Journal of Convergence Information Technology
Volume 3, Number 4, December 2008

Affective Space Calibration in Action-rich Media Affective Systems

Anestis A. Toptsis*Corresponding author, Alexander Dubitski
Dept. of Computer Science and Engineering,
York University, Toronto, Ontario, M3J 1P3, Canada,
anestis@yorku.ca, dubitski@yorku.ca

Abstract

Affective computing systems are software systems that take in account the emotional state of the user during their operation. Recently it is increasingly accepted that combining emotional with intellectual processing makes a system both more realistic and more intelligent. As such, affective computing systems are considered by many to represent the highest form of artificial intelligence and the state of the art in computer human interaction. A major difficulty in building an affective computing system is to identify what emotion attributes are most applicable for the development of the particular system that is under development. If such information is available then a system's developer can build the sought system by using only the emotion attributes that are most influential for that type of system. In this work we present a method that addresses this problem. We describe our method and present an evaluation from its implementation and experiments with two different media collections. Our findings indicate that the method can successfully pinpoint to the most influential emotion attributes and thus allow the prospective developer to select the most applicable attributes for building a production-grade affective computing system.

Keywords

Artificial Intelligence, Affective Computing, Human-Computer Interaction.

1. Introduction

Similar to conventional software systems, the lifecycle of affective computing systems is divided into two major phases: the system's development phase and the system's usage phase. What distinguishes conventional software systems form affective computing systems is that during the development phase of an affective computing system the user's affective state is also taken in account and incorporated into the workings of the systems. Also, during the usage phase, an affective system behaves according to the user's current affective state. The main goal of Affective Computing [1], [2] is to incorporate emotion-related information into computing systems. In recent years it is commonly agreed that emotion-involving affective processes have significant impact on the overall intelligence of any system, including humans [3]. In fact, as it is argued by one of the foremost figures in Artificial Intelligence, the emotional state of a person is simply yet another way of thinking [4]. There are several encouraging findings that demonstrate that incorporation of emotions in intelligent systems increases both user-friendliness and system performance. One common characteristic in projects like the ones described in [5] and [6] is that they rely on a predetermined set of emotions (such as the set of 6 basic emotions developed by Ekman [10]) and then based on the chosen set they develop a system that uses those emotions to improve the system's performance. Other projects, such as the ones in [8] and [9], prefer to use physiological measurements such as heart rate, blood pressure, skin resonance, which are allegedly affected by certain emotional states of a person (or, visa versa, the emotional state affects the physiological measurements). All above approaches are deemed superior to any approach which does not take in consideration the affective state of a person. However, all approaches also have a common weakness: *which emotions*, among a set of potentially tens (or hundreds) of emotions are the most important for the problem at hand? The complexity of this question is multifaceted. First, there is a plethora of classifications of emotions [11], i.e., there is significant controversy among emotion theory experts (psychologists and cognitive scientists) regarding what the best way to categorize emotions is, or even which emotions are most important. In general, when developing an affective computing system, we do not know which emotion classification is most appropriate for that system. Second, once we select a classification, we do not know which emotions within that classification are more important for the system under development. We argue that more often than not, not all emotions within a classification are equally important for any system. Note, ignoring the importance of different emotion attributes within a system may render the developed system either hard to use (if we choose to use too many emotions, for which the system would have to acquire input from the user), or diminish the accuracy of the system (if potentially *non-applicable* emotion attributes are used by the system's organization and retrieval mechanisms).

In this work we propose a methodology that can be used to make an educated guess of *which emotion attributes may be most applicable for the development of a particular affective computing system*. Specifically, we assume that the system under discussion is to organize a media collection in such a way that facilitates browsing that conforms to a user's emotional states. That is, if the user wants to experience media content that conforms to a certain "mood" of the user, then the system is able to retrieve such content. The proposed method assumes that an algorithm for organizing and mining a media collection is available. The purpose of the work presented in this paper is not to propose an algorithm for organizing the collection, but to determine which of the emotion attributes that the system intends to use are more influential than other emotion attributes for the purpose of building the desired system. Once such a determination is made, then an algorithm such as the one of [6], or [7], or some other method can be used to organize the collection *using only the emotion attributes that are found to be most influential* for the particular project. The benefits from the proposed method are: (1) after we know which emotion attributes are most influential for the type of system that we want to develop, we can use only those few emotion attributes for the development of the production-grade system. This will make the development of the system faster and simpler, and (2) the usability of the system will be increased since only few emotion attributes will have to be acquired from the user when she is using the system. The rest of the paper is organized as follows.

Section 2 presents definitions that are used throughout the rest of the paper. Section 3 provides related technical background and presents our method. In Section 4 we present an evaluation of our method. Section 5 summarizes our findings and discusses future research ideas.

2. Definitions

1) We define ASp, the *Affective Space*, as a set

$$ASp = \{e_1, ..., e_M\},$$

where e_i is an *emotion attribute*. An emotion attribute represents a particular emotion, such as "fear", "sadness", etc. As discussed in Section 1, there are hundreds of emotion attributes. A casual search of the term "emotion" on Google (October 19, 2008) produces about 54 million entries.

2) We define SM, the *State of Mind*, as a tuple of N values,

$$SM = \langle v_1, ..., v_N \rangle,$$

where v_j, j=1, …, N, is a number that represents the intensity of emotion attribute e_j, with $e_j \in ASp$. In general, for the development of a production-grade version of an affective computing system it is $N < M$, unless *all* emotion attributes of ASp are used.

3) We define an *Action* to be an activity performed during using the system and such that this activity is influenced by, and/or influences the user's SM. An example of an Action is when a user presses the mouse button too hard during working with a user un-friendly GUI (the amount of frustration in the user's SM is proportional to the pressure that the user exercises on the mouse button). Another example of Action is when a user selects a particular song for listening from a song collection (if the intensity of the "sadness" emotion attribute in the user's SM is high, then it is more likely that the user selects a "sad song" for listening, and visa versa; or if the user selects a "happy song" then the intensity of the "happiness" emotion attribute in the user's SM may increase).

