LWE)[7], Transductive SVM(TSVM)[8], and Weighted Neural Network(WNN)[9] are chosen to compare with proposed method. Similarly, the experiments are divided into two parts as shown in table 4. The result shows LEW had the best accuracy in the condition with prior, and proposed learning approaches acquire greatest performance in condition without prior, but, in both two conditions, the performance between different methods are pretty close. It's seem the results are mainly effected by the performance of the descriptor. The performance of descriptor not only influence the result of 2-D image, but also the relational model of each faces, so the experiments are reasonable. Furthermore, the results showed proposed system acquire best performance in automatic learning part.

Table III. Comparisons of system performance with different 2-D descriptors

		Descriptor				
		MLN-based	**B-SIFT**	**Edge-SIFT**	**BRISK**	**ZM**
With prior	WP1	0.9781	0.9664	0.8384	0.9556	0.9788
	WP2	0.9630	0.9766	0.9233	0.9676	0.9523
	WP3	0.9901	0.9963	0.9454	0.9899	0.9949
	All	0.9594	0.8982	0.8066	0.9432	0.9634
Without prior	WP1	0.9611	0.7688	0.6544	0.8103	0.7787
	WP2	0.9505	0.7043	0.7123	0.7197	0.7979
	WP3	0.9718	0.8044	0.7963	0.8243	0.8231
	All	0.9543	0.6431	0.6144	0.7741	0.7670

Table IV. Comparisons of different transfer learning approach

		Transfer learning approach			
		Proposed	**LEW**	**TSVM**	**WNN**
With prior	WP1	0.9781	0.9802	0.9511	0.9513
	WP2	0.9630	0.9763	0.9690	0.9601
	WP3	0.9901	0.9899	0.9799	0.9684
	All	0.9594	0.9677	0.9567	0.9541
Without prior	WP1	0.9611	0.9601	0.9543	0.9103
	WP2	0.9505	0.9543	0.9558	0.9197
	WP3	0.9718	0.9788	0.9699	0.9346
	All	0.9603	0.9553	0.9497	0.9486

VI. Conclusion

The automatic learning approaches for visual-servo system is an important part in industrial application. In this work, we reverse the concept of traditional visual-servo system. The robustness of feature points and descriptor is not the thing which should be most concerned. Instead, the relational model between input and output is the most important.

To learn the relationship between input and output, we proposed a system architecture which can automatic learning and self-supervised the performance of learning or inference results. Since the relationship between input and output is consisted by multiple different domains, the relational model is further segmented into three domain:2-D descriptor, 3-D object and Cartesian space of robot arm. Instead of modelling by three independent networks, these three domains are integrated into one multi-layers networks. Comparing with traditional multi-layers percetron, proposed multi-layer networks is not used to model a complex non-linear distribution, but model the transfer function between different layers. The knowledge in different domains could be transfer between different layers through transfer functions which are called relational models in this paper. Furthermore, a MLN-based descriptor is further proposed to assist inference relationship of each 2-D descriptor while the relational model is unknown. The experiments result shows the system could automatic learning the relational model, and performance could compete with other excellent methods.

.

References

[1] Torgny Brogrdh, *"Present and future robot control developmentAn industrial perspective"*, Annual Reviews in Control, Volume 31, Issue 1, pp. 6979, 2007.

[2] Ebrahim Mattar, *"Robotics Arm Visual Servo: Estimation of Arm-Space Kinematics Relations with Epipolar Geometry, Robotic Systems - Applications, Control and Programming"*, Dr. Ashish Dutta (Ed.), ISBN: 978-953-307-941-7, InTech, DOI: 10.5772/25605.

[3] So-Youn Park ,Yeoun-Jae Kim ,Ju-Jang Lee ,Byung Soo Kim, and Khalid A.Alsaif,*"Controlling robot arm manipulator using image-based visual servoing without pre-depth information"*, 37th IEEE Interantional Conferece on Industrial Electronics, pp.3157-3161,Nov. 2011.

[4] K. Deguchi, H. Sakurai, and S. Ushida,*"A Goal Oriented just-in-time visual servoing for ball catching robot arm*, in Int. Conf. on Intelligent Robots and Systems, Sept. 2008, pp. 30343039. Sašo Džeroski*"Multi-relational Data Mining: An Introduction"*, SIGKDD Explore Newsletter, Volume 5, Issue 1, July 2003, pp.1-16.

[5] Sinno Jialin Pan, and Qiang Yang,*"A Survey on Transfer Learning"*, Knowledge and Data Engineering, IEEE Transactions on , vol.22, no.10, pp.1345,1359, Oct. 2010

[6] T. Dietterich, L. Getoor, and K. Murphy,*"Statistical Relational Learning and its Connections to Other Fields"*, ICML-2004 Workshop on Statistical Relational Learning (SRL), Banff, Canada, July 2004.

