[[1]](#footnote-1)

A Self-Taught Vision System for Automatic Learning and Recognizing 3D Objects

Ren C. Luo∗, and Po-Yu Chuang†,

† Electrical Engineering Department, National Taiwan University, Taipei, Taiwan 106

*Abstract—Vision systems for 3D object recognition are widely applied on industrial robot arm recent years. In most of tasks, we only cares about the relation between input image and result rather than a delicate model of 3D object. We propose an intelligent vision system which can automatically construct relational model for input and output by self-taught learning method with only 2D image input. The relational model is established based on hierarchical Model. 2D images are used to infer the rotation angle for robot in Cartesian space. Therefore, we propose a descriptor based on Markov Logic Network(MLN) which is suitable finding relational model. The experimental results show the feasibility of proposed structure that can transfer knowledge in different domains, and complete assigned task by only modeling the relation between input and output.*

*Index Terms—Visual servo system, Hierarchical Model, Machine learning*

# INTRODUCTION

Robot arm with vision have been widely applied in automatic industrial production line in recent years [1-4]. Vision system provides additional information to make robot arm become more adaptive to complete various tasks. For vision system, model-based recognition methods are commonly used in industrial applications. The performance is mainly related to the manually labeled data. 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 task-oriented vision system which acquires ability in automatic learning relational model of input and target. 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 target face of 3D objects which are intended to be placed on top by robot arm, but the other faces of 3D objects are unknown. The labelled data is target faces, and inputs 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[5,6] which end in a linear or kernel classifier is not enough. We introduced a hierarchical model to tackle these problems.

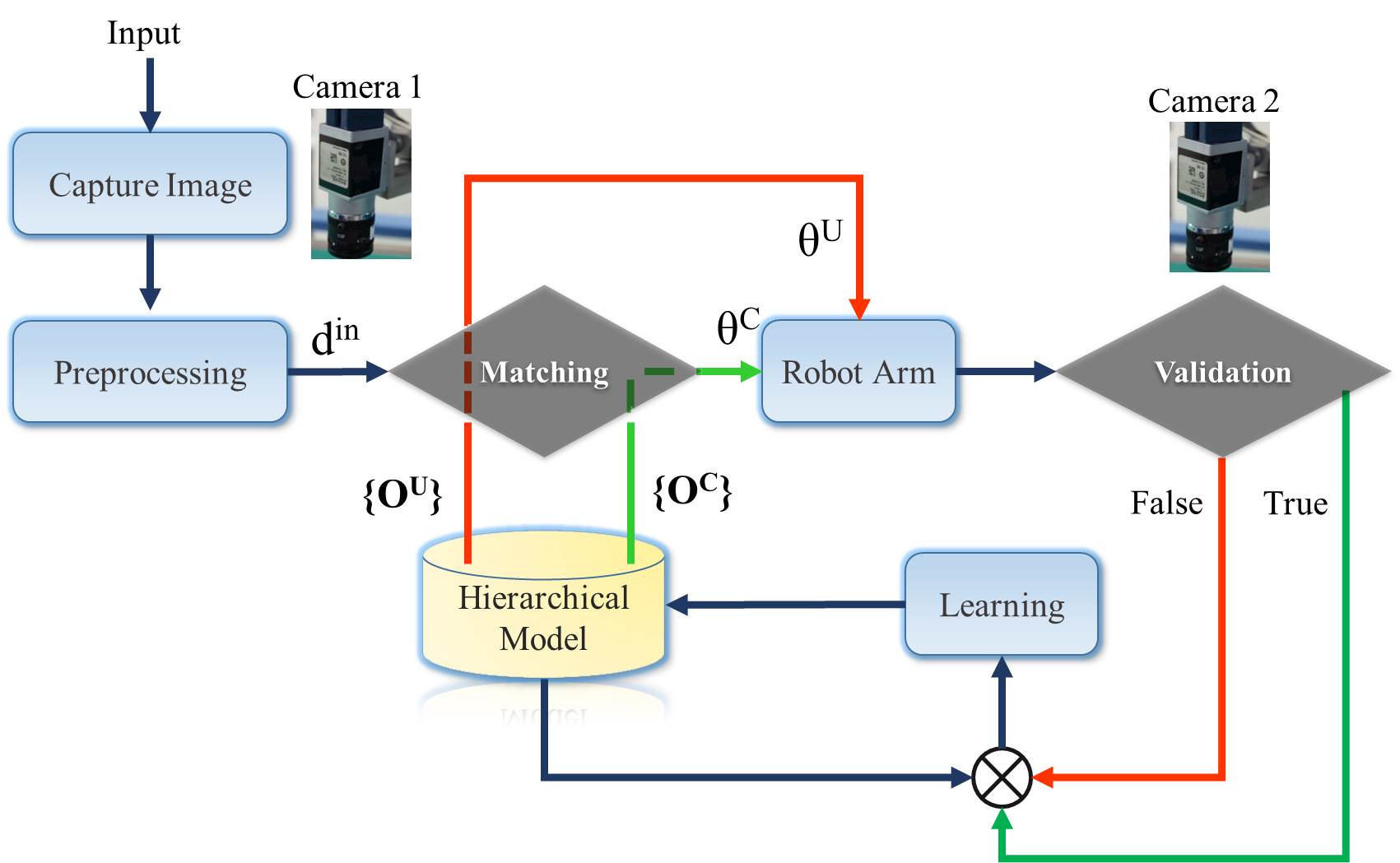
The learning of multilayer model achieve dramatically success recent years. Hinton et al. [7] proposed deep structure learning which hidden layers are formed by lower level feature to higher level hierarchically, and had been successfully applied on different research fields[8-11]. Comparing with traditional ***Artificial Neural Network (ANN)*** model, the deep learning method is aim to learn the representation of data in different level rather than produce 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**, **Descriptor**, **Object** and **RotationAngle**