4) We define *Affect* as the *amount of change* of SM due to an action. Note, the Affect value is high for actions that produce drastic fluctuations of the value of SM, whereas the affect value is low for actions that produce little fluctuation of the SM value. Throughout this paper we use the symbol ψ to denote the affect value.

5) We define *Suitability* as the *degree of appropriateness* of an emotion attribute for an affective system. Throughout this paper we use the symbol σ to denote the suitability value. Note, the ψ-value (i.e., the affect) and the σ-value (i.e., the suitability) are not necessarily the same for a given affective system. An affective computing system may have high ψ-value for a particular emotion attribute (e.g. fear) but a low σ-value for the same emotion attribute. The reason is that the particular emotion attribute may exhibit drastic fluctuations in its intensity in that system, but at the same time the same emotion attribute may not be one of the pivotal elements that determine the quality of that system. An example of such a system could be a system that guides a user for making extremely short-term and real-time stock market

investment decisions. During the operation of such a system, the emotion attributes “anger” and “excitement” are most likely to exhibit very high Affect values due to human nature. However, the emotion attributes “calmness” and “clarity of mind” are most likely to be more applicable for being taken in account for the development of such a system.

6) A *K-Line* is an artificial representation of memory in the human brain. K-lines is a structure introduced in [12] and further discussed in [13] and [4], and it is useful for capturing memory experiences. In our work, each K-Line resembles a linked list consisting of *K-Nodes*.

7) A *K-Node* is a node similar to a node in a linked list. It contains a SM and an Action, as defined above. The SM is acquired from the user before or during the Action. The idea is that the SM of each K-node corresponds to the emotional state of the user, associated with the Action that is performed by that user.

3. Our Method

We assume that we face the task of creating a system that facilitates retrieval of affectively annotated content that conforms to a user’s emotional state. In creating such a system, we typically take in account the user’s emotional states during the organization of a collection and also during the process of mining appropriate content. A typical characteristic of methods for developing such systems is to model the emotional state of the user using a set of emotion attributes (such as “sadness”, fear”, etc) which are typically borrowed from available literature in Psychology and/or Cognitive Science. The problem with such an approach is that not all selected emotion attributes may be relevant or equally significant for the particular project at hand. In this section we propose a methodology that can be used to make an educated guess of which emotion attributes may be most influential for the development of such a system. Once such knowledge is acquired, then the system’s developer(s) can use *only the most influential emotion attributes* to develop a final production-grade version of the desired system.

The proposed method is outlined in Algorithm A.

Algorithm A

(A.1) choose an affective space

$$ASp = \{e_1, ..., e_M\}$$

and a state of mind,

$$SM_{all} = \langle v_1, ..., v_M \rangle ,$$

where v_j is a value that represents the intensity of emotion attribute e_j of the ASp.

(A.2) create a mini system, S, using SM_{all}.

(A.3) For (i=1, …, M) calculate ψ_i, the Affect value of e_i in S. In this step, the $\psi - values$ are also linearly normalized into a scale [0..100] and sorted in ascending order.

(A.4) Calculate the suitability values σ_a, σ_b, σ_c, for the emotion attributes e_a, e_b, and e_c, where e_a is the emotion attribute with affect value $\psi_a = \min_i \{\psi_i\}$, e_c is the emotion attribute with affect value $\psi_c = \max_i \{\psi_i\}$, and e_b is the emotion attribute with affect value ψ_{Mid}, ψ_{Mid} being the ψ_i which is *closest* to the midpoint $\frac{\psi_a + \psi_c}{2}$. Consider the parabola P determined by the points $P_a = (\psi_a, \sigma_a)$, $P_b = (\psi_b, \sigma_b)$, and $P_c = (\psi_c, \sigma_c)$.

(A.5) Reduce the size of ASp by leaving in it only those emotion attributes which have suitability values above a desired threshold level, based on the parabola P from step (A.4)

(A.6) Build the production-grade system using as ASp the ASp resulting from (A.5);

Step (A.1) of Algorithm A is performed by the prospective developer of the final system. In typical existing systems (such as the ones of [5] and [6]) the ASp is usually small (six emotion attributes) and follows Ekman’s original classification, one of the many available emotion classifications. For the purposes of Algorithm A, the chosen ASp does not need to follow any classification (although it can) and it can be as large as desired.

Step (A.2) of Algorithm A is performed by using Algorithm B, shown below.

Algorithm B

```
(B.1) Create K-lines using past
      user's experiences.
(B.2) Create K-line mesh using the
      K-lines formed in previous step.
```

Step (B.1) of algorithm B is performed according to Algorithm C, shown next. Algorithm C is an adaptation of an algorithm presented in [6] for the purpose of organizing media repositories.

Algorithm C

```
1. Periodicity = P;
2. s_ID = 1;
3. repeat {
       a. PerformSession( S_id)
       b. S_id++;
4. }until (system or user quits);
```

(method)

```
PerformSession ( k ) {
1.  count = 0;
2.  repeat {
3.  user selects Action A;
4.  if (isKlineIntersection(k, S)){
          a. user enters SM_all ;
          b. count = 0;
5.  }
6.  elseif ( count % P == 0 ) {
          a. user enters SM_all ;
          b. count = 0;
7.  }
8.  else { SM_all = null; }
9.  form_Knode( SM_all , null, A, null );
10. count++;
11. } until (system or user quits) ;
}     // end PerformSession ( k )
```

(method)

`Form_Knode(` SM_{all} , Action**`)`**

- **Input:** the data fields of a K-node per the description of Section 2.
- **Action:** forms a K-node and initializes its fields using the input. Appends the K-node to the K-line that is in progress.
- **Intelligence:** all songs selected during the formation of the same K-line are different.

