Emotion Expression 3-D Synthesis from Predicted Emotion Magnitudes

Ricardo A. Calix

Purdue University Calumet, Gyte Bldg., Room 251, 2200 169th Street, Hammond, IN, 46323-2094
ricardo.calix@purduecal.edu

Abstract

Many studies have been conducted on how to detect emotion classes or magnitudes from multimedia information such as text, audio, and images. However, the methods that can use predicted emotion classes and magnitudes to render emotion expressions in Embodied Conversational Agents (ECA) are still unclear. This paper proposes a computer graphics methodology that uses predicted non-linear regression values to render facial expressions using mesh morphing techniques. Results of the rendering technique are presented and discussed.

Introduction

Currently, robots and Embodied Conversational Agents (ECA) are starting to have robust techniques to detect emotion content via speech or image recognition. However, to complete the Human Computer Interaction cycle these systems still need to have ways of mapping these detected emotion states and magnitudes to system outputs. Many studies such as Liu et al. (2003), Luengo et al. (2010), Moilanen and Pulman (2007), and Calix et al. (2010) have been conducted on how to detect emotion classes or magnitudes from multimedia information such as text, audio, and images. However, the methods that can use predicted emotion classes and magnitudes to render emotion expressions in Embodied Conversation Agents (ECA) are still unclear. This paper proposes a computer graphics methodology that uses predicted values to render facial expressions using mesh morphing techniques.

Specifically, this paper discusses the use of regression approaches to estimate emotion magnitudes from sentence level speech features and then use these predicted magnitudes as weights for mesh morphing-based facial expression rendering.

The speech extension of the Affect Corpus 2.0 (Calix and Knapp 2011a) is used to train and test the emotion magnitude prediction model. A total of 33 speech features extracted at the sentence level are used. Blender's Mancandy 3-D puppet is used as the humanoid mesh to perform emotion expression rendering. This paper's main focus is on the system response using 3-D virtual agents.

Literature Review

Machine Learning

To address the issue of magnitude prediction, regression approaches can be implemented. Common regression approaches include linear and non-linear regression models such Linear Regression, Artificial Neural Networks (ANNs), and Support Vector Regression (SVR).

In linear regression, a set of features is used to find a function of a line that best fits the data. Concretely, we are looking for a hypothesis $h_\theta(\vec{x}) = \theta^T \vec{x}$ that can linearly fit the data. Where θ is the weight vector and $\vec{x}$ is the feature vector. The analysis is performed by minimizing the sum of the square errors of the predicted values versus the real values of the training samples. The cost function to be minimized is expressed as follows:

$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(\vec{x}_i) - y_i)^2 \quad (1)$$

This optimization is solved using the gradient descent technique or more directly using the normal equation approach. In the normal equation approach, the weight parameters are solved analytically with the following equation:

$$\theta = (X^T X)^{-1} X^T y \quad (2)$$

where θ is the weight vector, X is a matrix of all feature samples, T is the transpose symbol for the transpose of the matrix, and y is the vector of regression values.

Artificial neural networks are an alternative to linear regression when the data that is being modeled is not linear. A multilayer perceptron can learn new features from the initial data that can follow a non-linear shape. When using an artificial neural network, one of the first steps is to select the network architecture. At least one hidden layer is required with a given number of neurons in the layer. A reasonable approach, when using more than one hidden layer, is for each layer to have the same number of neurons and for the number of neurons per layer to be between the number of input features and 3 times that value. Usually, more neurons are better than fewer but a large number of neurons make the training of the algorithm more computationally expensive.

Artificial Neural Networks and SVR are important methods to model regression equations. ANNs and SVRs have the advantage that they can be used to map non-linear data to higher dimensional spaces and do not suffer from parametric assumptions and the requirements of linear regression such as normality. ANNs are common in the literature but can suffer problems related to data over fitting.

