On the Affective Bias of Emotional and Physiological States During Handling Media Content in K-line Mesh-Driven Repositories

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
doi: 10.4156/jdcta.vol3.issue1.toptsis

Abstract

The emerging paradigm of ubiquitous computing can be better realized in the presence of intelligent machines that adjust their operation based on the user's affectively driven behavior or disposition. In this work we provide a method that can be used to investigate how two types of affective attributes influence certain types of affective computing systems. Specifically, our method compares the suitability of emotion-based attributes versus the suitability of physiology-based attributes in affective computing systems that handle media content. We implement our method and use it to evaluate affective computing systems built for a variety of relevant affective attributes. Our findings indicate that, under certain conditions, the emotion-biased systems are more sensitive than the systems which are driven by purely physiology-based attributes.

Keywords

Artificial Intelligence, Affective Computing, Human-Computer Interaction, Multimedia, Ubiquitous Computing.

1. Introduction

In recent years research toward integration of computing with emotion and socio-cognitive processing gains significant momentum. It is increasingly recognized that the new paradigm of ubiquitous computing (many computing devises per person) [1] could be better realized in the presence of affectively aware machines that take in account the emotional state of a user and adjust their operation based on the user's behavioral patterns. These patterns are usually a combination of pure logic, emotional, and aesthetic considerations. As such, it is advantageous for a computing system to understand the background influences that generate such patterns and to behave accordingly. As pointed out more than 25 years ago from one of the fathers of AI,

"The old distinctions among emotion, reason, and aesthetics are like the earth, air, and fire of an ancient alchemy. We will need much better concepts than these for a working psychic chemistry." [2], page 1.

Affective Computing [3, 4], has emerged as a new discipline whose main purpose of to integrate affectively-laden information into computing systems. It is a new multi-disciplinary area that combines computer science, engineering, psychology, cognitive science, artificial intelligence, and human-computer interaction. Considering how much our every day lives revolve around consumption of information and media content, it is not surprising that one of the possibilities for Affective Computing projects pointed out in [3, 4] is to enable a machine to select and deliver media content (and especially music) according to users' emotions. Several approaches have been proposed to automate the delivery of media content. Examples are the projects described in [5, 6, 7, 8, 9, 10, 11, and 12]. In the next section we elaborate on these works and discuss how they connect with the work presented in this paper.

In this work we provide a method of how two types of affective attributes influence media handling. Specifically, our method compares the suitability of emotion-based affective attributes versus the suitability of physiology-based affective attributes, during handling of media content. Our findings indicate that under normal every-day life activities, emotional attributes are significantly more important than physiological attributes. The rest of the paper is organized as follows.

Section 2 discusses background work, provides a motivation for the work presented in this paper and presents our methodology. Section 3 presents an algorithm that we use for organization and retrieval of media content. This algorithm is a major tool that we use for carrying out the evaluation of our method.

Section 4 presents an evaluation of our method. Section 5 presents our conclusions and future research directions.

2. Background and Methodology

Based on existing literature, research works in media content handling can be classified into two main categories: *information-centric*, and *user-centric*, as shown in Figure 1. These categories are also referred to as content-based and context-based, respectively.

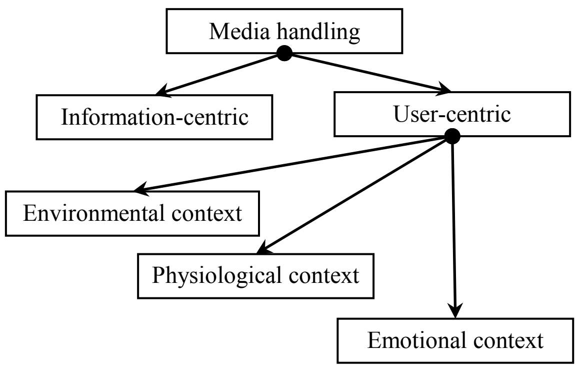

Figure 1. Approaches for handling media content.

In the *information-centric* approach, the interest is to develop systems that facilitate efficiency in data access, and/or content-based organization and retrieval, without taking in account any of the users' preferences toward particular types of content. Examples of such works are the projects presented in [5, 6]. In the *user-centric* approach the interest is to develop systems that take in account the user's state of being, in addition to whatever content access mechanisms are provided by the system. Examples of such reports are the early work of [7] and more recently the works in [8, 9, 10, 11, and 12]. All these works are user-centric approaches or, as often referred to, context-based approaches. In [8] the context is comprised of *environmental* and other attributes that, in general, a user cannot influence (current weather temperature, humidity, noise, user's gender and age, season and time of day). In [9, 10] the context is expressed mainly by the user's *physiological* state of being, and media selection activity is monitored based on the user's pace while performing some physical activity. Finally, in [11, 12] the context is expressed by the user's *emotional* state and media selection activity is monitored based on the intensity of certain user's emotions, such as sadness, happiness, fear, etc. (the standard basic emotion classification of Ekman [13] is used). Clearly, both the physiological *and* the emotional state of a person can be influenced – at least to a certain degree, by environmental factors (such as weather conditions or time of day) and, moreover, they are both states that can be influenced *by the person* as well! For example, a person can increase her heart rate (physiological attribute) by running faster, or by climbing stairs. Likewise, the intensity of a person's disgust (emotional attribute) will likely be increased if she experiences media content that communicates a violent or otherwise distasteful activity.

