Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics
San Antonio, TX, USA - October 2009

# Classification of 3D Models for the 3D Animation Environments

He Yuesheng
Computer Science Department
Hong Kong Baptist University
Hong Kong
yshe@comp.hkbu.edu.hk

Yuan Yan Tang
Computer Science Department
Hong Kong Baptist University
Hong Kong
yytang@comp.hkbu.edu.hk

***Abstract*—Classification the 3D models in the graphical environment is a key problem with applications in computer graphics, virtual reality, especially the intelligent virtual human for Humanoid Animation. The challenging aspects of this problem are to find a suitable shapes' feature that can be used to compare them quickly and a proper method to classify the models on the features. We propose a method of classifying shapes' features for surface-based 3D shape models based on their shape similarity. The features of shape of 3D models are computed by first converting an input surface based model into an oriented point set model and then computing features histograms of distance and Shape Distributions. Then, Support Vector Machines (SVM) are used to classify the features of the models. By the classification the models can be given semantic meanings in the 3D environment.**

***Index Terms*—Computer Graphics, Virtual Human, Animation, Shape, 3D Model, SVM.**

## I. INTRODUCTION

There has been an explosion in the use of animated humanoid avatars in the entertainment industry and in simulations for training and education, but these efforts have been hindered by the highly technical nature of composing meaningful humanoid behaviors and by the ability to interact between the 3D graphical environment, as well as by the difficulty in generating realistic human actions which are reusable across applications[11][13]. Since the human avatars always have many degree of freedom, the search for the actions' space is exponential. However, if the human avatar has the ability to recognize the environment, it is possible to reduce the computational complexity to linear. Thus, recognition of 3D models is a key issue of this area. If the 3D models can be recognized by the system, then they can be given topological and/or semantic meaning and easy to be used and stored, for instance, interacted with 3D human-like animation object(virtual Human) [18][12]. Moreover,proliferation of 3D models on the Internet and in in-house databases prompted development of the technology for effective recognition of three-dimensional (3D) models.

A 3D model could be described by its textual annotation by using a conventional text-based search engine. This approach wouldnt work in many of the application scenarios for the 3D shape models. However,the annotations added by human beings depend on different applications and other factors. therefore,it is extremely difficult to describe by words a shape that is not in a well known shape or semantic category. It is thus necessary to develop content-based recognition systems for 3D models that are based on the features intrinsic to the 3D models, one of the most important of which is the shape.

In the study of the shape similarity recognition of 3D models, first step is to extract robust, concise, yet expressive shape features, and on the development of similarity (or, dissimilarity) comparison methods that conform well to the human notion of shape similarity[1][5][7].

In developing the shape features for 3D models, we first have to decide which class of 3D shape representation we are targeting. A 3D shape may be defined by using any of a number of shape representations, many of which are not mutually compatible. Some of the shape representations are mathematically well founded, allowing for computations of such well-defined properties as volume, surface curvature, or surface (or volume) topology. Unfortunately, since most 3D file formats (VRML, 3D Studio, etc.) have been designed for visualization, they contain only geometric and appearance attributes, and usually lack semantic information to be recognized. A great part of shape representations are less nicer. For example, a polygon-soup model is a topologically disconnected collection of independent polygons and/or polygonal meshes[1][2]. Neither volume nor surface curvature can be computed for the model.

The second step is to classify the features of shapes been extracted from the 3D models. Because the shapes seldom have any topology or solid model information; they rarely are manifold; and most are not even self-consistent, it is important to separate them on their features.

In this paper, we proposed a method for computing 3D shape signatures and dissimilarity measures for arbitrary objects described by possibly degenerate 3D polygonal models. The object is to represent the signature of an 3D model on measuring global geometric properties of the object. We present methods to describe the features of models and reduce the dimensionality of them to use them in the 3D animation system.