[7] Jing Gao and Wei Fan and Jing Jiang and Jiawei Han, *"Knowledge Transfer via Multiple Model Local Structure Mapping"*, in the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,pp. 283-291,New York, USA,2008

[8] T. Joachims, *"Making large-scale svm learning practical."*, advances in kernel methods - support vector learning, MIT-Press, 1999. D. Lowe, *"Distinctive Image Features from Scale-Invariant Keypoints"*, International Journal of Computer Vision 60(2), pp. 91110, 2004.

[9] A. J. Carlson, C. M. Cumby, J. L. R. Nicholas D. Rizzolo, and D. Roth,*"Snow learning architecture"*, Technical report UIUCDCS, 1999.

[10] H. Bay, A. Ess, T. Tuytelaars, and L. Gool,*"SURF: Speeded up robust features"*, Comput. Vis. Image Understand., vol. 110, no. 3, pp. 346359, Mar. 2008.

[11] Zen Chen and Shu-Kuo Sun,*"A Zernike Moment Phase-Based Descriptor for Local Image Representation and Matching"*, IEEE Trans. Image Process., vol. 19, no. 1,pp. 205219, Jan. 2010.

[12] A. Alahi, R. Ortiz, and P. Vandergheynst,*"Freak: Fast retina keypoint"*, CVPR, 2012.

[13] S. Leutenegger, M. Chli, and R. Siegwart,*"Brisk: Binary Robust Invariant Scalable Keypoints"*, International conference on Computer Vision, 2011.

[14] Vijay Chandrasekhar, Gabriel Takacs, David Chen, Sam S. Tsai, Jatinder Singh, and Bernd Girod,*"Transform coding of image feature descriptors,"*, SPIE 7257, Visual Communications and Image Processing, 2009.

[15] Matthew Richardson and Pedro Domingos,*"Markov logic networks,"*, International Journal of Machine Learning, Volume 62, Issue 1-2,pp 107-136, Feb. 2006.

[16] L. Mihalkova, T. Huynh, and R.J. Mooney,*Mapping and Revising Markov Logic Networks for Transfer Learning,*, Proc. 22nd Assoc. for the Advancement of Artificial Intelligence (AAAI) Conf. Artificial Intelligence, pp 608-614, July 2007.

[17] Kok, Stanley and Domingos, Pedro,*"Learning the Structure of Markov Logic Networks"*, Proceedings of the 22Nd International Conference on Machine Learning, pp 441-448, Germany, 2005.

[18] Parag Singla and Pedro Domingos,*"Discriminative training of Markov logic networks"*, Proceedings of the international Conf. on Artificial Intelligence, 2005.

[19] J. Shi and J. Malik,*"Normalized cuts and image segmentation"*, Pattern Analysis and Machine Intelligence, IEEE Transactions on , vol.22, no.8, pp.888,905, Aug 2000.

[20] Karthikeyan Vaiapury, Anil Aksay and Ebroul Izquierdo,*"GrabcutD: Improved Grabcut Using Depth Information"*, Proceedings of the 2010 ACM Workshop on Surreal Media and Virtual Cloning, pp 57-62, New York, USA, 2010.

[21] Z. Zivkovic,*"Improved adaptive Gaussian mixture model for background subtraction"*, Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on , vol.2, no., pp.28,31 Vol.2, 23-26 Aug. 2004.

[22] G. Frahling and C. Sohler,*"A fast k-means implementation using coresets"*, Proceedings of the twenty-second annual symposium on Computational geometry (SoCG),2006.

[23] Tong Simon, and Daphne Koller,*"Support vector machine active learning with applications to text classification"*, The Journal of Machine Learning Research 2 pp 45-66, 2002.

[24] Fei Sha and Fernando Pereira,*"Shallow parsing with conditional random fields"*, Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology, Volume 1, 2003.

[25] Yanning Zhang,Zhi-Hua Zhou, Changshui Zhang and Li, Ying,*"B-SIFT: A Highly Efficient Binary SIFT Descriptor for Invariant Feature Correspondence"*, Intelligent Science and Intelligent Data Engineering, pp 426-433, 2012

[26] S. Zhang, Q. Tian, K. Lu, Q. Huang and W. Gao,*"Edge-SIFT: Discriminative binary descriptor for scalable partial-duplicate mobile search"*, IEEE Trans. Image Process., vol. 22, no. 7, pp. 28892902, Jul. 2013.