In this paper we present an algorithm that organizes a media collection and subsequently provides a retrieval mechanism that extracts and delivers media content from the organized repository, *for the purpose of investigating the influence and suitability of emotional attributes versus physiological attributes in such an environment*. Both the organization and retrieval are based on the affective state of the user, which can be expressed either through emotional or through physiological attributes. Our purpose is to evaluate our algorithm's performance for cases when the context is emotionally biased versus cases when the context is physiologically biased. Algorithm M, shown next, outlines our methodology.

Algorithm M

Step M1:

Define $AA = \{a_i\}$, i=1, …, K, the set of all affective attributes used in this algorithm. Without loss of generality, assume

$$AA = AAE \cup AAP \text{ and } AAE \cap AAP = \varnothing$$

with

$$AAE = \{a_1, a_2, ..., a_w\},\ AAP = \{a_{w+1}, a_{w+2}, ..., a_K\}$$

where $a_1, a_2, ..., a_w$ are all emotionally based affective attributes, and $a_{w+1}, a_{w+2}, ..., a_K$ are all physiologically based affective attributes.

Step M2:

`for i=1, 2, …,w`

Develop and evaluate system ASE_i, based on attribute a_i. Note, the context in all ASE_i systems is *emotionally* biased, since all a_i's are emotion-based attributes.

Step M3:

`for j=w+1, w+2, …,K`

Develop and evaluate system ASP_j, based on attribute a_j. Note, the context in all ASP_j systems is *physiologically* biased, since all a_i's are physiology- based attributes.

Step M4:
Compare the performance of systems ASE_i from step M2, with the performance of systems ASP_j from step M3.

Step M5:
Based on the findings from step M5, determine which, if any, of the two types of system (ASE_i of step M3, or ASP_j of step M4) is better.

In *Step M1* of algorithm M, we decide *what* affective attributes to test for suitability for a given affective system. Based on the earlier discussion, we are interested in evaluating the suitability of attributes that are either physiological or emotional in nature. Therefore, the set AA of all selected attributes contains only those two types of attributes. Note, also, that the subsets AAE and AAP form a partition of set AA. This means that if an attribute can be of both emotional and physiological nature, this attribute is not selected at all in our set AA, or else the intersection of AAE and AAP would not be empty. The same applies for attributes whose intensity value exhibits variations in different parts of the human body, and for which the variations could be due to either physiological or emotional factors. An example of such an attribute is the "skin temperature", as this attribute is obviously of physiological nature, but it is also known to be of emotional nature and it also exhibits variations in its value at different parts of the human body (e.g., the "cold-feet" condition) [14].

In *step M2* and *step M3* of algorithm M we decide *how* to test the affective attributes that are selected in step M1. In doing so, we develop systems ASE_i (i=1,…,w) and ASP_j (j=w+1,…,K) and evaluate the suitability of each attribute using those systems. Each of the ASE_i and ASP_j systems is a repository of media content, organized in a certain way (described in Section 3) after annotating its contents with the affective attribute under consideration. Once the repository is organized, a retrieval mechanism (also described in Section 3) is employed to deliver media content, based on the same affective attribute that is used to organize the repository. The suitability of the affective attribute is determined by the reported user's satisfaction regarding the retrieved media content.

In our algorithm, the affective state is a generic placeholder and it can be replaced by either an emotional attribute (such as "happiness" or "sadness"), or by a physical attribute (such as "heart rate"), depending on what kind of state of being of the user need be investigated in every occasion. Our method is based on an AI-related data structure called K-lines, introduced by Minsky and refined over the last 30 years in [15, 16, 17], and it is an adaptation of methods presented by the authors in [11, 12]. Notably, the methods presented in [11, 12] do not take in account any physical attributes and they are less flexible than the algorithm presented here, since they assume that the state of being of the user is captured by the *entire* spectrum of attributes borrowed from an off-the-shelf emotion classification [13]. The main ingredient of each K-line is a *K-node*. The concept of K-node is first introduced in [11] and it is expanded and formalized in [12]. Here, we adapt the structure used in [12] to fit the purposes of the method presented by algorithm M above. Figure 2 shows the K-node used in this paper.