Through this model, the feature in different domains could be correlated through hierarchical structure, but the system still cannot automatically learn the relation between input images and output rotation angle. Being a automatic system, the ability which could "infer" latent edges between labeled and unlabeled data 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 quite different, so we design three modules to tackle self-taught 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 methods [12-16] 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 case. 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 on the ***Markov Logic Network (MLN***) [17-20]. MLN is an approach combines first-order logic and probabilistic graphical model. First-order logic enables 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 edges. Transfer information module is realized by Self-taught Clustering algorithm [21]. Self-taught Clustering algorithm is a transfer learning method [22-25] 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 will rotate target object from input face to output face. Finally, the validation module is an eye-to-hand camera which used to validate the error between the output face and desired target face. Then, the validation module feedback the error to the model in order to refine the existed model. Through these three modules, proposed system can automatically learn the relations between input images and corresponding rotation angles with only labelled 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 recognizing object is described in section 3. Section 4 introduces how to model the proposed hierarchical networks, and learn by self-taught learning. Then, we compare the performance of proposed system with several states of development in section 5. Finally, reviewing performance and conclusions are presented in final section.

 Fig. 1 System architecture

# SYSTEM ARCHITECTURE

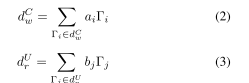
The main purpose of this system is to automatically derive the relationship between input face and target face of 3D assigned objects. The only prior knowledge are the target face. Input is arbitrary assigned object with random face on top, so input is very likely an unknown face of assigned object showed-up rather than prior target face. Therefore, system has to infer the correlation between input and existed priors. Proposed system is shown in Fig. 1. Camera 1 captures images of all input objects with random faces on top, and constructs MLN-based descriptor for each input. Then, system matches the input with data in database and output rotation angle for robot arm. After robot arm placing an object, camera 2 will validate result, and feedback error for refining existed model. The system architecture in Fig. 1 is realized by a hierarchical-deep model in Fig. 2.

The variables in the same layer are independent, and vertical adjacent two layers are full connected. Variables in **Feature**(**ΓC**) layer are extracted image features, and variables in both **Classiﬁed Descriptor**(**DC** ) and **unclassiﬁed Descriptor**(**DU**)are MLN-based descriptor. Variables in **Rotation angle(ΘC)** and **Inferred Rotation angle**(**ΘU** ) are set of rotation { Row (α), Pitch(β) , Yaw(γ) } angles respect to target faces. Finally, variables in **Object**(**OC** ) are combinations of rotation angles.

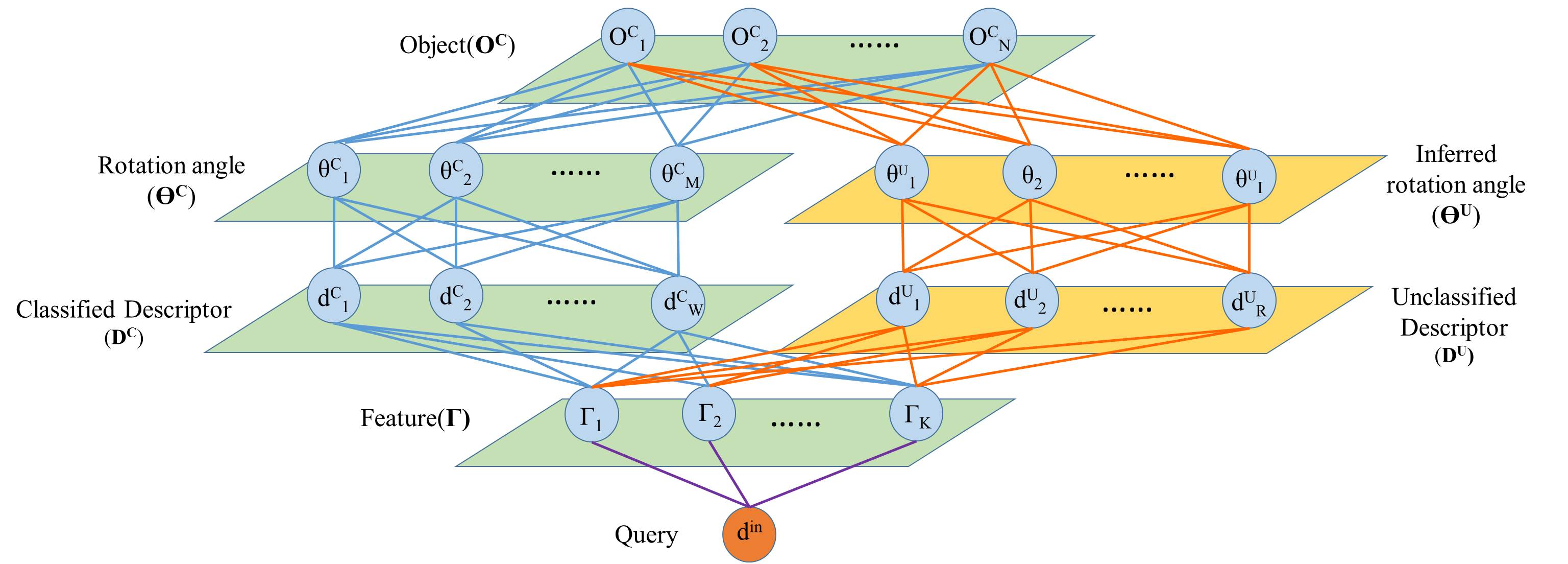
The difference between classic ***Deep Belief Networks(DBN*)** is that proposed model exist two parallel parts in Fig. 2. **DC** -**ΘC** and **DU** -**ΘU** have no connection between each other, but both have full connection with deepest layer **OC** and first layer **ΓC**. To handle tons of unknown 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 evidences have been discovered. For variable dCw  in layer **DC**, the sparse coding result should be formulated as:

 (1)

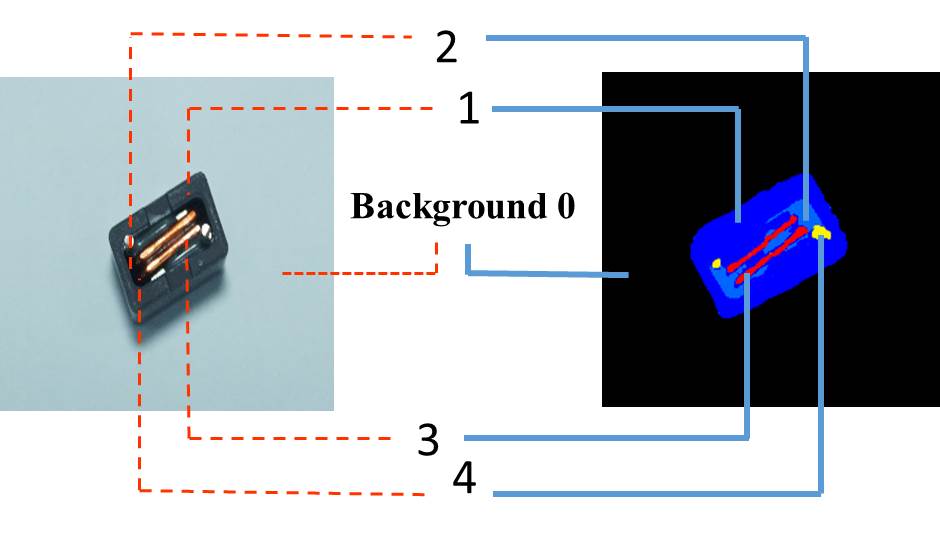
Although Eq.(1) can handle the problem of latent edge, it's impractical to sample all possible conditions whenever new evidence showing up. Therefore, proposed model separate descriptor layer into two parallel parts as:



dCw  is considered a prior descriptor which the edge between **ΘC** and **OC** had been established. Therefore, the left part of parallel layers can be considered as static model until there is a query classified to the **DU**. **DU** layer is the set of descriptors which we haven't known that these descriptors are correspondent to which object. Therefore, we propose an inference method to infer the possible rotation angle, and camera 2 will check inferred results. If inference is success, variable dUr and θU are used to re-estimate correlation between layers through hierarchical structure. Therefore, latent edges can be revealed though more success inferences.



**Fig. 2. Hierarchical-deep model for self-taught system**



(a) Result of background subtracting and clustering



(b) Serial captured frames

**Fig. 3. Preprocessing of input objects**

# MLN-based Descriptor

## The concept of constructing MLN-based descriptor

Being a self-taught system, deriving more valuable information from raw data helps system deriving more reliable results with scarce prior knowledge. Most of present image descriptors [12-16] are constructed based on strong sparse feature point, because these points are consistent even in different environment. These kinds of descriptor can efficiently and precisely match given image. Nevertheless, most of observed face is not in prior data, so we need a descriptor which can infer the relation between observations and priors. To avoid losing information, we choose normal distributed feature instead of sparse feature. Since different faces of an object may exist different strong features, normal distributed feature is more suitable for our case. Reference regard as a stable and normal exist in geometric space.

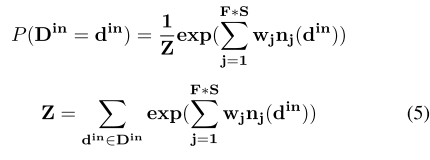
**Table I. Example of predicates and ﬁrst-order logic formulas**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key Atom | 1 | 2 | 3 | 3 |
| 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) |  |  |  |

For prepossessing of input images, each channel of RGB domain is classified into 5 parts, and get 125 classes in RGB domain. An input image will be segmented by these classes. In Fig. 3(a), an observed face of input object is segmented into 4 classes, and class 0 is background. Hereafter, predicates for MLN networks are constructed by segmented results. We have only two kinds of predicate ***ne(a,v)*** and ***des(x)***for MLN model. Variable ***a*** is an atom cluster, and variable ***v*** is a neighbor of atom cluster, so predicate ***ne(a,v)*** represent adjacency of atom cluster. Variable ***x*** in ***des(x)*** is a MLN-based descriptor. The variables of feature layer in Fig. 2 are predicates ***ne(a,v)***. Since every classes can be the atom cluster, we have C1252 binary variables in feature layer. Taking Fig. 3(a) as an example, the predicates of preprocessed image are shown in Table Ⅰ, and first order logic is formulated as:

∀a∀v *ne(a,v) ⇒ des(x)* (4)

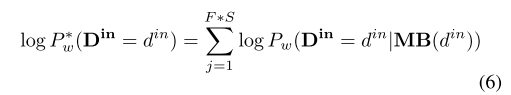
Each image will further be down sampled, and derived several images with different scales. For each image, we derive ***F\*S*** formulas where ***F*** is number of serial captured images and ***S*** is number of images with different scales. Through these formulas, a MLN model can be constructed. The probability distribution over possible world din specified by MLN is given by:



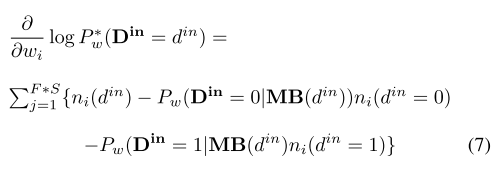
Where din is the descriptor of input image. nj (din ) is the number of true grounding of formula j in din, and wj is weight of formula j .