Shapes of models are the objects we used in the methods. Thus,we used a pair of methods to represent features of shapes - 1-dimensional descriptor (e.g. Osada, et al.)[1] and 2-dimensional descriptor (e.g. mutual Absolute-Angle Distance histogram(AAD)[2].

The 3D graphical environments are often extremely large and complex.The whole scene contains a great number of

978-1-4244-2794-9/09/$25.00 ©2009 IEEE

3D models. Therefore it is a requirement of a large storage and computational resource to deal with their shape descriptors(often very high dimensional).In order to reduce the features' dimensionality and obtain better online recognizing performance, we use DCT transformation to transform the feature vectors. By this method, the dimension has been reduced without influencing the effect of recognition.

Intuitively, a given set of feature points which belongs to either one of two classes can be linearly separated by SVM with the hyperplane leaving the largest possible fraction of points of the same class on the same side. Since the model features are vectors representing different shapes.Then we classify the models using SVM kernels.

The paper is organized as follows. In the next section, we introduce the ways to extract features of shapes and reduce the dimensionality. The classification method is described in Section 3, and the method and results for the experimental evaluation of our algorithm are presented in Section 4. We conclude the paper in Section 5.

## II. Features of Shapes

A method for shape similarity comparison of 3D models can be classified by the shape representation it is targeting. Some of the shape comparison algorithms assume well-defined shape representation, that are, 3D solid represented by using voxels, boundary representation, or constructive solid geometry . Others assume topologically well-defined 2-manifold surfaces . However these methods cant be used to compare polygon soup models. In this section, we review shape similarity comparison methods for not-so-well-defined shape representations, especially those for polygon-soup models. Another possible classification is by the method used to achieve invariance of the shape comparison method to a class of geometrical transformations[3].

Osada et al. proposed what they call shape distributions. Osadas shape distributions, a set of shape features, have the advantage of being invariant, without pose normalization, to similarity transformations[1]. Moreover, they are designed to be applicable to a not-so-well-defined mesh-based model, i.e., a polygon soup defining a non-solid object consisting of non-manifold surfaces, multiple connected components, and such degenerate surfaces as zero-area polygons. D2 is one of the distributions.

We choose this distribution, because it is simple and computational efficient.

The 2-dimensional feature descriptor uses vectors to represent shapes. For instance, the mutual Absolute-Angle Distance histogram (AAD) shape feature is a 2D histogram. This method considers not only the distance, but the the orientation of vectors as well[4].

In the applications of 3D animations, the high dimensionality of the feature descriptors restricts the computational complexity of recognizing process. Such complexity determine the size and kind of environment we can deal with in real time. Thus, a optimal way of dimensionality reduction is to be used.

The process of obtaining the features is:

1) Calculating the items of features (e.g. L1-norm, L2-norm distance and inner product);
2) Normalization by a certain criteria (e.g. By maximum,by average or by median);
3) Dimensionality reducing by statistically optimal way;
4) Making histogram.

### A. 1-Dimensional Descriptor

The first issue of the method is to select a function whose distribution provides a good signature for the shape of a 3D polygonal model. Ideally, the distribution should be invariant under similarity transformations, and it should be insensitive to noise, cracks, tessellation, and insertion/removal of small polygons. In general, any function could be sampled to form a shape distribution, including ones that incorporate domain-specific knowledge, visibility information (e.g., the distance between random but mutually visible points), and/or surface attributes (e.g., color, texture coordinates, normals and curvature). However, as our interesting is on the topology information of the models,we use the features based on geometric measurements (e.g., angles, distances, areas, and volumes).

Shape features should be independent of the representation, topology, or application domain of the sampled 3D models. As a result, the shape similarity method can be applied equally well to databases with 3D models stored as polygon soup, meshes, constructive solid geometry, voxels, or any other geometric representation as long as a suitable shape function can be computed from each representation.

One of the important requirement for a 3D shape similarity comparison method is invariance of the method to a required class of geometrical transformations. Most of the time, an invariance to similarity transformation, that is, a combination of translation, rotation, and uniform scaling, is required for a 3D shape similarity comparison.