(method)

`isKlineIntersection ( k, A ))`

- **Input:** a K-line ID k and an Action A.
- **Returns**: *True*, if Action A which is about to be part of a K-node of K-line k was selected earlier during the formation of another K-line. *False*, otherwise.

Algorithm C performs several sessions. During each session the user selects Actions to be executed. No two same Actions can be selected within the same K-line. According to the conditions specified in lines 4 and 6 of Algorithm C, the user is prompted for her SM during the selection of certain Actions.

Step (B.2) of algorithm B is performed using an adaptation of an algorithm for organizing media collections, presented in [7]. This algorithm organizes an affectively annotated collection that facilitates retrieval of affectively annotated content that conforms to the current mood of the user. For ease of reference, we refer to this method as method **KLMF** throughout this paper. In essence, method KLMF creates a network of nodes as shown in Figure 1.

Each node in Figure 1 is a *K-node*. A K-node contains information about the user's state of mind expressed as a tuple of values that represent intensities of the emotion attributes of the used affective space. Each K-node also holds information about the particular affectively annotated piece of data Acted upon by the user during the creation of that K-node. Each line in Figure 1 is a *K-line*. A K-line represents a sequence of memory experiences of the user as formed when the user selected specific affectively annotated content during the formation of the K-line. Figure 1 shows a K-line mesh consisting of 4 K-lines. (For readability purposes, each K-line is drawn with a different color and different thickness).

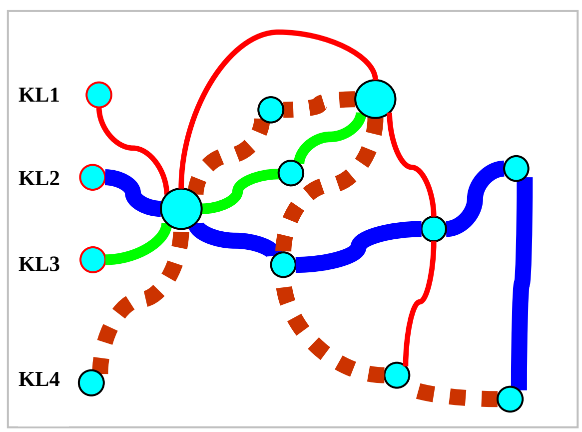

Figure 1. A K-line mesh

Step (A.3) of Algorithm A determines ψ_i (for i=1,…,M), the amount of change of each of the emotion attributes $e_1,\ldots,e_M$ from the K-line mesh formed in step (A.2). To do this, we use information that is accumulated during the execution of method KLMF that forms the K-line mesh. Specifically, during the execution of KLMF, a graph like the one shown in Figure 2 is created and associated with *each* of the emotion attributes of the ASp *and each* of the K-lines that comprise the K-line mesh. I.e., for an ASp of M emotion attributes and a K-line mesh of 4 K-lines, M*4 such graphs are created during step (A.2).

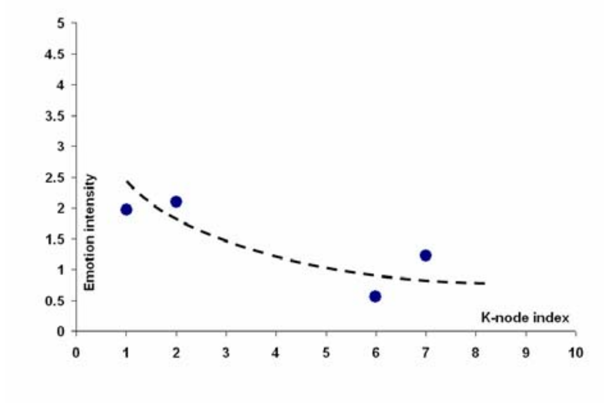

Figure 2. Average intensity of one emotion attribute in one K-line. Emotion attribute e_i; K-line *KL-j*

Figure 2 shows one of the M graphs for one of the K-lines of the K-line mesh. We use all such graphs for step (A.3) of our method presented in this work. The dashed line in Figure 2 shows the estimated intensity of an emotion attribute e_i for a K-line *KL-j*. The purpose of step (A.3) is to calculate the amount of fluctuation of emotion attribute e_i across the entire K-line mesh. Next, we describe how this calculation is performed.

Assume that the K-line mesh contains L K-lines and $C_1^{<i>}, C_2^{<i>}, \ldots, C_L^{<i>}$ are the curves of the i-th emotion attribute for the K-line mesh. Consider the set

$$EC^{<i>} = \left\{ C_1^{<i>}, \ldots, C_L^{<i>} \right\}.$$

Partition set EC into sets

$$EC_1^{<i>},\ EC_2^{<i>}, \ldots, EC_w^{<i>}$$

such that all curves in each set $EC_j^{<i>}$, j=1,…,w, are *equivalent*. That is,

$$EC^{<m>} = \bigcup_{i=1}^{w} EC_i^{<m>} \quad \text{and}$$

$$EC_i^{<m>} \cap EC_j^{<m>} = \varnothing \quad \text{for any i, j, and}$$

$$C_{ij}^{<m>} \approx C_{ik}^{<m>}, \text{ for any two curves } C_{ij}^{<m>}, C_{ik}^{<m>}$$

in $EC_i^{<m>}$. (the symbol $\approx$ is used to denote equivalency).

Definition:

Two curves C_a and C_b are *equivalent* (denoted by $C_a \approx C_b$), if and only if the difference of their covering areas is small enough within a system defined threshold value ε. The shaded (red) area in Figure 3 is the difference of the covering areas of the two curves shown in that Figure. (the covering area of a curve is the area defined by the curve itself, the straight line segments defined by the projections of the end points of the curve on the horizontal axis, and the segment of the horizontal axis between the two projections). The curves in Figure 3 are calculated by the KLMF process. The horizontal axis in Figure 3 represents K-node indices and the vertical axis represents intensity values of an emotion attribute.