## 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 consider 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 consider relational data. The pseudo-log -likelihood of Eq.(5) can be written as:



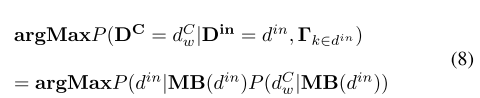
Where MB(din) is Markov blanket while din is observed. The MLN weights are learned generatively by maximizing the pseudo-log-likelihood of Markov blanket. The gradient of the pseudo-log-likelihood with respect to the weight is:



Where ni (din = 0) is the number of true grounding of j th formula while force din = 0, and similar for ni (din = 0\1). The learning of pseudo-log-likelihood in our approach are further boosted by ***Limited-memory Broyden-Fletcher-Goldfarb-Shanno(L-BFGS)*** optimizer [20] to make entire process become more efficiency.

## Matching of MLN-based descriptors

For each constructed input descriptor din, system would search for the matched descriptor in the database, and further arrange it to the proper layer of **DC** or **DU** as shown in Fig.2. Since input is possible to be assigned to one of parallel layers, matching step is separated into two parts. One is using pseudo-log-likelihood for deciding observation should be assigned to which layer. The pseudo-log-likelihood of descriptors matching could be formulated as:



If input descriptor doesn’t match any descriptor in **DC** layer,

the descriptor become a variable of **DU** layer. For a variable

in **DU** , we would like to infer rotation angle to make input

object can be placed on corresponding target face. Since the

rotation angles for descriptors in **DU** had been identiﬁed, the

second part for matching is try to ﬁnd a descriptor in **DC**

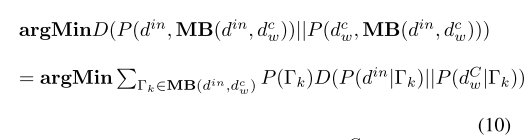
which have max co-cluster with input descriptor. Finding max

co-cluster can be alternately considered as minimizing loss of

information as:



The common feature Γk∈din ∩dWC is further represented by co-Markov Blanket of din and dC , and the loss of mutual information can be further formulated by KL divergence **[28]** as:

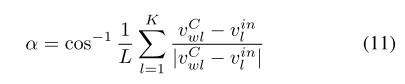


By Eq.(10), classiﬁed descriptor dCw with min DL-divergence is considered acquired max co-cluster with dCw.The relation between the co-cluster become the evidence forinferring rotation angle of din. Through Eq.(8) and Eq.(10),the input descriptors are classiﬁed to corresponding layer, andbecome inputs **ΘC** or **ΘU** layer.

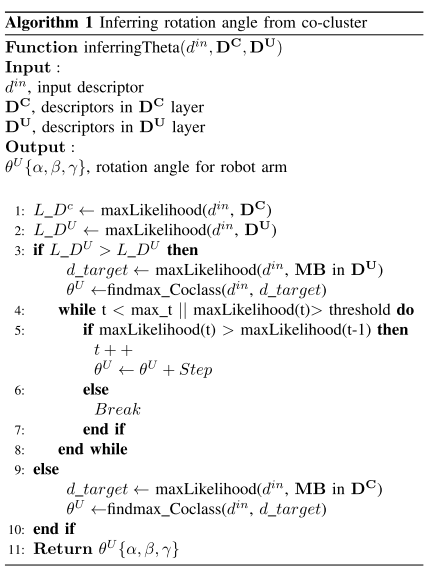
# Hierarchical Model

## Inference of rotation angle in **ΘU** layer

Inference rotation angle θiU is based on max co-cluster between din and dC. A set of co-cluster {Cw1 , Cw2 , ..., CwL } can be derived by minimizing KL divergence. The center of co-cluster with respect to center of camera in Cartesian space can be derived into two sets: **Vin** ={v1 , v2 , ..., vL } and **Vw** ={vCw1 , vCw2 , ..., vCwL }. The roll angle α of robot arm is calculated by:



Where roll angle α is the mean angle of co-cluster in two descriptors. As for pitch angle β and yaw angle γ, the pitch and yaw angle are hard to be estimated by 2D descriptor directly. We make random sample these two angles in value π/2 , and −π/2 initially, and approximate to actual angles by algorithm 1.



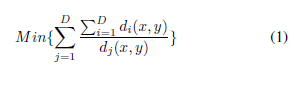
## Inference and learning of hierarchical-deep model

Proposed hierarchical model is a generative model of **Deep**

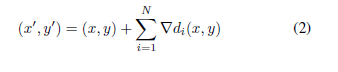
**Belief Network (DBN)**. Structure between each layer is shown in Fig. 4. each layer is considered as an **Restricted Boltzmann** **Machine (RBM)[8]** except **Γ** , **DC** , and **DU** . The MLN is trained by pseudo-log-likelihood as mentioned previously, and RBM is trained by greedy layer-wise training [30].

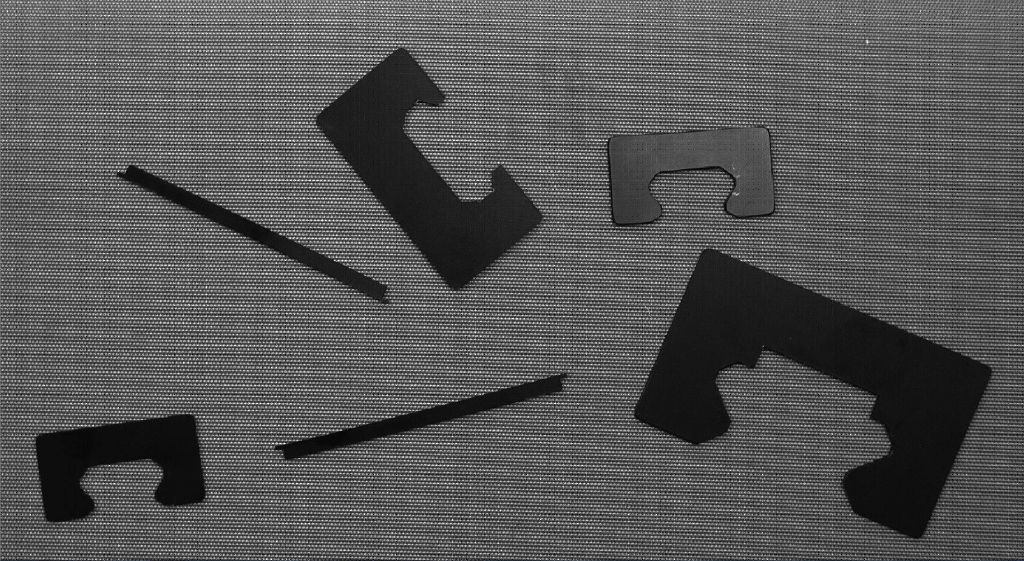
O is center point of image, and xW, yH is max width and height, and weighted coefficient depends on the ratio of Euclidean distance. The result of reference class through discriminative histogram is shown on Fig.1(b). The white block templates are members of reference class, and difference parts can be further estimated number and position by relation with reference.