According to Smola and Scholkopf (2004), Support Vector Regression modeling is a regression approach based on the key ideas of Support Vector Machines (SVM). SVRs can perform better than linear regression and ANNs in some cases. The SVR technique is better because it tries to minimize not just empirical risk (least square error) but also structural risk. The objective in SVR is to fit a linear regression model to a data set in a higher dimensional feature space after mapping the data set from a non-linear input space. Similarly to regular least squares regression, a line (Equation 3) is fitted to the dataset in feature space by minimizing the sum of errors. Equation 3 denotes the dot product in the input space, x is the input vector, w is the weight vector, and b is the bias. Additionally, SVR allows

$$f(x) = \langle w, x \rangle + b \tag{3}$$

for an Epsilon (E) error margin. Therefore, only errors outside this margin are considered. This error margin helps to improve generalization by excluding samples inside the error margin or tolerance level. These errors or deviations above the accepted Epsilon (E) error margin are referred to as slack variables and are formally described as an E-intensive loss function (Smola and Scholkopf 2004). The weight vector and the bias are obtained by minimizing the objective function (Equation 4) subject to the constraints specified in Equation 5.

Embodied Conversational Agents

Responses that an automated system can have to human emotions are an important aspect in affective HCI. To complete the interaction cycle, an automated system needs to provide a response based on emotion inputs. The system response refers to what the machine will do in response to the detection of a particular emotion class or magnitude. Responses can be in tone of speech, facial expression, type of language, physical proximity in the case of robots, and environment background. A facial response system, for instance, can be implemented in hardware using a robot or in software using animation of computer graphics.

Robotic approaches have advantages with relation to presence because they are material tangible entities which a human can relate to and physically perceive. “Embodied Conversational Agents” (ECA) are another approach in human computer interaction. They are important in virtual worlds because they provide a visual simulation of human faces and expressions (Pelachaud 2009). These interfaces can be used as dialogue systems which the system uses to respond to the user. Important studies in this area include Massaro et al. (2001), Gratch (2002), and Cassell (2001).

Facial Expression Responses

One of the most influential studies on how facial expression of emotions is conceptualized was done by Paul Ekman (Ekman 1998, 1978). In this study, the authors showed that certain emotions are consistent across cultures and can be expressed by a set of face muscle positions.

Implementation of a facial response by rendering a 3-D animation of a human face is usually referred to as facial expression in virtual agents. Substantial research has gone into this area and there are many successful techniques that can be used to accurately render facial expressions using computer graphics (CG). In Noh and Neumann (1998), the authors provide a good survey of the main facial modeling and animation techniques.

They divide the field of facial modeling/animation into two basic groups which are: geometry manipulations and image manipulations. Geometry manipulations are further divided into interpolation and parameterization. Since this research is focused on automatic rendering, only interpolation methods are discussed. For the more detailed description of the field as a whole, the reader may refer to Noh and Neumann (1998).

Interpolation through mesh morphing between targets is an important technique used for facial expression rendering. Mesh morphing (Akenine-Moller et al. 2008) is the process of seamlessly generating a mesh by combining the vertices from a neutral mesh with the equivalent vertices from one or more pose meshes. This process is achieved using GLSL (OpenGL Shading Language) shader programming. The objective of mesh morphing is to perform an interpolation

on a per vertex basis that allows different points to be combined. Therefore, a difference vector is calculated between the neutral mesh point and the target mesh point. These difference vectors are added to the neutral vector and can be adjusted by weights. Using weights to adjust the deformation allows for facial expressions, talking sequences, and gestures to be modified by simply changing the weights of the per vertex difference between a neutral pose and a target. This technique has advantages with regards to speed, efficiency and quality of rendering because it performs all processing in the GPU using shader programming techniques. Wang et al. (2007) used this technique to render emotional expressions. Rendering can be done on any type of 3-D animation software such as OpenGL or Blender. For this paper, examples of these techniques are provided using GLSL shaders, and C\C++ with OpenGL.

