

Automatic 3D Model Annotation by a Two-Dimensional Hidden Markov Model

Guo Jing*, Zhou Mingquan, Li Chao

College of Information Science and Technology, Northwest University, Xi'an, China

*Corresponding author, e-mail: apalada@sohu.com

Abstract

In this paper, a new method of 3D model automatic annotation is proposed based on a two-dimensional Hidden Markov Model (2-D HMM). Growing importance in the last years Hidden Markov Models are a widely used methodology for sequential data modeling. Recent years, HMMs are applied to research of automatic annotation, such as images and models annotation. The three basic problems with HMM-liked model are also solved in our model. Our modeling process has two steps, those are training and testing. In the proposed approach, each object is separated into several bins by a spiderweb model and a shape function D_2 is computed for each bin. These feature vectors are then arranged in a sequential fashion to compose a sequence vector, which is used to train HMMs. In 2-D HMM, we assume that feature vectors are statistically dependent on an underlying state process which has transition probabilities conditioning the states of two neighboring bins. Thus the dependency of two dimensions is reflected simultaneously. To classify an object, the maximized posteriori probability is calculated by a given model and the observed sequence of an unknown object. Comparing with the general HMM, 2-D HMM gets more information from the neighboring bins. So the system of 2-D HMM performs well on images and model annotation. Analysis and experimental results show that the proposed approach performs better than existing ones in database.

Keywords: 3D model classification, hidden markov model, two-dimensional hidden markov model, expectation maximization (EM) algorithm

Copyright © 2014 Institute of Advanced Engineering and Science. All rights reserved.

1. Introduction

In the past few years, we have observed the availability of technologies for the effective acquisition of digital 3D models of real objects, the establishment of open standards for 3D data interchange, and the increasing use of 3D models in a variety of applications in medicine, engineering, and cultural heritage etc. As a result, many collections of 3D models are created and are available for study and usage, such as Princeton University [1, 2], National Taiwan University [3, 4], Konstanz [5, 6] etc.

To make full use of advantages of existing digital 3D data, classification is usually the first step. Putting the unknown sample into a predefined object class [7, 8] is the basic target of 3D model classification. In some practical applications such as Molecular Biology, Astronomy, Mechanical Engineering, Medical Images etc, we only need to know the label of the object. So classification is very valuable in society and economy.

Generally, people try to description a 3D object by characters. Description of a 3D model includes global and local features. Many algorithms are used to extract features of models. Osada used shape distributions [9] as the global feature to represent 3D shape. Surface curvatures [10] are estimated at a geometry vertex of the 3D object mesh by considering the variations of the surface normal over the platelet of vertex. These algorithms mainly focus on the global features of a 3D model. Other methods represent 3D models by a series of local shape features [11]. These methods which adopt visual features for similarity comparison gradually come to a realization of its limitations. These methods assume that there is an inherent mapping between low-level features and high-level semantics. Now, it becomes clear that the assumption does not hold for many applications. How to narrow down the semantic gap still remains an open issue. Automatic model annotation has emerged as a major approach to bridge the semantic gap.

In this paper, a new method of 3D model automatic classification is proposed based on 2-D HMM. HMM is introduced as a general framework for context dependent classifiers. Our method constructs a statistical structure by a 2-D HMM. The observation of HMM is described by a set of local feature vectors. The sequence is then arranged by a certain fashion that is related to global features. The structure in the classification algorithm takes mixture of global and local features in 3D model as its classification criterion which performs very well in practice. The rest of the paper is organized as follows. Section 2 is previous works on HMM. Section 3 presents generation of the 2-D HMM. Section 4 is outline of our algorithm. Experiment results are shown in section 5. Section 6 is a summary of the full.

2. Previous Works on HMM

Since the theory of hidden Markov models (HMM s) was developed in the 1960s by Baum and Petrie (1966), Baum and Eagon (1967), Baum (1972), HMMs have been widely applied to speech recognition context [12]. And it is only in the last decade that they have been widely used for several other applications, as handwritten character recognition, DNA and protein modeling, gesture recognition, and behavior analysis and synthesis etc.