The Osada, et al.[1] is the set of shape distributions which has been used to represent the geometric properties.

The D2 definition is:

- ***D2***: Measures the distance between two random points on the surface.

Figure 1 shows two original 3D models which are going to be made the feature descriptors.

We use the L2-norm to measure the distance between the points of the surface:

$$d = \sqrt{(p_i - p_j)^2}$$

The following step is normalizing the result by by maximum distance. Thus the results are between 0 and 1.

Then the D2 shape histogram has been made. Figure 2 shows the different feature histograms according to the models.

To achieve geometrical transformation invariance, all the methods employ step of normalization.

### B. 2-Dimensional Descriptor

The 2-Dimensional Descriptor is 2D histogram of distances and angles formed by pairs of oriented points that are generated on the surfaces of the given 3D shape model.

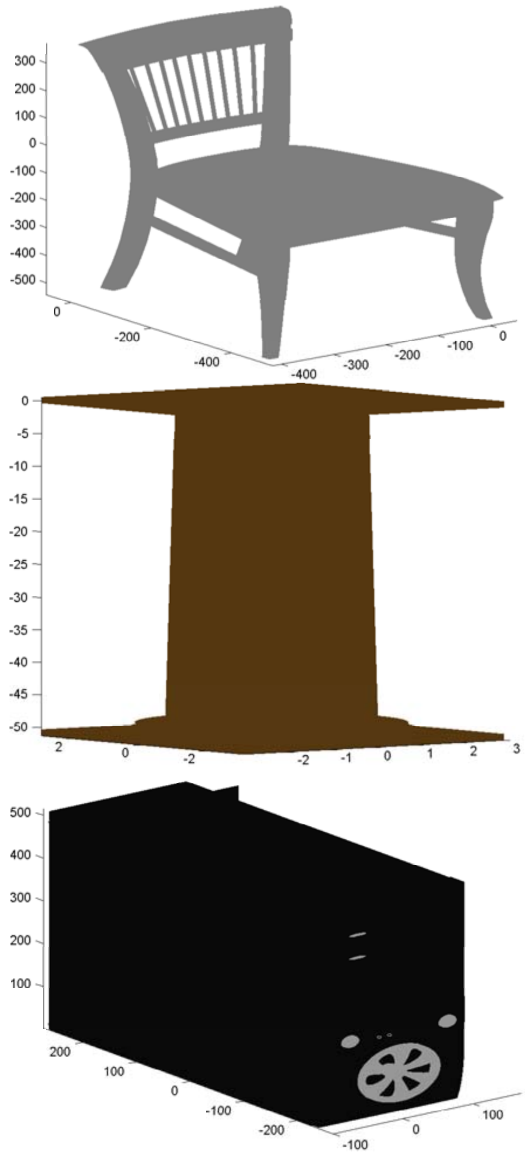

Fig. 1. The Three Different 3D Models - The first one is a chair, second one is a Column and third is a case.