Methodology

This paper is only concerned with the methodology necessary to produce a facial expression response without worrying about the actual medium. In theory, the methodology could work with any 3D (computer graphics) model or physical (mechanized) model such as a robot. Therefore, only the techniques to define a mapping model and the actual implementation medium are discussed. In this case the implementation medium used is a 3-D mesh of a humanoid man. Blender's "Mancandy" 3D puppet is used as the 3-D model although any 3D model could be used.

Machine Learning

The machine learning technique used in this work is Support Vector Regression. According to Smola and Scholkopf (2004), the SVR methodology tries to obtain a regression function that is as flat as possible (by minimizing structural risk or the norm of the weight vector) and with low prediction error (empirical error from the data). The difference between the predicted values and the actual values from the training set (predicted error) should be no more than a certain specified value (E). To deal with non-linear data, a kernel trick is used to map non-linear data to higher dimensional linear space. Common kernels include linear, radial basis function (RBF), polynomial kernels, and sigmoidal. Additional kernels can also be designed which are more specific to the data set being analyzed. Formally, the soft margin optimization problem can be formulated as follows:

Minimize:

$$\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}(\xi_i + \xi_i^*) \tag{4}$$

Subject to:

$$Y_i - w\cdot\phi(X_i) - b \le E + \xi_i \tag{5}$$

$$w\cdot\phi(X_i) + b - Y_i \le E + \xi_i^*$$

$$\xi_i, \xi_i^* \ge 0$$

where "i" is each sample, the variable C (cost) represents the tradeoff between prediction error and w, w is the weight vector, $\phi(.)$ is a function for the high dimensional feature space, ε_i and ε_i^* are the slack error variables, with E as the acceptable error level. The objective function (Equation 4) subject to constraints (Equation 5) is optimized to find the weight vector (w) and the bias (b) using Lagrange multipliers. A single minimum solution using quadratic programming optimization can be obtained which is one of the advantages of using this method. LibSVM (Chang and Lin 2001) in conjunction with WEKA (Witten and Frank 2005) were used to train and test the Support Vector Regression model.

The optimization model formulated with equations (4) and (5) can be represented as follows:

$$h(x)_i = \sum_{i=1}^{vs}(\lambda_i - \lambda_i^*)K(x_i, x) + b \tag{6}$$

where $K(x_i, x) = \phi^T(x_i)\cdot\phi(x)$ is the kernel function that maps the input vectors to higher dimensional. Here, x are the input vectors, and λ are the LaGrange multipliers.
One regression equation is estimated for every emotion class. In this case, five emotion regression equations were estimated for 5 emotion classes. The emotion classes are: happy, sad, angry, afraid, and surprised. Only five emotion classes are used to reduce the number of prediction equations and since GLSL techniques have a limit on the number of facial expression models that can be used. Of course, other approaches using other emotion classes or emotion primitives such as valence, expectation, power, and activation can be used in the future to provide more flexibility and a higher number of renderings.

Features

A regression approach to predict emotion magnitudes requires combining features in a linear or non-linear model. Emotion magnitudes are real scaled numbers

between 0 and 100 which indicate the intensity of a given emotion. The following features are used to train and test the model: A set of 33 sentence level speech features which include max and average F0 (pitch), max and average intensity, formants F1, F2, F3, F4, F5, and the standard deviation and mean for 12 Mel Frequency Cepstral Coefficients (MFCCs). The speech features are extracted from the Affect Corpus 2.0 using Praat scripts. A description of these features can be obtained in (Jurafsky and Martin 2008). Only speech signal features were used for this work.