Recently years, HMM is applied to automatic classification, such as images and models classification. In images classification, previous work extended the 1-D HMM [13], a pseudo 2-D HMM [14, 15] and 2D HMM [16]. To classify images, the samples are divided into blocks, and features of blocks are computed for given a sequence of obvious. [13] presents a spatial-HMM (SHMM) for automatically classifying and annotating natural images. Two generalization of the traditional HMM are trained in the sense that both vertical and horizontal transitions between hidden states are taken into consideration. J.Li et al. [16] proposed a new 2D MHMM to classify images into categories and propagate annotations from keywords which were manually assigned to those categories. One of exist issues of these methods is choosing block sizes. The same problem also exists in 2D HMM. To solve this problem, some methods in signal processing are proposed. Trellis coding [17] in image compression provides an example by using context information. Other works [18, 19] have looked into ways of taking advantage of context information to improve classification performance. Both block sizes and classification rules can vary according to context. The improvement achieved demonstrates the potential of context to help classification.

3. Generation of 2D HMM

In the process of Classification, one HMM is trained for each class of objects. We use a first-order HMM for training. A standard HMM is defined as a state sequence $q = (q_1, q_2, \dots, q_T)$ and observable sequence $o = (o_1, o_2, \dots, o_T)$ generated by the state sequence. Similarly, we consider that observable feature vectors of 3D object's surface points are related to the spatial structure of 3D object. Practically observable feature vectors are observable and the spatial structure of 3D model is unobservable. For example, a car model is constituted by the body and the bottom of the four wheels. In order to model relation between spatial structure and observed feature vector, a set of observations of training objects are used to estimate HMM parameters.

3.1. Definition of Hidden Markov Model

To describe process of producing the pattern, a first order Hidden Markov Model, which statistical structure is depicted by a parameter vector $\lambda = (\pi, A, B)$, is defined as following.

π : The initial state probability distribution, representing probabilities of initial states, that is to

$$\text{say, } \pi_i = P[q_1 = s_i], \quad 1 \leq i \leq N.$$

A : The state transition matrix, $A = \{a_{ij}\}$, where $a_{ij} = P[q_{t+1} = s_j | q_t = s_i], 1 \leq t \leq T$, satisfying

$$\sum_{j=1}^N a_{ij} = 1, 1 \leq i, j \leq N.$$

B : The conditional probability matrix, $B=\{b_j(k)\}$, satisfying $b_j(k) = P[v_k | q_t = s_j]$, where N is the number of states, M is the number of observations.

3.2. Observable Sequence of two-dimensional HMM

The space of 3D object is decomposed by a spiderweb model[9], as illustrated by Figure 1. In Figure 1 starting from positive direction of the first principle axis \vec{U}_1 , each bin is observed sequentially by clockwise to get an observable sequence. The feature vector is extracted in each bin by shape function D2[20]. The feature vector of bin (i, j) is denoted by $o_{i,j}$, where (i, j) represents the j -th bin of the i -th shell. Hence a 3D object corresponds to an observable sequence as $O = (o_{0,0}, o_{0,1}, \dots, o_{i,j}, \dots)$, $i = 0, 1, 2, \dots, w-1$, $j = 0, 1, 2, \dots, j, z-1$.

3.3. Prepared Work on 2D HMM

The 2D HMM assumes that feature vectors are generated by a Markov model that may change state once every bin. The transition probabilities condition on the states of two neighboring bins showed in Figure 2.

Suppose that there are M states $M = \{1, \dots, M\}$, and the state of bin (i, j) is denoted by $s_{i,j}$. The feature vector of bin (i, j) is $o_{i,j}$. Denote $(i', j') < (i, j)$, if $i' < i$ or $i' = i$ and $j' < j$, in which case, we say that the bin (i', j') is before bin (i, j) . In 2D HMM, we made an assumption that

$$P(s_{i,j} | s_{i',j'}, o_{i',j'} : (i', j') \in \Psi) = a_{m,n,l}, \quad \text{where} \quad \Psi = \{(i', j') : i', j' < (i, j)\}$$

$m = s_{i-1,j}, n = s_{i,j-1}, l = s_{i,j}$. In above assumption, we calculate the transition probability of one state by knowing the states of the two adjacent bins in darker shade in figure 2. The state transition of 2D HMM is explained by Figure 3. The second assumption is that the density of the observation o in state s follows a Gaussian distribution. Once the state of a bin is known, the feature vector is conditionally independent of the corresponding features of other bins. For a bin with state s and feature vector o , the distribution has density

$$b_s(o) = \frac{1}{\sqrt{(2\pi)^k |\Sigma_s|}} e^{-\frac{1}{2}(o-\mu_s)^t \Sigma_s^{-1} (o-\mu_s)}$$

, where Σ_s is the covariance matrix, and μ_s is the mean vector.