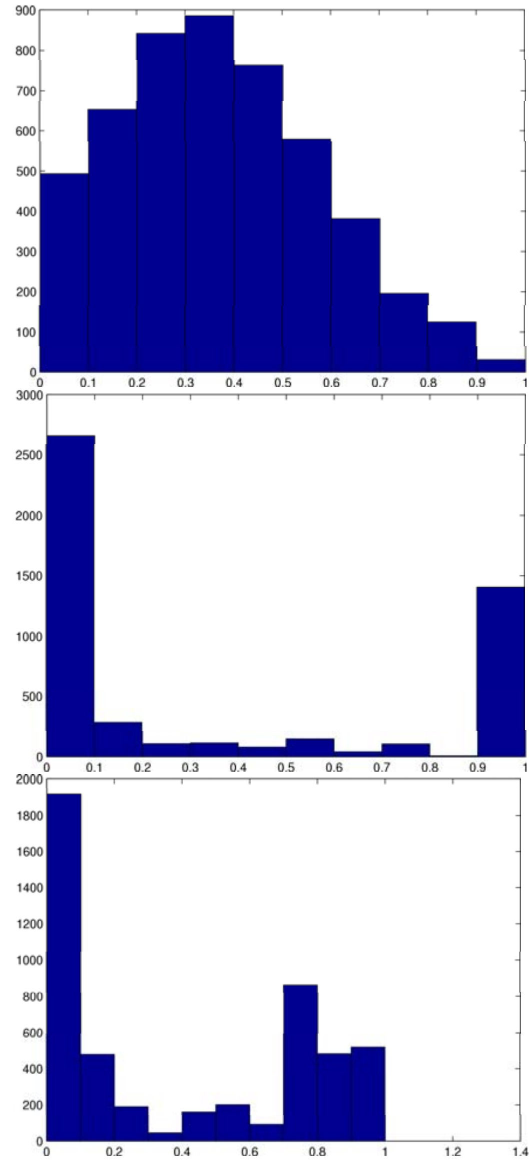

Fig. 2. The Three Different 3D Models' D2 Shape Histograms - The first one is the chair's, second one is the Column's and third is the case's.

We use the AAD [2][8] to form the histogram.

The AAD method is based on AD[2]. The AD shape feature measures, for each pair of points $p_1$ and $p_2$ , the 3D Euclidian distance $d = \sqrt{(p_1 - p_2)^2}$ between the points and the inner product $a =< n_1, n_2 >$ of the orientation vectors $n_1$ and $n_2$ of the points. The AD shape feature described above is sensitive to the sign of the orientation vector of the point set model. If the models to be compared have a consistent surface orientation, e.g. a consistent traversal order of the vertices among polygons, the AD shape feature performs well. If, however, the database contains models having surfaces that are inconsistently oriented, AD shape feature suffers.

The mutual Absolute Angle and Distance (AAD) is computed similarly to the AD, except that the AAD ignores the sign of the inner product. This makes the AAD a more robust shape feature than the AD for the models having unoriented or inconsistently oriented surface orientations.

Its definition is:

- ***AAD***: 2D histogram of distances and angles formed by pairs of oriented points that are generated on the surfaces of the given 3D shape model.

The AAD has properties:

- orientation insensitive shape feature;
- normalization prior to applying a pose orientation sensi-

tive shape feature.

Models of Figure 1 generated the different shape descriptors of AAD Figure 3. The mutual Absolute Angle and Distance (AAD) histogram is computed similarly to the AD, except that the AAD ignores the sign of the inner product. This makes the AAD a more robust shape feature than the AD for the models having unoriented or inconsistently oriented surface orientations.