System Response

In computer graphics, emotion magnitudes can be used as weights to adjust the emotion expression of 3-D character renderings. This approach can be implemented with the morph targets technique (Akenine-Moller et al. 2008) in which vertex level weighted emotion meshes are added to a neutral mesh. In this case, the weight, which determines the level of interpolation between the target and neutral mesh, can be determined by the emotion magnitude predicted by the model (Equation 6).
These weights are calculated based on the previously described regression models. Formally, the equation is as follows:

$$M_m = M_n + \sum_{q=1}^{t} w_q (M_n - M_t) \quad (7)$$

where M_m is the morphed mesh, M_n is the neutral mesh, t is the number of target meshes, M_t is the mesh for each target t, and w_q is the weight from 0 to 1 assigned to the morphing. This weight is the magnitude that is predicted by the regression model (Equation 6) for each emotion class.

Analysis and Results

Automatic Learning Results

The methodology is trained to predict emotion magnitudes from sentence level speech features. Support Vector Regression techniques were used to find the optimal model that can fit the data.

Table 1 - Regression Modeling Results

Analysis	RMSE	Correlation Coefficient
Speech features (prosody and spectral)	0.16 – 0.20	0.66 – 0.78

The results of the analysis are presented in Table 1. This analysis was performed using the SVR model with an RBF kernel.
Overall, the model was able to learn and achieved good prediction results. The SVR correlation coefficients for the 5 regression equations using an RBF kernel fell between a range of 0.66 and 0.78. The root mean square error (RMSE) results fell in the range of 16.00 to 20.00.
Finally, once all relevant processing has been done and emotion state magnitudes are estimated, the system can proceed to represent the emotion state of each sample. An XML tag can be added for each sample vector. For example, the following tag can be added for a given sample:

<enamex tag="sample"><pos tag="position"><emotion sent=3 happy=0.6 surprised=0.2/> <word f="Actor"/> </pos> </enamex>

Calix and Knapp (2011b) presents a more in-depth analysis of regression models that use text and speech features as well as feature recurrence.

Mesh Morphing

Mesh morphing requires the use of different instances of the same mesh in different poses. These meshes can be created in any animation software such as Blender or Maya. Once the meshes are created, they can be read into the system as "obj" files. The important aspect is how to store the data. One approach is to create a data structure to hold the vertex and normal data for the neutral pose plus the data from the target meshes. This new data structure is used instead of the traditional object vertex data structure used for the mesh object. The difference vectors for the vertices and normals can be calculated using vertex shaders. Since the change is performed at the vertex level, it is important to pass each neutral vertex as well as the corresponding pose vertices and normals to the GPU at the same time. To achieve this, the program needs to link each pose's information to the shader via the "attribute" parameters defined in GLSL. This way, the GPU will have access to the vertex information for the neutral mesh and all target meshes. The degree of interpolation can be controlled by passing weights via "uniform" parameters defined in GLSL.

```
if (morphWeight4 < 0)
      send_value = 1;
if (morphWeight4 > 1)
      send_value = -1;
morphWeight4 = morphWeight4 + 0.05*send_value;
```

Figure 1 – A simple talking sequence

An expressive talking sequence can also be implemented by incrementing a mouth-open weight using a step size. The weight grows until a threshold is reached. Once the maximum value is reached, the process is reversed and the step size decreases the weight (Figure 1). In Figure 1, morphWeight4 is the weight that adjusts the mouth interpolation, send_value is the direction in which the counter is adjusted (incrementing or decrementing), and 0.05 is the step size. The speed in which the mouth moves can be controlled by the step size. A larger step size can cause the mouth to open and close more quickly. This approach can also be used for expressive gestures.