While feature points mapped to the geometric space, one object might include multiple classes or one class might belong different objects. To further clarify the composition of one single object, we aspect to find invariant and, stable points, cell of potential object in geometric space which represent regional key points of each class, and all key points are respect to reference class. Cell is considered as center of feature class in distinct object, so cells to reference in every searching direction have to be local minimal:



It is expensive to be rechecked each element for convergence. Based on Eq. (1), di(x,y) is a vector which represents distance to reference in ith direction, so gradient of summation of d(x,y) must direct to the direction with maxima variance of magnitude. While (x,y) move along direction of maxima variance, the cell point, as center point, can be approach by several iteration. The new point (x',y') can be derived by:





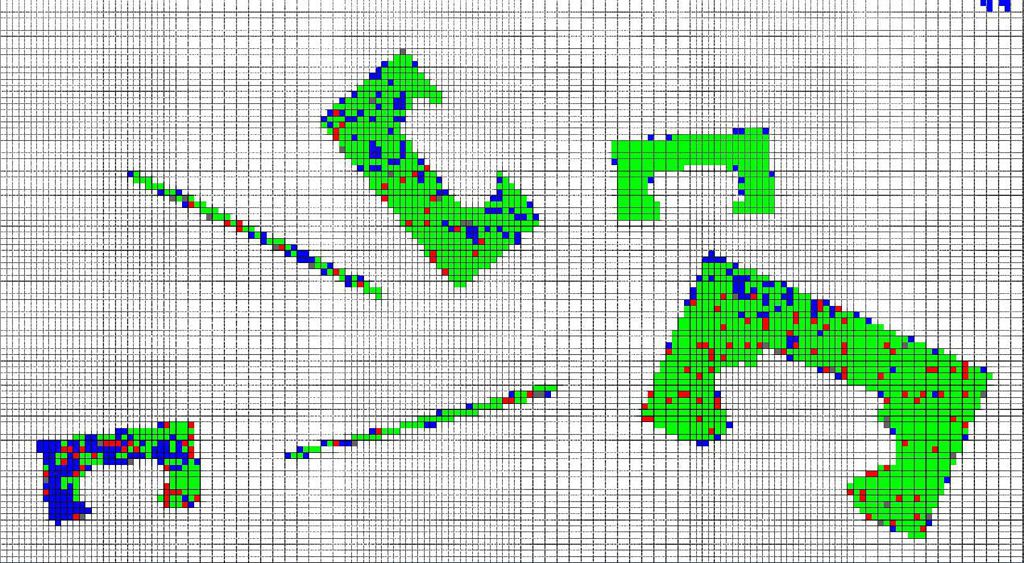


(a). Original

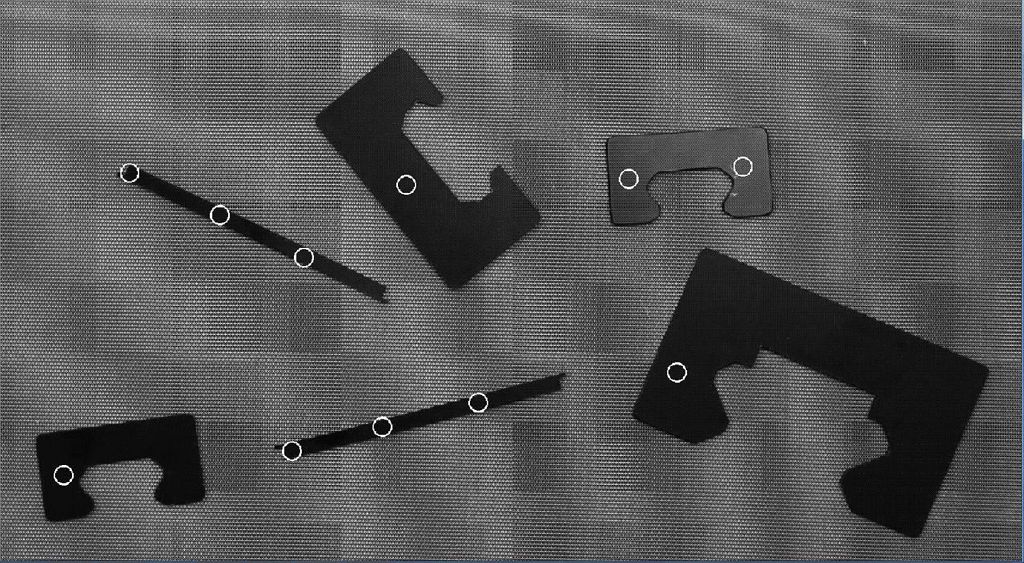
Fig. 1 Result of Reference Class

Since then local minimal of Eq. (1) can convergence through several iteration of Eq. (2). Depending on different situations, structure or pose, one object might result multiple cells as Fig. 2(b). Every object must have at least one cell to further expand, and multiple cells in single object would merge into one through succeeding expanding process.