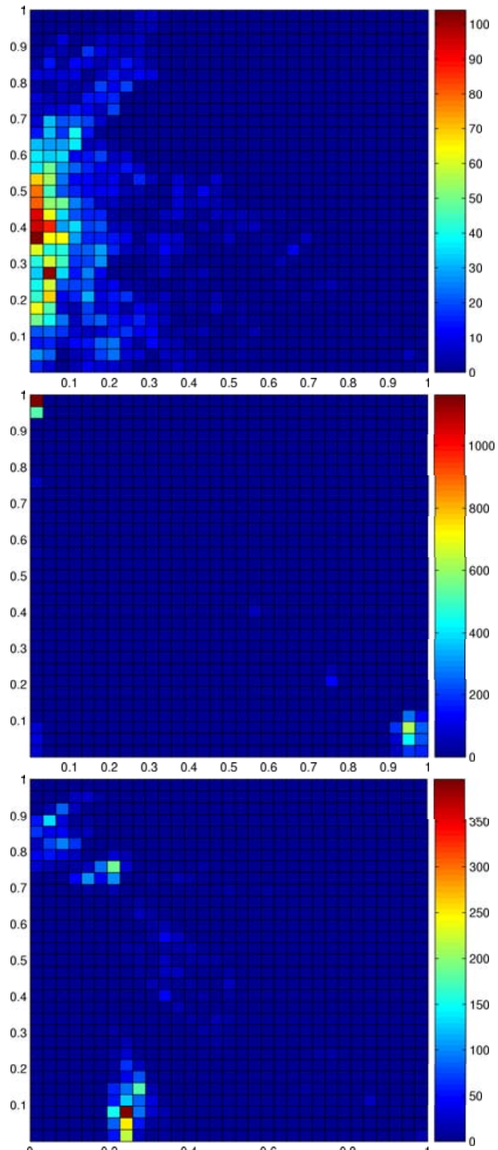


Fig. 3. The 3D Models' AAD Shape Histograms - The first one is the chair's, second one is the Column's and third is the case's.

## III. Classification and Recognition

For we use the referenced 3D models to instruct the work of classification,intuitively, SVM[9] works well in the problem. We use it to recognize the models following two ideas:

1) The feature descriptors of shapes are not always linearly separatable. However,an SVM captures the query concept by separating the relevant models from the irrelevant models with a hyperplane in a projected space, which usually is a very high dimensional one, where it has ability to separate them.
2) Once the classifier has been trained, SVM can use it to calculate and return the result in a very short time.

The SVM [9] methodology comes form the application of statistical learning theory to separating hyperplanes for binary classification problems. The central idea of SVM is to adjust a discriminating function so that it makes optimal use of the separability information of boundary cases. Given a set of cases which belong to one of two classes, training a linear SVM consists in searching for the hyperplane that leaves the largest number of cases of the same class on the same side, while maximizing the distance of both classes from the hyperplane. If the training set is linearly separable, then a separating hyperplane, defined by a normal w and a bias b, will satisfy the inequalities:

$$y_i(w \cdot x_i + b) \geq 1 \forall i \in \{1 \cdots N\} \tag{1}$$

where $x_i \in \Re^d$is a case of the training set $i = (1, \cdots, N)$, and d being the dimension of the input space, and $y_i \in \{-1, 1\}$ is the corresponding class.Since such a distance is $\frac{1}{||w||}$ , finding the optimal hyperplane is equivalent to minimizing $||w||^2$ under constraints (1).

The dual problem is:

$$\max \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j < x_i x_j > \tag{2}$$

under the constraint:

$$\sum_{i=1}^{N} \alpha_i y_i = 0 \quad \text{where} \quad 0 \leq \alpha_i \leq C \quad \forall i \in \{1 \cdots N\} \tag{3}$$

The SVM approach can be extended to non-linear decision surfaces through a non-linear function $\Phi$ which maps the original feature space $\Re^d$ a higher dimensional space H. Since the only operation needed on H is the inner product, if we have a kernel function k [10]:

$$k(x', x'') = \Phi(x') \cdot \Phi(x'') \tag{4}$$

The objective function becomes:

$$\max \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j K(x_i, x_j) \tag{5}$$

When the function $K$ satisfies Mercers condition[9], we use it to compute the inner product of $\Phi(x)$s. Thus, by using $K$ we are implicitly projecting the training data into a different (often higher dimensional) feature space. Some typical $K$s we used in this problems are:

The RBF kernel:

$$k(x', x'') = exp(-\frac{||x' - x''||^2}{\delta^2}) \tag{6}$$

where $\delta$ (a positive real) are parameters of the kernel.

The Poly kernel:

$$k(x', x'') = (x' \cdot x'' + 1)^p \tag{7}$$

We applied different SVM-based classification strategies and feature sets to semantically classify regions in graphical 3D models.

During training procedure, we compute a large amount of surface Points features with various combinations from parameter space. We apply different kernels to select the most discriminating features. SVM kernel parameters are determined by observer the result of the procedure. All feature settings and parameters determined on the training-set are then applied for classification and recognition tasks on the test-set. Due to the histogram representation of the features, the usage of a Histogram-Intersection kernel implying a L2-norm distance measure is straight forward and empirically delivered the best results.