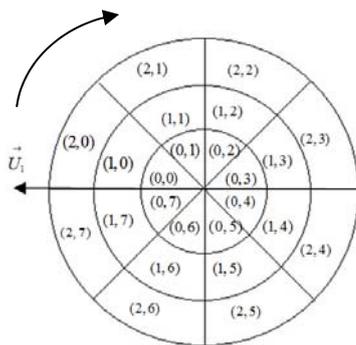


Figure 1. Generation of Observable Sequence

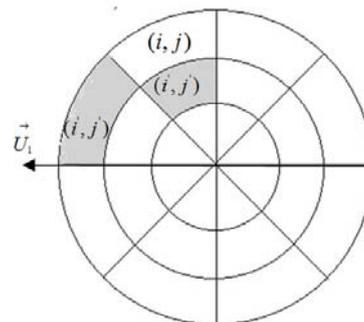


Figure 2. Explanation of Adjacent Bins of Bin (i, j)

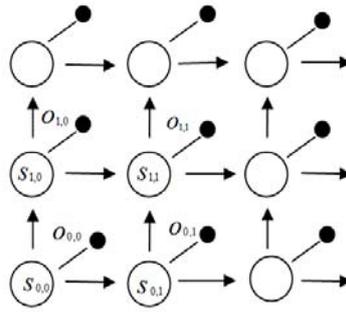


Figure 3. Explanation of State Transition of 2D HMM

4. An Outline of the Algorithm

An outline of our algorithm is as follows.

Step 1 Training

- Divide training objects into spiderweb model and extract feature vector for each bin.
- Select the number of states for HMM.
- Estimate model parameters based on a set of observable sequences of training objects.

Step 2 Testing

- Generate observable sequence (same as step I a) for a testing 3D object.
- Search for the set of classes with maximizing a posteriori probability, given the corresponding sequence of feature vectors to the trained HMM.

4.1. Training

Given a set of observable sequences O , training the model, is performed using the standard Baum-Welch re-estimation procedure to determine the parameter $\lambda = (\pi, A, B)$ that maximizes the probability $P(O|\lambda)$. This method is based on the well-known Expectation Maximization (EM) algorithm [21].

The EM algorithm provides an iterative computation of maximization, when the observed data are incomplete. The term "incomplete" reflects the fact that we need to estimate the distribution x of in sample space \mathcal{X} , but we can only observe x indirectly through s in sample space S . In many cases, there is a mapping $x = s(x)$ from \mathcal{X} to S , and x is only known to lie in a subset of \mathcal{X} , denoted by $\mathcal{X}(s)$, which is determined by the equation $s = s(x)$. We postulate a family of distribution $f(x|\lambda)$, with parameters $\phi \in \Theta$, on x . The distribution of $\mathcal{X}(s)$, $g(s|\lambda)$ can be derived as $g(s|\lambda) = \int_{\mathcal{X}(s)} f(x|\lambda) dx$.

The EM algorithm aims at finding a λ that maximizes $g(s|\lambda)$ given an observed s . Before describing the algorithm, we introduce a function $Q(\lambda'|\lambda) = E(\log f(x|\lambda')|s, \lambda)$ that is the expected value of $\log f(x|\lambda')$ according to the conditional distribution of x given s and parameters λ . The expectation is assumed to exist for all pairs (λ', λ) . In particular, it is assumed that $f(x|\lambda) > 0$ for $\lambda \in \Theta$.

The EM iteration $\lambda^{(p)} \rightarrow \lambda^{(p+1)}$ is defined as follows.

1) E-step: Compute $Q(\lambda|\lambda^{(p)})$.