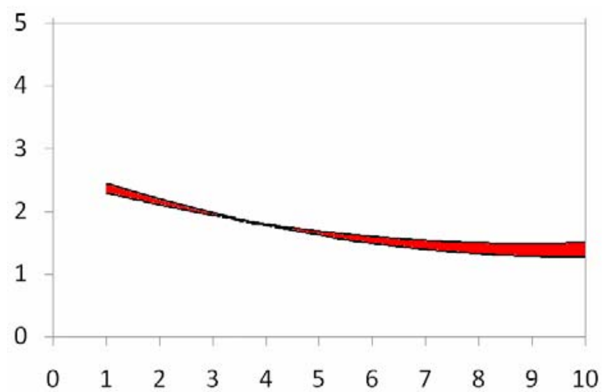

Figure 3. Two curves with small difference of covering areas.

After the partitioning operation, we have sets

$$EC_1^{<m>}, EC_2^{<m>}, \ldots, EC_w^{<m>}$$

such that all curves within a set $EC_j^{<m>}$ are equivalent. Figure 4 illustrates one such set of curves.

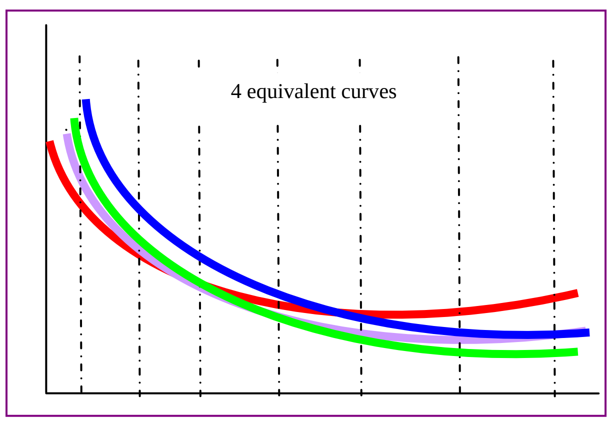

Figure 4. Four equivalent curves.

Next, we calculate the average curve,

$$average\,(EC_j^{<m>}) = \overline{EC_j^{<m>}}$$

for each set $EC_j^{<m>}$. Figure 5 shows the situation for 2 curves.

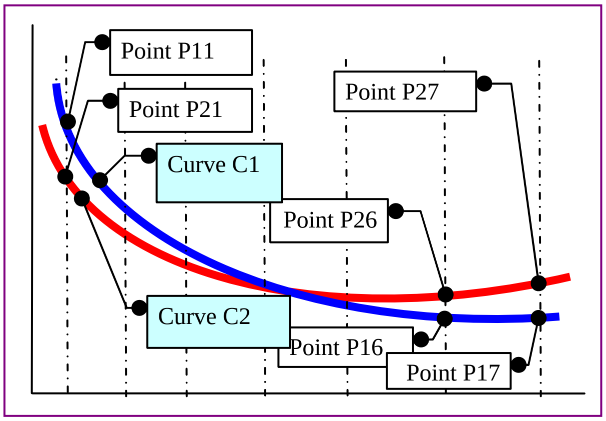

Figure 5. Calculating the average curve.

The average curve of

$$EC_j^{<m>} = \left\{ C_{j1}^{<m>}, C_{j2}^{<m>}, ..., C_{jN}^{<m>} \right\},$$

is the curve

$$\overline{EC_i^{<m>}} = \bigotimes_{i=1}^{N} \left[P_{xi}^{<m>} \right],$$

where:

- N is the size of the longest K-line among all K-lines that correspond to the curves of set $EC_j^{<m>}$.
- $P_{xi}^{<m>}$ is the point $\left(x, F_{ji}^{<m>}(x)\right)$, where x is the K-node index within the i-th K-line, and the i-th K-line corresponds to the curve that is determined by function $F_{ji}^{<m>}(x)$ calculated from the KLMF process.
- The operator $\bigotimes_{i=1}^{N}[\text{points}]$ denotes the interpolation process on N points using the least squares method (the least squares method is a method that allows a type of regression analysis and it was introduced by Gauss around 1794. Descriptions of the method can be found in various textbooks, including [18], [19], and [20]).

At the end of the above process, we have the average curves $\overline{EC_1^{<m>}}, \overline{EC_2^{<m>}}, ..., \overline{EC_w^{<m>}}$.

Next, we take the standard deviation ξ_i,

$$\xi_i = \xi\left(\overline{EC_i^{<m>}}\right) = \frac{1}{x_1 - x_2} \cdot \left(\left(\int_{x_1}^{x_2} \left(\overline{EC_i^{<m>}} \right) \right) - \mu_i^2 \right),$$

$$\mu_i^2 = \frac{1}{x_1 - x_2} \cdot \left(\int_{x_1}^{x_2} \left(\overline{EC_i^{<m>}} \right) \right)$$

where x_1, x_2 are the minimum and maximum x-values of the curve $\overline{EC_i^{<m>}}$, respectively. The resulting value of the standard deviation ξ_i, of each of these curves shows the degree of *change* of the corresponding curve. If

$$\xi\left(\overline{EC_{i_1}^{<m>}}\right) > \xi\left(\overline{EC_{i_2}^{<m>}}\right) > ... > \xi\left(\overline{EC_{i_w}^{<m>}}\right),$$

then the median of the sequence

$$s = \left[\xi\left(\overline{EC_{i_1}^{<m>}}\right), \xi\left(\overline{EC_{i_2}^{<m>}}\right), ..., \xi\left(\overline{EC_{i_w}^{<m>}}\right) \right]$$

is a number that corresponds to a curve $MC^{<m>}$. This is a curve of medium degree of change, compared to the degree of change of all the curves whose standard deviations are in sequence s. We choose this curve as the representative curve for the arousal fluctuation of the m-th emotion attribute.