## IV. Experimental Result

We used the 3D studio models from World Wide Web (randomly search and download) to construct the training and testing sets. The examples are shows in Figure 4

Fig. 4. Different kinds of 3D Models for the experiment - all models from world wide web.

Then, we built the 3D shape features for every models.

One issue we must be concerned with is sampling density. The more samples we take, the more accurately and precisely we can reconstruct the shape distribution. On the other hand, the time to sample a shape distribution is linearly proportional to the number of samples, so there is an accuracy/time tradeoff in the choice of N.

We compared the performance of the AAD and the D2 shape features by using the models. The parameters used for this experiment are as follows:

1) *1-D Descriptor*: The number of points per model $P_n = 256$. Distance is computed by using the L2 norm. The normalization is performed by using the maximum-based method.

| $Kernels$ | $D2shape$ | $AADshape$ |
|---|---|---|
| Gaussian | 52% | 55% |
| Gaussianslow | 53% | 56% |
| Multiquadric | 52% | 54% |
| Poly | 71% | 77% |

TABLE I
RESULT OF CLASSIFICATION

| $PolyKernel$ | $D2shape$ | $AADshape$ |
|---|---|---|
| p=1 | 58% | 64% |
| p=2 | 70% | 72% |
| p=3 | 71% | 77% |
| p=4 | 64% | 68% |

TABLE II
COMPARISON OF RESULT OF CLASSIFICATION

2) *2-D Descriptor*: The number of points per model $P_n = 128$.Distance is computed by using the L2 norm-based method The normalization is performed by the maximum-based method.

Their shape histograms are constructed as follows:

1) *1-D Descriptor*: Bins numbers $B_d = 256$;
2) *2-D Descriptor*: Bins numbers $B = D \times I$ (distance D = 32, inner product I = 32).

We further investigated the robustness of our method by testing it with different polygon tessellations of two 3D shapes. Thus, we test the different positions in the environment of the models and rotation of the models. Those did not influence the basic property of every feature descriptors.

To classify and recognize the certain features, we used the SVM. For instance, the object of the work is to recognize more chair-like models from others. We first trained the machine on the training set, then used it to classify the testing set.

Although the results are quite better than those of random guessing, they are still far from satisfactory. Moreover, the 2-Dimensional feature always performs better than 1-Dimensional one. However, the Ploy Kernel achieved more accurate rate of recognition than others. The Table I shows the result.

Otherwise, when we have compared the ploy kernel with different parameters to classify the result is as good as the original ones, the The Table II shows the result.

## V. Conclusion and Discussion

In this paper, we proposed and evaluated a method of classification and recognition of shape features for shape similarity search of 3D models. The shape features, which have been represented by 1-Dimensional and 2-Dimensional descriptors, are robust against topological and geometrical irregularities and degeneracies, which make them applicable to 3D Studio format and other so called polygon soup models. They are also invariant to similarity transformation, a quality valuable in classifying 3D shape models.

According to the experiments, though the 2-D one has computational cost somewhat higher (have to compute inner product) than the D2, they significantly outperformed D2 in our classification experiments by SVM. Although a further comparison has not been made, the 2-Dimensional descriptor might have the better performance than that of the 1-Dimensional methods, such as the Table I shows. However, the computational costs of former one is higher than the later one. Thus, both the distance and angular information are useful for recognizing 3D shapes.

As a future work, we would like to improve our shape feature, for example by adding some form of multi-resolution approach to matching 3D shapes. We also would like to explore a hybrid shape feature that combines, possibly adaptively, shape features having different characteristics.

Depend on the shape features, the SVM has been used to train the system to classify different shapes. According to the experimental result, the poly kernel perform well in the task. In future, we would like to explore different kind of kernels and parameters of them.