2) M-step: Choose $\lambda^{(p+1)}$ to be a value of $\lambda \in \Theta$ that maximizes $Q(\lambda|\lambda^{(p)})$.

When using HMMs, a practical but fundamental issue to be addressed is the determination of their structure, namely the topology and the number of states. In this paper, the number of states can be chosen in range of 2 to given limitation 10.

Specifically to say, in E-step, the complete data x are $\{s_{i,j}, o_{i,j}; (i,j) \in \square\}$, and the incomplete data y are $\{c_{i,j}, o_{i,j}; (i,j) \in \square\}$. The function is Equation (1).

$$\begin{aligned} f(x|\lambda') &= P(s|\lambda') \prod P(u|s, \lambda') \\ &= P(s|a'_{m,n,l}; m, n, l \in M) \prod P(u|s, \mu'_m, \Sigma'_m; m \in M) \\ &= \prod_{(i,j) \in \square} a'_{s_{i-1,j}, s_{i,j-1}, s_{i,j}} \cdot \prod_{(i,j) \in \square} P(u_{i,j} | \mu'_{s_{i,j}}, \Sigma'_{s_{i,j}}). \end{aligned} \quad (1)$$

Then we get Equation (2) from (1).

$$\log f(x|\lambda') = \sum_{(i,j) \in \square} \log a'_{s_{i-1,j}, s_{i,j-1}, s_{i,j}} + \sum_{(i,j) \in \square} \log P(o_{i,j} | \mu'_{s_{i,j}}, \Sigma'_{s_{i,j}}) \quad (2)$$

Further, we can make the equation 3 directly by taking expectations on Equation (2).

$$E(\log f(x|\lambda') | y, \lambda^{(p)}) = \frac{1}{\alpha} \sum_s P(s|y, \lambda^{(p)}) \sum_{(i,j) \in \square} \log a'_{s_{i-1,j}, s_{i,j-1}, s_{i,j}} + \frac{1}{\alpha} \sum_s P(s|y, \lambda^{(p)}) \cdot \sum_{(i,j) \in \square} \log P(o_{i,j} | \mu'_{s_{i,j}}, \Sigma'_{s_{i,j}}) \quad (3)$$

In the M-step, we set $\lambda^{(p+1)}$ to the λ' that maximizes (3). In equation (3), two parts can be maximized by choosing corresponding parameters. When maximizing the first part, we define:

$$H_{m,n,l}^{(p)}(i,j) = \sum_s I(m=s_{i-1,j}, n=s_{i,j-1}, l=s_{i,j}) P(s|y, \lambda^{(p)}) \quad (4)$$

As the probability of being in state m at bin $(i-1, j)$, state n at bin $(i, j-1)$, and state l at bin (i, j) , given the observed feature vectors, classes, and model $\lambda^{(p)}$. Equation (5) can be deduced from Equation (4).

$$a'_{m,n,l} = \frac{\sum_{(i,j) \in \square} H_{m,n,l}^{(p)}(i,j)}{\sum_{l=1}^M \sum_{(i,j) \in \square} H_{m,n,l}^{(p)}(i,j)} \quad (5)$$

Next, consider the maximization of the second part in (3). We let $L_m^{(p)}(i,j) = \sum_s I(m=s_{i,j}) P(s|y, \lambda^{(p)})$, which is the probability of being in state m at block (i,j) , given the observed feature vectors, and model $\lambda^{(p)}$. The above expression is then

$\sum_{m=1}^M \sum_{(i,j) \in \square} L_m^{(p)}(i,j) \log P(o_{i,j} | \mu'_m, \Sigma'_m)$. It is known that for Gaussian distributions, the ML estimate

μ'_m of is the sample average of the data, and the ML estimate Σ'_m of is the sample covariance matrix of the data. Since, in our case, the data are weighted by $L_m^{(p)}(i,j)$, the ML estimate of μ'_m and Σ'_m are:

$$\mu'_m = \frac{\sum_{i,j} L_m^{(p)}(i,j) o_{i,j}}{\sum_{i,j} L_m^{(p)}(i,j)}, \quad \Sigma'_m = \frac{\sum_{i,j} L_m^{(p)}(i,j) (o_{i,j} - \mu'_m)(o_{i,j} - \mu'_m)^t}{\sum_{i,j} L_m^{(p)}(i,j)}$$

In summary, the estimation algorithm iteratively improves the model estimation by the following steps.