The above process is repeated M times, one time for each of the M emotion attributes that comprise $SM_{all} = \{e_1, ..., e_M\}$. Upon completion of this process, we have a set of curves

$$MC_{all} = \left\{ MC^{<1>}, MC^{<2>}, ..., MC^{<M>} \right\}.$$

Curve $MC^{<i>}$ depicts the Affect value (amount of change) of the m-th emotion attribute across the entire K-line mesh. Note, this fluctuation is a result of taking actions of type Action. The standard deviation for curve $MC^{<i>}$ is a number ξ_i that represents the amount of change of the i-th emotion attribute, across the entire K-line mesh. By linearly normalizing ξ_i to a value ψ_i, such that $0 \le \psi_i \le 100$, ψ_i is the *relative* impact, in terms of change, of the i-th emotion attribute as compared to the impact of all other emotion attributes.

Step (A.4) of Algorithm A determines the suitability of *three* emotion attributes for our system.

0. Select three ψ_i's, $\psi_a = \min_i\{\psi_i\}$, $\psi_c = \max_i\{\psi_i\}$, and ψ_b = closest ψ_i value to the midpoint $\frac{\psi_a + \psi_c}{2}$. Assume that e_a, e_b, and e_c are the emotion attributes that correspond to the selected ψ_a, ψ_b, and ψ_c.

1. For each emotion attribute e_a, e_b, and e_c perform a K-line mesh creation similar to the one done in step (A.2), but *using an affective space consisting of only one emotion attribute*. Form one K-line mesh for each $SM_i = \{e_i\}$, i=a, b, c.
2. For each formed K-line mesh, record the user's satisfaction for the performance of that mesh. The user's satisfaction is recorded as a percentage. 0% is recorded if the K-line mesh performs in total disagreement of what the user expected; 100% is recorded if the K-line mesh performs in complete agreement to what the user expected. These percentages are calculated using one of our algorithms from [6] or [7]. Due to space considerations, we do not reproduce our algorithm here, but in summary, the algorithm is based on a relevance feedback mechanism in which the system retrieves perceivably appropriate K-nodes and the user reports her degree of satisfaction for each retrieved K-node.
3. The values recorded during step 2 above, are the three sought *suitability values* σ_a, σ_b, and σ_c.
4. Using the three suitability values σ_a, σ_b, and σ_c calculated in step 3 above, and their corresponding $\psi - values$ ψ_a, ψ_b, and ψ_c calculated during step (A.2) of Algorithm A, consider the points $P_a = (\psi_a, \sigma_a)$, $P_b = (\psi_b, \sigma_b)$, and $P_c = (\psi_c, \sigma_c)$, in a 2-dimensional space. Figure 6 illustrates three such points.

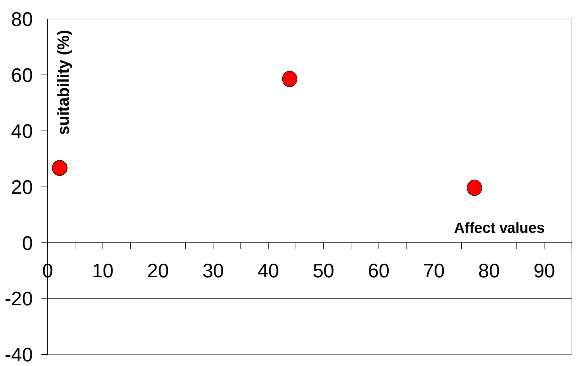

Figure 6. The three points $P_a = (\psi_a, \sigma_a)$, $P_b = (\psi_b, \sigma_b)$, and $P_c = (\psi_c, \sigma_c)$ plotted in a 2-dimensional plane.

The plotted points in Figure 6 represent the reported user suitability values for each of the emotion attributes associated with the three selected $\psi - values$.

The values on the horizontal axis are the affect $\psi - values$ calculated in step (A.3), sorted in ascending order.

Given the three 2-dimensional points $P_a = (\psi_a, \sigma_a)$, $P_b = (\psi_b, \sigma_b)$, and $P_c = (\psi_c, \sigma_c)$, consider the parabola

$$f(x) = Ax^2 + Bx + C \qquad \textbf{(1)}$$

where

$$A = \sigma_a - \Omega \cdot \psi_a{}^2 - \left(\frac{\sigma_b - \sigma_a}{\psi_b - \psi_a} - \Omega \cdot (\psi_b + \psi_a) \right) \cdot \psi_a ,$$

$$B = \frac{\sigma_b - \sigma_a}{\psi_b - \psi_a} - \Omega \cdot (\psi_b + \psi_a) ,$$

$$C = \Omega ,$$

and

$$\Omega = \left(\frac{\sigma_c - \sigma_a}{\psi_c - \psi_a} - \frac{\sigma_b - \sigma_a}{\psi_b - \psi_a} \right) \cdot (\psi_c - \psi_b)^{-1} .$$

The parabola specified by expression (1), has the characteristic that it passes through these three points. Figure 7 illustrates such a parabola.

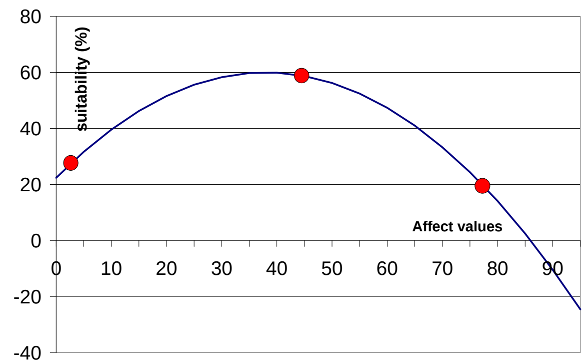

Figure 7. Parabola specified by expression (1), passing through the three points $P_a = (\psi_a, \sigma_a)$, $P_b = (\psi_b, \sigma_b)$, and $P_c = (\psi_c, \sigma_c)$.