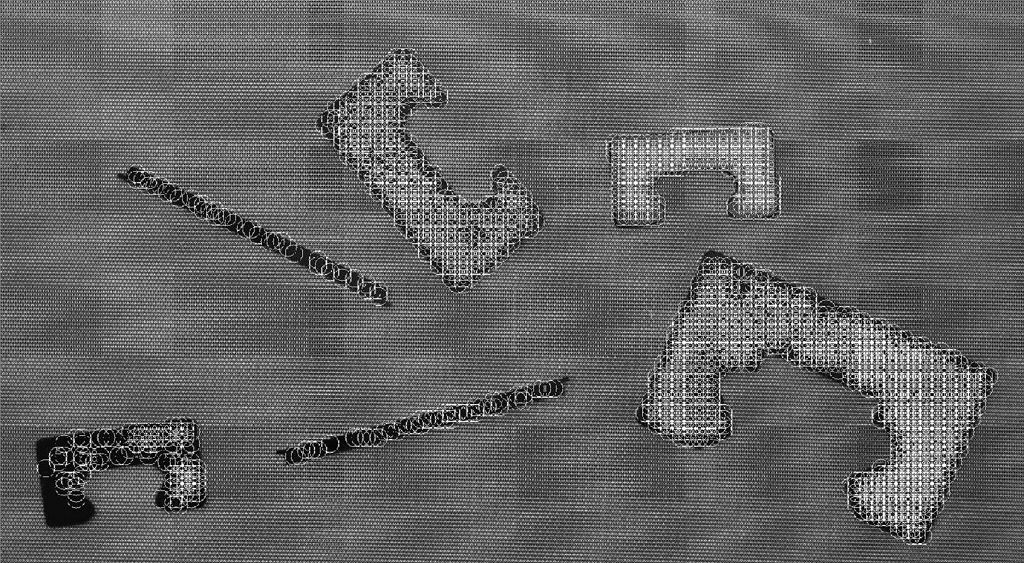
Additionally, the reference class should distribute homogeneously, but, in reality, background might divide into several feature classes caused by nonhomogeneous lighting, or viewpoint. If members of reference are not large enough, the



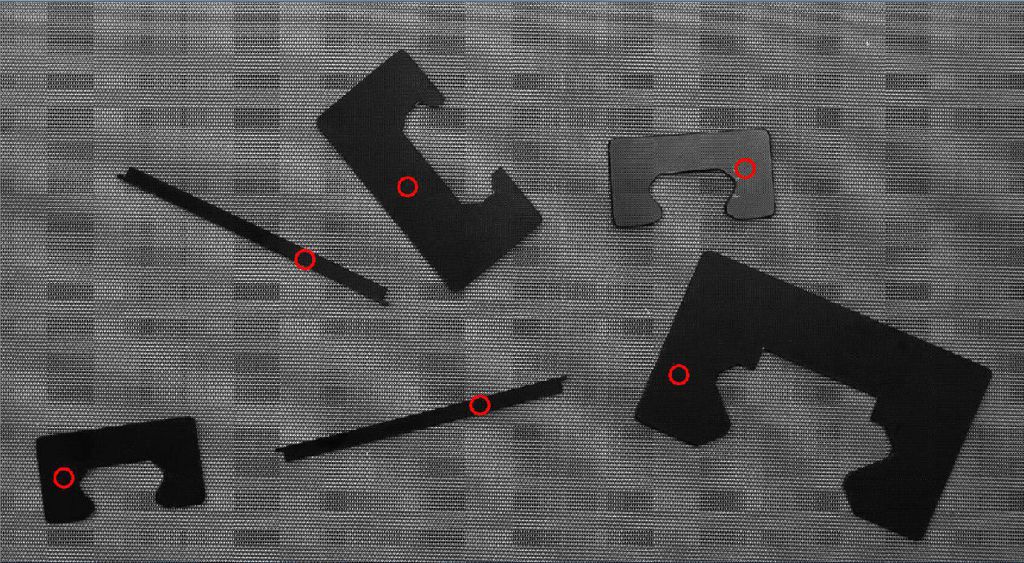
(a). Classification in feature phase



(b). Cell points



(c). Result of cell expansion



(d). Result of cell merging

Fig. 2 Demonstration of recognition process

result of cell searching would include many redundant clusters of cells which are part of background. This result might waste computing time and cause misidentification of object. Hence, if member of reference is not large enough, the reference class would be further merge other similar class, and expand region until the number of cells convergent to one stable constant.

# Cell Expansion and Object Recognition

Cells are considered as expansion center of each object. Considering the geometric constrains, these cells could be further expanded, and fulfill entire distinct object. In view of

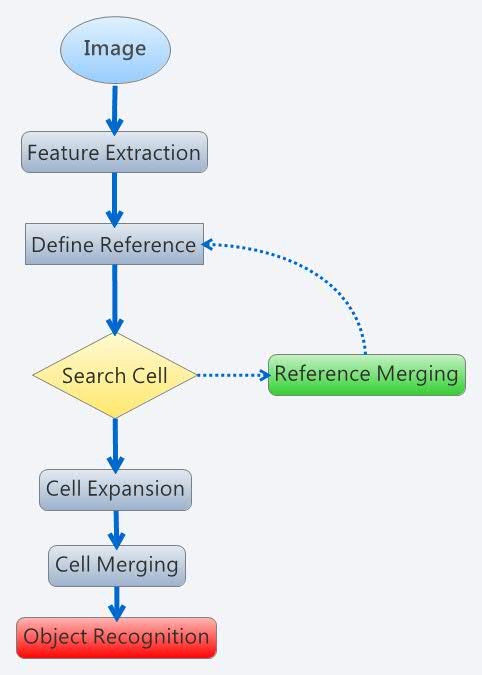
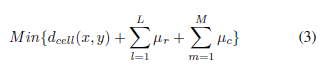