1. Given the current models $\lambda^{(p)} = (A^{(p)}, B^{(p)}, \pi^{(p)})$.
2. Given the observed feature vectors $o_{i,j}$, the mean vector and covariance matrices are updated by:

$$\mu_m^{(p+1)} = \frac{\sum_{i,j} L_m^{(p)}(i,j) o_{i,j}}{\sum_{i,j} L_m^{(p)}(i,j)}, \quad \Sigma_m^{(p+1)} = \frac{\sum_{i,j} L_m^{(p)}(i,j) (o_{i,j} - \mu_m^{(p+1)}) (o_{i,j} - \mu_m^{(p+1)})^t}{\sum_{i,j} L_m^{(p)}(i,j)}$$

Where,

$$L_m^{(p)}(i,j) = \sum_s I(m = s_{i,j}) \cdot \frac{1}{\alpha} I(C(s) = c) \cdot \prod_{(i,j) \in \mathbb{N}} a_{s_{i-1,j}, s_{i,j-1}, s_{i,j}}^{(p)} \cdot \prod_{(i,j) \in \mathbb{N}} P(u_{i,j} | \mu_{s_{i,j}}^{(p)}, \Sigma_{s_{i,j}}^{(p)}).$$

The transition probabilities are updated by:

$$a_{m,n,l}^{(p+1)} = \frac{\sum_{i,j} H_{M,N,L}^{(p)}(i,j)}{\sum_{l=1}^M \sum_{i,j} H_{M,N,L}^{(p)}(i,j)},$$

Where,

$$H_{M,N,L}^{(p)}(i,j) = \sum_s I(m = s_{i-1,j}, n = s_{i,j-1}, l = s_{i,j}) \cdot \frac{1}{\alpha} I(C(s) = c) \cdot \prod_{(i,j) \in \mathbb{N}} a_{s_{i-1,j}, s_{i,j-1}, s_{i,j}}^{(p)} \cdot \prod_{(i,j) \in \mathbb{N}} P(u_{i,j} | \mu_{s_{i,j}}^{(p)}, \Sigma_{s_{i,j}}^{(p)}).$$

$\mathbb{N} = \{(i,j) : 0 \leq i < w, 0 \leq j < z\}$, refers to all the bins of an object.

3. Repeat step 2, until $\{\lambda^{(p)}\}$ converges to some constant approximately.

4.2. Testing

The classification step is performed by assigning an unknown object to the class of the model showing the maximum likelihood. i.e., assigning an unknown item to the class whose model shows the maximal likelihood $\arg \max [P(o|\lambda)]$. o is an observable sequence of unknown object that is generated using the above method.

$$\begin{aligned} P(o|\lambda) &= \sum_s P(s|\lambda) \cdot P(o|s, \lambda) \\ &= \sum_s P(s | a_{m,n,l} : m, n, l \in M) \cdot P(o | s, \mu_m, \Sigma_m : m \in M) \\ &= \sum_s \int_s \prod_{(i,j) \in \mathbb{N}} a_{s_{i-1,j}, s_{i,j-1}, s_{i,j}} \cdot \prod_{(i,j) \in \mathbb{N}} P(u_{i,j} | \mu_{s_{i,j}}, \Sigma_{s_{i,j}}) \end{aligned}$$

5. Experiment Results

Our experiments are based on the Princeton Shape Benchmark database (2005). This public database has been largely used in object recognition literature. It contains 1814 3D models which have been split into a training database and a test database. We use models of training database to train HMM's parameters and the others of testing database to test. We compare our method with adaptive weighted asymmetric (AdaBoost) hidden Markov models (ADHMM) and Shape Distribution (SD) proposed in references [20, 22] and result show that our method performs better.

We tested our approach by varying the free parameters of the techniques. The first experiment demonstrates that the accuracy of changes under different parameters, i.e., the number of bins L , and the number of states of HMMs n . We set $L = w \times z$, where w is number of

circles, z is number of bins in each circle. We do not want to choose a large size since this obviously entails crude classification. But if we choose a small size, only very local properties belonging to the small bin are examined in classification. Some information about surrounding bins is neglected. We set the number of bins from 16 to 80.