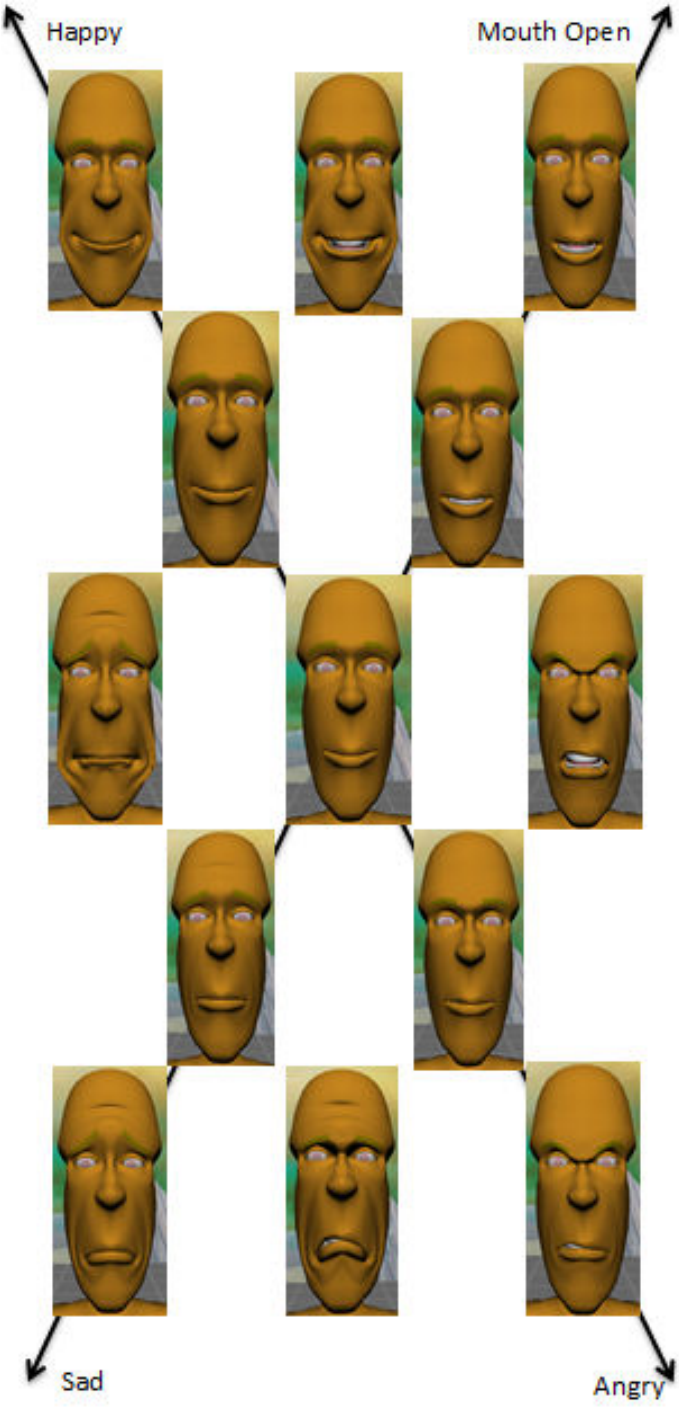

Figure 2: Emotion magnitude renderings

The advantage of real valued renderings is that the amount of facial expressions that can be generated is un-limited. Additionally, several emotion expressions can be combined to create new ones. An example of this approach can be visualized in Figure 2. Figure 2 represents a four dimensional space with axes for happy, sad, angry, and open mouth. Many emotion renderings can be generated by combining the different dimensions in this space. The transformation of the mesh from a neutral expression to other emotions is achieved efficiently, aesthetically, and quickly (Figure 2). In Figure 2, the neutral expression can be seen in the center of the figure. The lower left hand corner represents the maximum sad expression, the lower right hand corner represents the maximum angry expression, the upper left hand corner represents the maximum happy expression, and the upper right hand corner represents the maximum open mouth expression. Facial expressions in between these points represent combinations of these basic axes.

Conclusions

The prediction model combined with the morph targets technique can provide a powerful tool to automatically render emotional facial expressions from emotion detection in speech. It is important to note that this technique could be easily extended to gestures and that the application is not limited to emotion expressions alone.

Future work will include more text based features, and higher semantic approaches such as named entity recognition, nominal entity recognition, and anaphora resolution to better disambiguate the actors in each sentence. Additionally, perceptual usability studies will be performed to measure the effectiveness of the rendered facial expressions.