Step (A.5) of Algorithm A is based on the findings of step (A.4). During this step, the system developer may consider reducing the size of the original affective space to only those emotion attributes that have suitability values for which the points of the calculated parabola from the previous step are located above a threshold line.

Using the suitability $\sigma - values$ from step (A.4) of algorithm A, and the affect $\psi - values$ from step (A.3) of algorithm A, we construct a graph like the one shown in Figure 8.

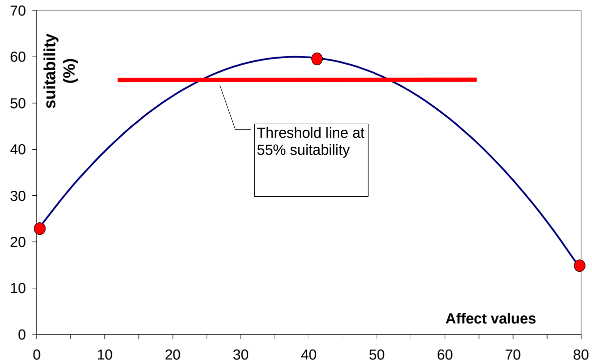

Figure 8. Decision making for suitability threshold of 55%.

Figure 8, illustrates a calculated parabola for a system of 80 emotion attributes (that is, $ASp = \{e_1,...,e_{80}\}$) with a suitability threshold line at 55%.. The plotted points in Figure 8 correspond to the reported $\sigma - values$, i.e. user suitability values for the three emotion attributes used in step (A.4). The values plotted on the horizontal axis are the $\psi - values$ calculated in step (A.3) of algorithm A, sorted in ascending order. A system defined threshold suitability value (55% in Figure 8) is used to determine which of the emotion attributes are most influential for our system. In Figure 8, the emotion attributes that correspond to $\psi - values$ ranging from 25 to 51 are highly influential in our system (i.e., very applicable to be used in building a production-grade version of this system), whereas all the other emotion attributes are not influential. Based on this finding, the

$$ASp = \{e_1,...,e_{80}\}$$

can be reduced to

$$ASp_{suitable} = \{e_{25},...,e_{51}\}.$$

That is, the size of the affective space is cut to about one third of its initial size. The significance of this is that once it is found that those emotion attributes are the ones that are most influential in the type of system that performs actions of type Action, then a production-grade system can be built based only on those most influential emotion attributes rather than all emotion attributes. The major advantage of such a system will be its ease of use, since it will have to deal with fewer emotion attributes than a system which is built without the knowledge of which emotion attributes are most influential for that system.

4. Evaluation

We evaluate the method described in section 3 for two different media collections. The first collection, SC, is a collection of 300 songs. The second collection, VCC, is a collection of 200 video clips. From the SC collection, 75 songs are used to build mini systems for the purpose of constructing K-line meshes that are used to calculate the suitability values as described in Section 2. From the VCC collection, 100 video clips are used to construct the corresponding mini systems, according to step (A.2) of Algorithm A. We use 10 emotion attributes, borrowed from the emotion attributes of the most recent Ekman's classification [14], i.e., our affective space is ASp = { *Amusement, Excitement, Sadness, Happiness, Anger, Relief, Embarrassment, Pleasure, Satisfaction, Pride* }. Figure 9 and Figure 10 show the calculated $\psi - values$ for ψ_i (i=1,…, 10), and the $\sigma - values$ percentages for each of the two collections.

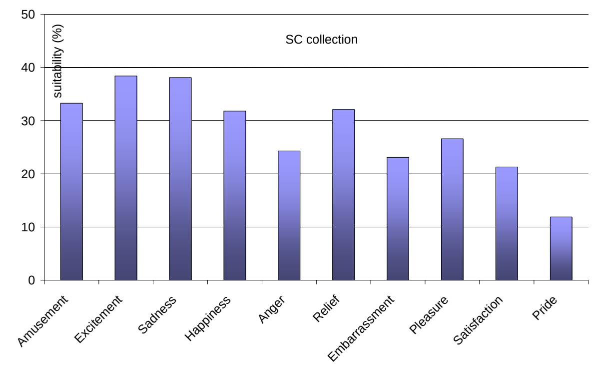

Figure 9. Suitability of emotion attributes for a Song collection.

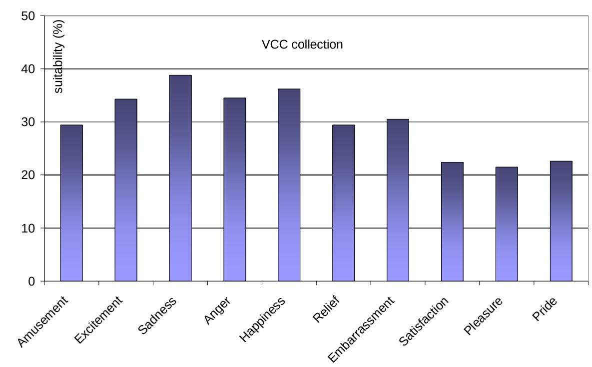

Figure 10. Suitability of emotion attributes for a Video Clip collection.

Table 1 and Table 2 show the $\psi - values$ for Figures 9 and 10. Note, the horizontal axes of Figures 9 and 10 list emotion attribute names, but these names are listed according to increasing order of the calculated ψ_i values for the corresponding emotion attribute.

Table 1. $\psi - values$ for Figure 9

	$\psi - value$
Amusement	6
Excitement	11
Sadness	67
Happiness	68
Anger	71
Relief	73
Embarrassment	74
Pleasure	76
Satisfaction	79
Pride	81

Table 2. $\psi - values$ for Figure 10

	$\psi - value$
Amusement	5
Excitement	7
Sadness	10
Anger	12
Happiness	61
Relief	64
Embarrassment	68
Satisfaction	69
Pleasure	70
Pride	72

Next, following step (A.4) of Algorithm A, we select the three $\psi - values$ and we form the quadratic curve defined by those values. Figures 11 and 12 show the three points corresponding to those $\psi - values$ and the curves for the SC and VCC collections, respectively.