We choose 100 models from 20 categories of training database, which may include some categories such as human, sports_car, fighter_jet, etc. In each category, we use 5 models to train HMMs and the rest to test. From the Table 1, we can see that the system performs unsatisfactorily when $L = 16$. Clearly, the accuracy increases gradually with the number of bins L increases. Actually, even if we set $n = 4$, the system is able to recognize the objects with an accuracy larger than 75%. In particular, the accuracy can arrive at 98% with $L=80, n=6$.

The second experiment is to investigate the accuracy of three methods. In this stage, we select 100 objects of 25 categories (each category contains at least 6 objects) from database. In each category 5 objects are used for training, the others are used for testing. The results are then computed using the best configuration of parameters derived from the previous analysis, i.e. using $L = 10 \times 8, n = 6$. The performances of the three methods are illustrated in Figure 4. It is clear that our method performs better than the other ones.

The last experiment is to demonstrate our technique is invariant to model rotation and transformation. And then 15 different models are selected from the database. For each model, we perform 2 scaling (scale factors: 0.5 and 1.5 respectively), 3 rotations (90 degrees around x-, y-, z- axis respectively). So we get more 5 new model files of each model and put them together with the original models. In this way, a small test database with 90 models of 15 categories is generated. We then select randomly models from the test database to test. Classification accuracies are proposed in Table 2. From the figure, we can see that the system is very robust and both methods are insensitive to the model's rotation and transformation, and produce comparable results.

Table 1. The Accuracy of Different Parameters

$L=$ $n=$	2×8	4×8	6×8	8×8	10×8
2	0.7172	0.7256	0.7561	0.7608	0.7813
3	0.7374	0.7481	0.7590	0.7747	0.7810
4	0.7690	0.7863	0.8002	0.8012	0.8142
5	0.7898	0.7978	0.8149	0.8363	0.8891
6	0.8182	0.8381	0.89397	0.9412	0.9815
7	0.8444	0.8491	0.8528	0.9434	0.9300
8	0.8576	0.8580	0.8723	0.9014	0.9079
9	0.8662	0.8508	0.8966	0.8827	0.8967
10	0.8800	0.8817	0.7699	0.8523	0.8815

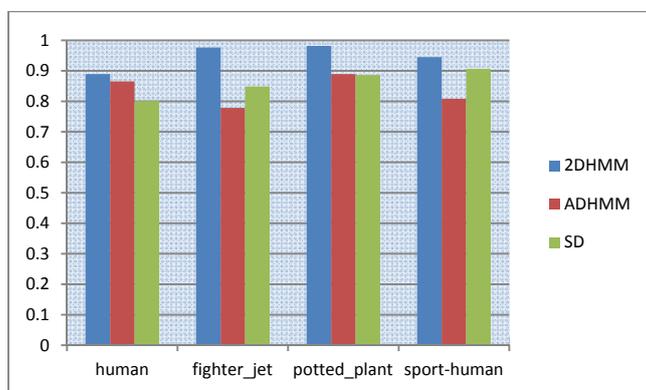


Figure 4. The Average Accuracy of the Three Methods

Table 2. The Result of the Third Experiment

	2DHMM	ADHMM	SD
bed	91.4	80.6	83.0
Tree	88.7	80.8	76.5
bicycle	90.0	85.5	79.6
tank	88.3	79.3	80.0
track	92.5	66.0	87.1

6. Conclusion

In this paper, a new method for automatic 3D object Annotation has been proposed based on the two-dimensional Hidden Markov Model approach. Specifically, each model is separated into several bins by spiderweb model. For each bin, the feature of D2 is computed. The sequences of vectors (one for each bin) are subsequently modeled using HMMs, paying particular attention to the initialization and the model selection issues. Classification is carried out by using a nearest neighbor rule, where distance is computed using the HMM likelihood function. A thorough experimental evaluation has shown that the proposed approach is very promising for classifying 3D objects from large database. Furthermore, the proposed method remains quite accurate even in case of model of transformation, scale and rotation.

An interesting extension of the method could go toward the investigation of the use of the 3D model segmentation. In particular, we are investigating the separate 3D model into meaningful parts by HMMs.