References

Akenine-Moller, T., Haines, E., Hoffman, N. 2008. Real-Time Rendering. A K Peters, Wellesley, Massachusetts.

Calix, R., Mallepudi, S., Chen, B., Knapp, G. 2010. Emotion Recognition in Text for 3-D Facial Expression Rendering. IEEE Transactions on Multimedia, Special Issue on Multimodal Affective Interaction, Volume 12, Issue 6, pp. 544-551.

Calix, R., Knapp, G.M. 2011a. Affect Corpus 2.0: An Extension of a Corpus for Actor Level Emotion Magnitude Detection. In Proceedings of the 2nd ACM Multimedia Systems (MMSys) conference, San Jose, California, U.S.A, pp. 129-132.

Calix, R. A.; Knapp, G. M. 2011b. Actor Level Emotion Magnitude Prediction in Text and Speech, Springer Multimedia Tools and Applications, http://dx.doi.org/10.1007/s11042-011-0909-8

Cassell, J. 2001. Embodied Conversational Agents: Representation and Intelligence in User Interface. AI Magazine, 22(3), pp. 67-83.

Chang, C.-C., Lin, C. 2001. LIBSVM: a library for support vector machines. Retrieved from http://www.csie ntu.edu.tw/~cjlin/libsvm.

Ekman, P., Friesen, W. 1978. The Facial Action Coding System: A technique for the measurement of facial movement. San Francisco: Consulting Psychologists Press.

Ekman, P., Friesen, W. 1998. Constants across culture in the face and emotion. In: Jenkins, Oatley, & Stein (eds.), Human Emotions: A Reader. Malden, MA: Blackwell, chapter 7.

Gratch, J., Rickel, J., Andre, E., Badler, N., Cassell, J., Petajan, E. 2002. Creating Interactive Virtual Humans: Some Assembly Required. IEEE Intelligent Systems, 17(4): 54-6.

Jurafsky, D., Martin, J. 2008. Speech and Language Processing, 2nd ed., New Jersey: Prentice Hall.

Liu, H., Lieberman, H., Selker, T. 2003. A Model of Textual Affect Sensing using Real-World Knowledge. In Proceedings of the 2003 International Conference on Intelligent User Interfaces, IUI 2003, January 12-15, 2003, Miami, FL, USA, ACM 2003, ISBN 1-58113-586-6, pp. 125-132.

Luengo, I., Navas, E., Hernaez, I. 2010. Feature analysis and evaluation for automatic emotion identification in speech. IEEE Transactions on Multimedia, Vol. 12, No. 6, pp. 490-501.

Massaro, D., Cohen, M., Beskow, J., Cole, R. 2001. Developing and evaluating conversational agents. Embodied conversational agents, MIT Press, Cambridge, MA.

Moilanen, K., Pulman, S. 2007. Sentiment Composition. In Proceedings of Recent Advances in Natural Language Processing (RANLP 2007), 27-29, Borovets, Bulgaria, pp. 378-382.

Noh, J., Neumann, U. 1998. A survey of facial modeling and animation techniques. Technical Report, USC, pp. 99-705.

Pelachaud, C. 2009. Modeling multimodal expression of emotion in a virtual agent. Philosoph. Trans. Roy. Soc. B Biol. Sci., B, vol. 364, pp.3539-3548.

Smola, A., Scholkopf, B. 2004. A tutorial on Support Vector Regression. Statistics and Computing, Vol. 14, pp. 199-222.

Wang, A., Emmi, M., Faloutsos, P. 2007. Assembling an Expressive Facial Animation System. In Proceedings of the 2007 ACM SIGGRAPH symposium on video games, New York, New York.

Witten, I., Frank, E. 2005. Data Mining: Practical Machine Learning Tools and Techniques. 2d. edition, Morgan Kaufmann Publishers Inc., San Francisco.