## REFERENCES


[1] Robert Osada, Thomas Funkhouser, Bernard Chazelle, and David Dobkin, *ACM Transactions on Graphics*, Vol. 21, No. 4, October 2002, Page 807—832.

[2] Ryutarou Ohbuchi, Takahiro Minamitani, Tsuyoshi Takei, *International Journal of Computer Applications in Technology*,Vol. 23, No. 2/3/4, 2005, Page 70—85.

[3] J. Fehr, H. Burkhardt, Harmonic Shape Histograms for 3D Shape Classification and Retrieval, *MVA2007 IAPR Conference on Machine Vision Applications*, May 16-18, 2007, Tokyo, JAPAN.

[4] Ding-Yun Chen, Xiao-Pei Tian, Yu-Te Shen and Ming Ouhyoung, Visual Similarity Based 3D Model Retrieval,*EUROGRAPHICS 2003 / P. Brunet and D. Fellner (Guest Editors)* Volume 22 (2003), Number 3,Page 223—232.

[5] FRED L. BOOKSTEIN, Principal Warps: Thin-Plate Splines and the Decomposition of Deformations, *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE.* Vol.II., NO. 6, JUNE 1989, page 576—585.

[6] Edward Grabczewski, John Cosmas, Peter Van Santen Damian Green, Takebumi Itagaki, Fred Weimer,3D MURALE: Multimedia Database System Architecture,*Association for Computing Machinery*, 2002

[7] Michael M. Bronstein and Ron Kimmel, Topology-invariant similarity of nonrigid shapes Alexander M. Bronstein, June 10, 2008.

[8] Ryutarou Ohbuchi, Akihiro Yamamoto and Jun Kobayash, Learning Semantic Categories for 3D Model Retrieval, *9th ACM SIGMM International Workshop on Multimedia Information Retrieval (ACM MIR 2007).*

[9] Vladimir Naumovich Vapnik,*The Nature of Statistical Learning Theory.*

[10] Dmitry Zelenko, DMITRY Chinatsu, Aone CHINATSU, Anthony Richardella, Kernel Methods for Relation Extraction, *Journal of Machine Learning Research*, FEB 2003, Page 1083—1106.

[11] Norman I.Badler, Cary B.Phillips, Bonnie Lynn Webber,*Simulating Humans: Computer Graphics Animation and Control*, 1993.

[12] T. Conde, D. Thalmann, An Integrated Perception for Autonomous Virtual Agents: Active and Predictive Perception, Computer Animation and Virtual Worlds, Volume 17, Issue 3-4, John Wiley, 2006

[13] Moccozet L, Thalmann N. M., Dirichlet Free-Form Deformation and their Application to Hand Simulation[J], *Proceedings Computer Animation97*, IEEE Computer Society, 1997, Page 93—102.

[14] Catherine Zanbaka1, Amy Ulinski, Paula Goolkasian, Larry F. Hodges, Social Responses to Virtual Humans: Implications for Future Interface Design, *CHI 2007 Proceedings*, Page 1561-1570.

[15] Edward M. Sims, Reusable, lifelike virtual humans for mentoring and role-playing, *Computers & Education*, 49 2007, Page 75-92.

[16] Lucio Ieronutti , Luca Chittaro, Employing virtual humans for education and training in X3D/VRML worlds, *Computers & Education*, Page 93-109.

[17] Alain Rakotomamonjy,Variable Selection Using SVM-based Criteria,*Journal of Machine Learning Research*, March 2003, Page 1357—1370

[18] Weixiong Zhang, Randall W. Hill, Jr., A Template-Based and Pattern-Driven Approach to Situation Awareness and Assessment in Virtual Humans, *Agents*, 2000, Page 116—123.

[19] Z.M. , D. Reidsma, A. Nijholt, Human Computing, Virtual Humans and Artificial Imperfection, *ICMI'06*, Page 179—184.