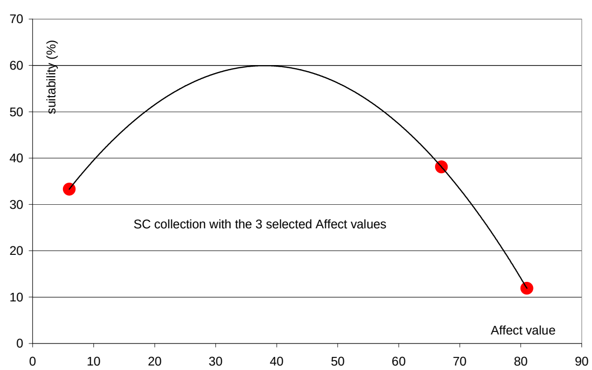

Figure 11. Three pivotal points in SC collection and the suitability determining curve.

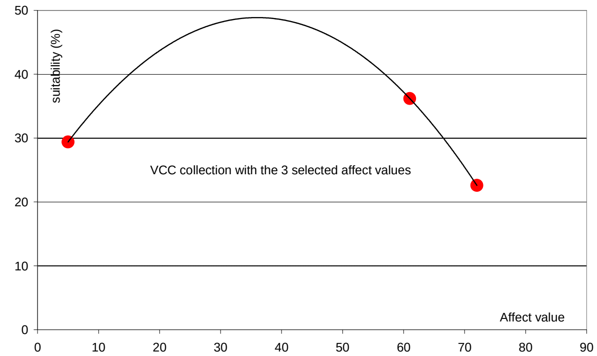

Figure 12. Three pivotal points in VCC collection and the suitability determining curve.

According to our algorithm presented in Section 3, the calculated curves of Figure 11 and Figure 12 represent an estimate of the $\sigma - values$ that were not calculated. That is, the $\sigma - values$ that were not calculated, are estimated to correspond to points of the calculated curve. Based on this estimate, the system's developer can then make an educated guess of what are the most appropriate emotion attributes to use for the development of a production-grade version of the affective system.

As part of our experiment, in order to confirm the working of our algorithm, we also calculate the actual $\sigma - values$ that are missing in Figure 11 and Figure 12. (Note, the system's developer will never have to calculate those values; actually, the main idea of our method is to allow the system's developer to avoid such calculations since they are a very time-consuming process). Figures 13 and 14 show the calculated $\sigma - values$ that are missing from Figures 11 and 12, along with the curves.

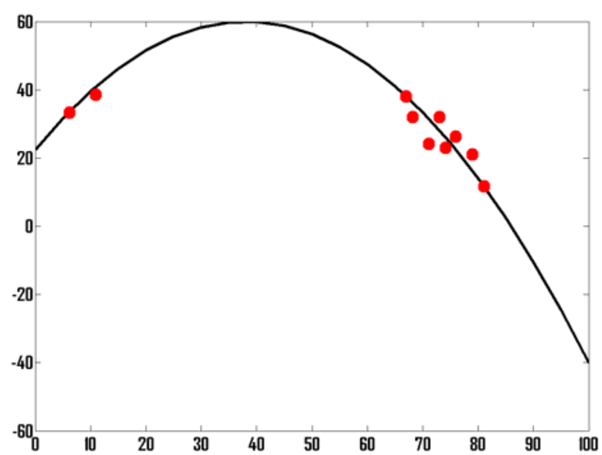

Figure 13. SC collection curve that approximates suitability; actual suitability values are also displayed.

As we see in Figure 13 the actual $\sigma - values$ for the SC collection are very close to the calculated parabola. This means that the parabola calculated by our algorithm using the three pivot points $P_a = (\psi_a, \sigma_a)$, $P_b = (\psi_b, \sigma_b)$, and $P_c = (\psi_c, \sigma_c)$, indeed provides a viable estimate of the missing $\sigma - values$. The actual $\sigma - values$ are shown in Table 3.

Table 3.

$\sigma - values$ **for SC collection**
33.3
38.4
38.1
31.8
24.3
32.1
23.1
26.6
21.3
11.9

The same applies for the VCC collection, as illustrated by Figure 14. The actual $\sigma - values$ for the VCC collection are shown in Table 4.

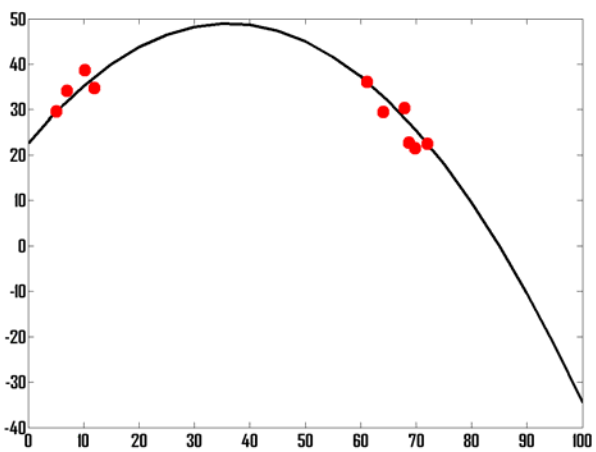

Figure 14. VCC collection curve that approximates suitability; actual suitability values are also displayed.

Table 4.