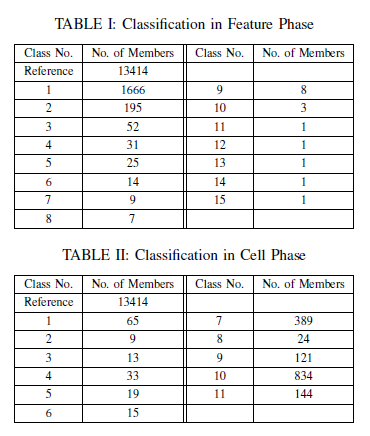
Fig. 3 Flow chart of proposed algorithm

feature space, the feature point is hard to merge into complete object, because one object might include multiple classification of feature. The objects are hard to recognize though feature space only, but object must be distinct in geometric space. Thus, in this chapter, we desire to apply the boundary of geometric become constrains of cell expansion, and, furthermore, combine different feature classes into one object.

First of all, feature clusters in geometric space have to be identified. Boundaries between clusters had been defined by the reference class. While points in same feature class belongs different objects, the cross line between two points must cross over members of reference class and, moreover, cells are considered as basis of objects. Hence the points can be classified by the distance between cells, and become member of closest cell. The distance between points and cell has to apply some additional condition. Each object is distinguished by reference class, and cells are further divided by classification of feature class. Thus, if cross line of point and cell pass through reference class or different feature class, the distance would apply penalty factor to make points can be assigned into corresponding cell.



Where dcell is distance between point (x,y) and each cell. ɲr is penalty factor for path crossing over reference class, and ɲc  is for different class member. All points can be classified though Eq. (3), and assigned into new class which defined by cell. Comparing results of classification by features and cells in Table. 1 and Table. 2 , there are several isolated points in feature classes as class 11 to 15, but, in cell classification, one cell class must include cluster of points. From Fig. 2(c), there are several points are unclassified, because these points are isolated points, or



singular points in image. If one point is surrounded by reference class, or none of cell is same class in feature space, the magnitude of distance would be rapidly increased causing by penalty factor. Thus, noise or singular points would be wiped out in classifying phase.

Although points are further classified into cell phase, one recognized object might include multiple cells due to multiple features in an object. Hence, we would like to evolve classification from cell phase into object. In object phase, each object has one and only one cell.

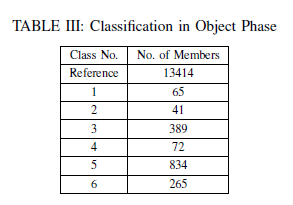
Recalling the identification of cell, cell is the center of feature in each object, and further regard as segmentation of object. Each expansion result is part of one object, so object can be rebuilt through geometric relation of cell expansion result. The classification in object phase can be derived by following criteria:

***C1: If expansion areas of two cells are overlapped, two areas can be merged.***

***C2: If two expansion areas can be connected without reference, two areas and related region can be merged.***

***C3:If expansion area of one cell is segmented by reference class, the result would not be recognized as object.***

According to C1 to C3, the cells are further classified in object phase as Fig. 2(d) and Table 3. Each class in Table 3 represents an object, and member is the points on the object. Through these points, distinct objects can be reconstructed and recognized. The entire proposed algorithm is shown in Fig. 3.

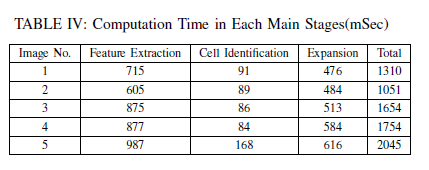


|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
| (a). Test image 1-5 | (b). Result |
| Fig. 4 Result of object recognition | |

# Results of Object Recognition

The proposed method had been exhaustively studied in previous chapters, and following, the method tested by five set of images in Fig.4(a). Image size is 1693\*931, and template of feature extraction is 10\*10, these images include reference and object with different texture, and objects with different geometric structure. Fig.4(b) are results by proposed method. Small white circles means cells at beginning stage, and big red circles are cells after cell expansion and merged. Rectangular region is maximum size of recognized object.

Through these results, objects in images are typically recognized by proposed method. Every single object only exist



one and only cell as mention before. In our experiment, not every cell can be identified easily because of nonhomogeneous lighting or shadow of object. These factors might cause failure to recognize or misidentification of object. Reflection of light in third and fourth row of Fig.4 is kind of possible factor of misidentification. Nevertheless, reflection of light would not be recognized as object by proposed method, because the structure is incomplete and smashed by different classes. As the same way, the left upper corner of fifth row in Fig. 4 is also not identified as object.

Timing has been recorded on a desktop with quad-core i-5 3.20 GHz Processor (without multiple process) running Window 7 64bit. In Table 4, the timing almost depends on number of classes in feature extraction stage, and number of cells. Since fifth images result fifteen cells in image, processing time is longest. Processing time of almost every picture is under two second. The running can be further reduced by using small size image to save time of feature extraction.