Acknowledgements

The work described in this paper was supported by the National Science Foundation of China (Grant No. 60736008) and Northwest University National Science Foundation (Grant No. ND0936, No. JD10316).

References

- [1] 3D model search engine, <http://shape.cs.princeton.edu>.
- [2] P Min, JA Halderman, M Kazhdan, TA Funkhouser. *Early experiences with a 3D model search engine*. Web 3D Symposium. France. 2003; 7–18.
- [3] 3D model retrieval system, <http://3d.csie.ntu.edu.tw/~dynamic/>.
- [4] DY Chen, XP Tian, YT Shen, M Ouhyoung. *On visual similarity based 3D model retrieval*. Computer Graphics Forum (EG 2003 Proceedings). Spain. 2003; 22(3): 223-232.
- [5] 3D model similarity search engine, <http://merkur01.inf.uni-konstanz.de/CCCC/>.
- [6] DV Vranić. *An improvement of rotation invariant 3D shape descriptor based on functions on concentric spheres*. Image Processing. In ICIP 2003, 2003; 2: III - 757-60.
- [7] Csakany P, Wallace AM. Representation and classification of 3D objects. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*. 2003; 33(4): 638-647.
- [8] Huber D, Kapuria A, Donamukkala RR. *Parts-based 3D object classification*. Computer Vision and Pattern Recognition, 2004. CVPR 2004. Washington, DC, USA. 2004; 2: 82-89.
- [9] Mihael Ankerst, Gabi Kastenmüller, Hans-Peter Kriegel, Thomas Seidl. *3D Shape Histograms for Similarity Search and Classification in Spatial Databases*. 6th International Symposium, SSD'99. Hong Kong, China. 1999; 207-226.
- [10] Vandeborre JP. *A practical approach for 3D model indexing by combining local and global invariants*. 3D Data Processing Visualization and Transmission, 2002 Processing. Italy. 2002; 644-647.
- [11] Liu yi, Wang xulei, Zha hong bin. 3D Partial Shape Retrieval Based on Local Bag of Words Models. *Acta Scientiarum Naturalium Universitatis Pekinensis*. 2009; 45(6): 967-968.
- [12] Lawrence R Rabiner. *A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition*. Proceedings of the IEEE. 1989; 77(2): 257–286.
- [13] Feiyang Yu, Horace HS. *Automatic Semantic Annotation of Images using Spatial Hidden Markov Model*. IEEE International Conference on Multimedia and Expo. Canada. 2006; 305-308.
- [14] SS Kuo, OE Agazzi. *Machine vision for keyword spotting using pseudo 2D hidden Markov models*. Acoustics, Speech, and Signal Processing. Minneapolis. 1993; 5: 81–84.
- [15] CC Yen, SS Kuo. *Degraded documents recognition using pseudo 2D hidden Markov models in Gray-scale images*. SPIE Proceedings. 1994; 2277: 180–191.
- [16] Jia Li, Amir Najmi, Robert M Gray. *Image Classification by a Two-Dimensional Hidden Markov Model*. Acoustics, Speech, and Signal Processing. 1999; 6: 3313-3316.

-
- [17] A Gersho, RM Gray. *Vector Quantization and Signal Compression*. Boston, MA: Kluwer. 1992: 623.
- [18] CH Fosgate, H Krim, WW Irving, WC Karl, AS Willsky. Multiscale segmentation and anomaly enhancement of SAR imagery. *Image Processing*. 1997; 6(1): 7-20.
- [19] J Li, RM Gray. *Context based multiscale classification of images*. ICIP 98. Chicago. 1998; 3: 566-570.
- [20] R. Osada et al. Shape Distributions. *ACM Transactions on Graphics*. 2002; 21(4): 811-813.
- [21] Yujian Li et al. *Hidden Markov models with states depending on observations*. *Pattern Recognition Letters*. 2005; 26 (7): 977–984.
- [22] Liu xiaoming, Yin jian wei, Feng zhilin, Dong Jinxiang. 3D model classification based on adaptive weighted asymmetric AdaBoost hidden Markov models. *Journal of Zhejiang University (Engineering Science)*. 2006; 40(8): 1300-1305.