$\sigma - values$ for VCC collection
29.4
34.3
38.8
34.5
36.2
29.4
30.5
22.4
21.5
22.6

Comparison with previous work: To the best of our knowledge there is only one published work [15] related to the one appearing in this paper. This is a method described by the authors in which an affective system's most applicable emotion attributes are estimated by performing brute-force and *complete calculation* of *all* the $\sigma - values$ of *all the emotion attributes*. The work of [15] shows promising results, however its main drawback is that it is not practical for large affective spaces, i.e., when there are many initial emotion attributes under consideration. Contrary to the work in [15], the method presented in this paper does not depend on the initial size of the affective space ASp. This is because the method presented here uses only three pivotal $\psi - values$ and then, based on the calculated parabola (as shown, for example, in Figure 13) provides an estimate for *all* $\sigma - values$ without having to explicitly calculate those values. The savings realized from this approach are significant in terms of time required to calibrate the affective space prior to building a production-grade system. Also, as demonstrated by our experiments, the estimates provided by our method here are actually very close (within an insignificant margin as, for example, is shown in Table 3) to the actual $\sigma - values$ that would be calculated from the method in [15].

5. Conclusion

Our method determines the most suitable emotion attributes for building an affective computing system that organizes a media collection. A prospective system developer can apply our method and determine the most influential emotion attributes for the system sought to develop. Based on such findings, the developer can then calibrate the affective space to be comprised of the emotion attributes that are most applicable to that system, rather than using all the attributes that come with a typical off-the-self emotion classification. The benefit of this approach is that the final production-grade system will be developed faster, since the size of the affective space recommended by our method is most likely significantly smaller than the size of any affective space recommended by any off-the-self emotion classification. In addition to the above benefit, our method is efficient (i.e., it can be executed in a fairly short amount of time) since it requires *only three* emotion attributes to perform its estimates for the suitability of *all* emotion attributes of the initial affective space. As demonstrated by the experiments presented in Section 4, those carefully chosen three emotion attributes provide an estimate for the suitability of all other emotion attributes, with suitability values that are remarkably close to the actual suitability values should the latter were calculated explicitly.

In the future, we are interested to automate the acquisition of the intensities of emotion attributes for our system. Currently, in several stages during the execution of our method, our system acquires the intensity values of the user's emotions by seeking the feedback of the user. Although such an approach is certainly very personable, unfortunately, according to [3], [16], the average human does not possess enough self-knowledge to be able to clearly express at any given time what her emotional state is. We believe that the system's quality will benefit from mechanisms that can acquire a person's emotional state automatically. Several such acquisition mechanisms exist (for example, the devise described in [17]), although all of them are rather impractical to use in outside-the-lab conditions. However, for developing the mini system in step (A.2) of Algorithm A, such devises may be useful. Such devises can be proven especially useful for steps (A.3) and (A.4) of Algorithm A, during which the measurement of only one emotion attribute is required. In the future, we plan to experiment with such devises.

6. References

[1] Picard, R. W., "Toward computers that recognize and respond to user emotion", IBM Systems Journal (2000). Vol. 39 (No. 3 & 4), pp. 705-719.

[2] Picard, R. W., "Affective Computing", The MIT Press, 2000.

[3] Goleman, D., "Emotional intelligence", Bantam Books, 1995.

[4] Minsky, Marvin, "The Emotion Machine", Simon & Schuster, 2006.

[5] Liu, H. Lieberman, H., Selker, T., "A Model of Textual Affect Sensing using Real-World Knowledge", Proc. of IUI 2003, Miami, FL, USA, 2003, pp. 125-132.

[6] Anestis A. Toptsis and Alexander Dubitski, "Organization and retrieval in affectively annotated K-line indexed media repositories", Proc. Software Engineering and Applications, SEA 2008, Orlando, FL, USA, 2008, pp. 148-153.

[7] Anestis A. Toptsis and Alexander Dubitski, "Iterative K-line Meshing via Non-Linear Least Squares Interpolation of Affectively Decorated Media Repositories", The Open Artificial Intelligence Journal, *in press*.

[8] Elliott, G.T. and Tomlinson, B., "Personal Soundtrack: Context-aware playlists that adapt to user pace", Proc. CHI 2006, Canada, ACM Press, pp. 736-741.

[9] Dornbush S., English J., Oates T., Segall Z., Anumpam J., "XPod: A Human Activity Aware Learning Mobile Music Player", Proc. Workshop on Ambient Intelligence, 20th International Joint Conference on Artificial Intelligence, 2007.

[10] Ekman, P., "Facial expression of emotion", American Psychologist, 48, 1993, pp. 384-392.

[11] Emotion Classification (accessed September 14, 2008) http://en.wikipedia.org/wiki/Emotion_classification.

[12] Minsky, Marvin, "K-Lines: A Theory of Memory", Cognitive Science, 4, 1980, pp. 117-l33.

[13] Minsky, Marvin, "The Society of Mind", Simon & Schuster, 1986.

[14] Ekman, P., "Basic emotions", in T. Dalgleish and T. Power (Eds.) The Handbook of Cognition and Emotion,. Sussex, U.K.: John Wiley & Sons, Ltd., 1999, pp. 45-60.

[15] Anestis A. Toptsis and Alexander Dubitski, "Heuristic Determination of the Most Influential Affective Attributes in K-line Indexed Media Collections", Proc. Distributed and Intelligent Multimedia Systems, DIMS 2008, Orlando, FL, USA, 2008, pp. 461-466.

[16] Goleman, D., "Social Intelligence", Bantam Books, 2006.

[17] Peter, C., Ebert, E., and Beikirch, H., "A wearable multi-sensor system for mobile acquisition of emotion-related physiological data", Proceedings of the First International Conference on Affective Computing and Intelligent Interaction: ACII 2005, Springer, 2005, pp. 691-698

[18] C.R. Rao, H. Toutenburg, A. Fieger, C. Heumann, T. Nittner and S. Scheid, “Linear Models: Least Squares and Alternatives”, Springer Series in Statistics, 1999.

[19] T. Kariya, H. Kurata, “Generalized Least Squares”, Wiley, 2004.

[20] J. Wolberg, “Data Analysis Using the Method of Least Squares: Extracting the Most Information from Experiments”, Springer, 2005.