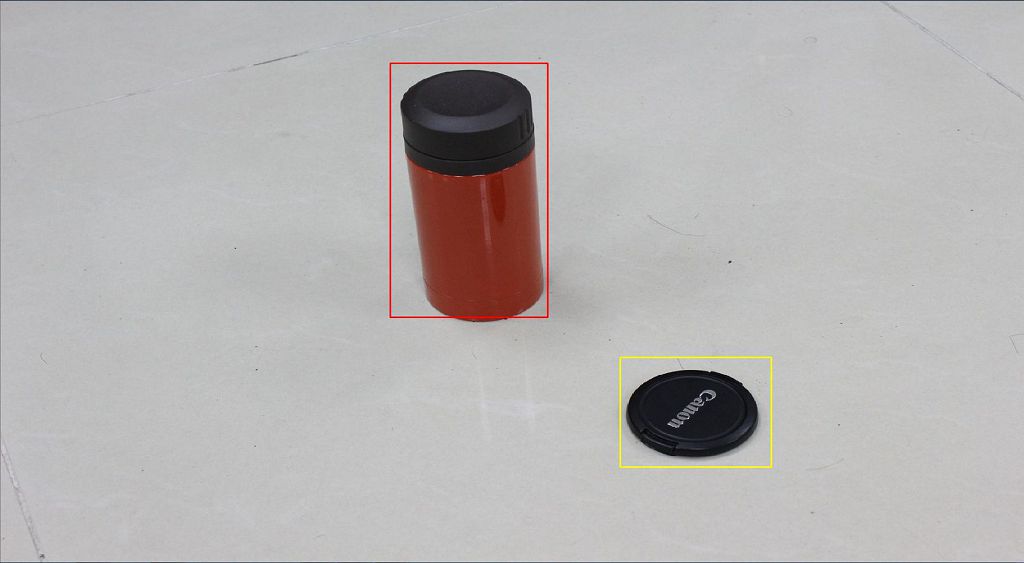
# Hypothesis and Descriptor of Unsupervised Learning

Through the cell expansion algorithm, all objects in this frame are recognized and labeled. And we want to argue, in next frame, whether the detected object will be assigned with the same label if the object is the same one in this frame. Also the algorithm should detect and mark the new object which might not appear in previous frames with a new label.

We analyze the components of an object detected by proposed method. The object must contain all information of merging classes. And each object should map a cell which is the final merging result of expansion. Therefore, a geometric position of the cell belongs to an object could be figured out via the proposed algorithm. We define this kind of cell is object-cell. With a slightly changing background and a subtle moving view, the same object appear in adjacent frames should share the same cell components which are constructing classes in the object. More particularly, the texture features belonging different classes of recognized object would be recorded. Another hypothesis is the position of cells of the same object should be stable if the object is a still-life in the frames. The shifting distance of cells is small enough for the same object in this case. Then the components of the object and the position of cells could be employed as descriptor information. If the objects in adjacent frames are shared same components, and the Euclidean distance between both cells is less than a threshold, objects are considered as the same one. On the other hand, the object which doesn't meet the aforementioned criterions above will be justified as a new object and assigned a new label. In third row of Fig. 5(a), only lens cap is recognizes as an object and labeled with a yellow rectangular. Fig.5(b) is captured by changing the view of camera.



(a). Labeled Object



(b). Recognition of labeled and unknown objects

Fig. 5 Result of descriptor labeling

The proposed method detects both lens cap and the traveling cup. Obviously, a new object invades in this frame. Comparing with the descriptor information, although the cup has advantages in the distance of nucleus, the lens cap in right frame will be recognized based on a very close texture structure. In fact, the cup consists of two texture’s information which belongs to cup cap and body. Hence the cup is labeled with red rectangular, a new mark.

# Conclusion and Future Work

The experimental results presented robust performance of proposed algorithm in variant environments. The region of each distinct object can be recognized precisely through cell expansion and merging. Based on computation result, computing time generally consume by feature extraction and cell expansion stage. Nevertheless, this result is processed on large image, and time consuming of this two stages depend on the number of points. Thus, we convinced this method would be efficiency in general application. Furthermore, we establish a simple descriptor system based on some hypothesis. The descriptor is built for serial frame recognition. The learned object can be recognized and new objects would be labeled by new marks and so on.

Currently, proposed method is focused on unsupervised object recognition. Robust descriptor system has to be further established. This method plain to be a real time learning system, so identify learned object and unknown object would be depending on robust descriptor system. We expect apply relation between cells and classes into descriptor system, because structure of object which includes different feature classes is relatively stable property in object recognition. Furthermore, we are interested in implementing unsupervised recognition algorithms on a robot platform, and implement in more complex environment.

References

[1] M. Burl, M.Weber, and P. Perona, “A probabilistic approach to object recognition using local photometry and global geometry.” In Proc. ECCV,

1998, pp. 628641.

[2] R. Fergus, P. Perona, and A. Zisserman, “Object class recognition by unsupervised scale-invariant learning”, in CVPR, 2003, pp. 264271.

[3] R. Fergus, P. Perona, and A. Zisserman, “A sparse object category model for efcient learning and exhaustive recognition”, in CVPR (1), 2005, pp.

380387.

[4] B. Leibe, A. Leonardis, and B. Schiele, “Combined object categorization and segmentation with an implicit shape model”, in ECCV04 Workshop on Statistical Learning in Computer Vision, Prague, Czech Republic, May

2004, pp.1732

[5] D. J. Crandall and D. P. Huttenlocher, “Weakly supervised learning of part- based spatial models for visual object recognition”, in ECCV (1), 2006, pp. 1629.

[6] M. Brown and D.G Lowe , “Unsupervised 3D Object Recognition and

Reconstruction in Unordered Datasets”, in 3DIM, 2005, pp.56-63.

[7] J. M. Winn and N. Jojic, Locus, “Learning object classes with unsupervised segmentation”, in ICCV, 2005, pp. 756763.

[8] Lexing Xie and P. Perez, “Slightly Supervised Learning of Part-Based Ap- pearance Models”, in CVPR, 2004, pp.107

[9] M. Najjar; C. Ambroise, and J.P. Cocquerez, “Feature selection for semisu- pervised learning applied to image retrieval”, in ICIP,2003,PP.559-62 vol.3

[10] V.V. Chamundeeswari, D. Singh, and K. Singh, “An Analysis of Texture

Measures in PCA-Based Unsupervised Classification of SAR Images”, Geoscience and Remote Sensing Letters, 2009, pp. 214 - 218

1. [↑](#footnote-ref-1)