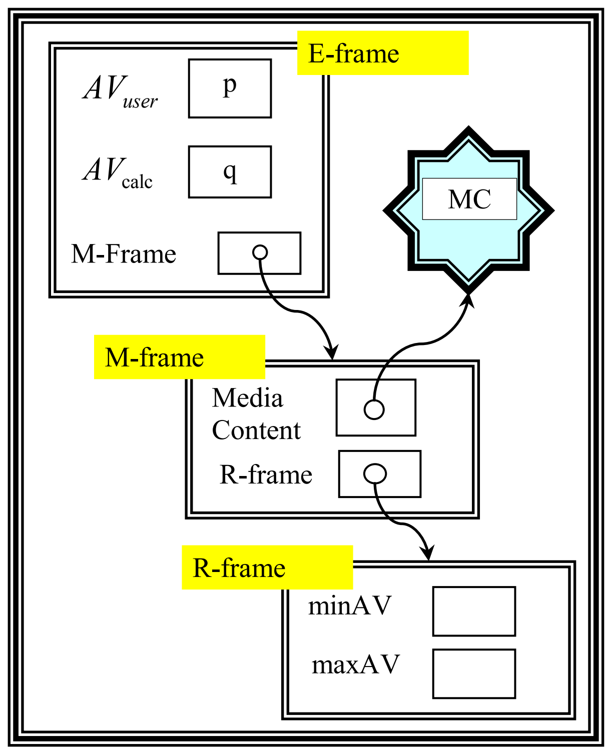

Figure 2. A K-node

As shown in Figure 2, a K-node is a multi-frame structure consisting of 3 *frames*: the E-frame, the M-frame, and the R-frame. A frame is an AI-related structure introduced by Minsky [18]. As described in [18], a frame can contain several placeholders – such as AV_{user} and AV_{calc} of the E-frame. Some of those placeholders can be frames themselves – such as the M-frame placeholder inside the E-frame, and the R-frame placeholder of the M-frame. The placeholders AV_{user} and AV_{calc} in the E-frame, are scalars with values such as the values p and q shown in Figure 2. The value of AV_{user} represents the intensity of an effective attribute, as acquired from the user. For example, for the affective attribute "sadness", AV_{user} p = 3.12 would mean that the user-acquired value for "sadness" has intensity 3.12 within a certain predetermined scale (in our implementation a scale from 0 to 5 is used, 0 representing the lowest possible intensity and 5 representing the highest possible intensity). The AV_{calc} placeholder (initialized with value q in Figure 2) represents the intensity of the same affective attribute (i.e., the attribute that is also associated with AV_{user}) but it is calculated automatically by our algorithm instead of being acquired from the user. The idea is that as K-nodes are formed, it is typically inefficient to initialize AV_{user} during the creation of *every* K-node. As such, our K-node formation algorithm (described below), acquires AV_{user} only periodically and then initializes AV_{calc} automatically for all K-nodes. The M-frame in Figure 2 is a sub-frame of E-frame (as indicated by the pointer from the M-frame placeholder within the E-frame). The M-frame contains 2 placeholders – the Media Content and the R-frame. The Media Content is a pointer to a piece of media content, MC. This can be of any type of media, such as a song, a video clip, an image, a piece of text, and so on. The MC is depicted by the star-shaped icon in Figure 2. The R-frame is a sub-frame of the M-frame, as indicated by the pointer emanating from the R-frame placeholder of the M-frame. The purpose of the R-frame is to determine a range of AV values for which MC is *appropriate to be experienced*, i.e., either to be selected by the user (e.g., for listening in case of a song, or for reading in case of a paragraph of text, etc.) or to be retrieved by the system and delivered to the user. The criterion for a MC to be selected is that the interval of affective intensity values [minAV, maxAV] specified by the minAV and maxAV values of the R-frame that corresponds to the M-frame that points to that MC, covers the AV_{user} value specified in the E-frame of the same K-node. That is,

$$AV_{user} \in [\min AV, \max AV]$$

for that K-node.

3. Repository Organization and Retrieval

In this section we describe our method for handling media content. Based on the discussion in Section 2, we employ K-lines as our main structure. In what follows we describe an algorithm that is used to create K-lines that store users' past experiences in selecting media content, influenced by the users' affective state. The method presented in this section originates from our earlier work in [11, 12]. A preliminary version of this method appears in [11]. A modified and updated version appears in [12], where the method is described in considerable detail. The description in this section is based directly on our method in [12]. It is adapted here to facilitate the purposes of our proposal as outlined by algorithm M of section 2. The algorithm is comprised of three parts: Part I, part II, and part III. Part I is the training phase. During this phase, a set of K-lines is created. Each of those K-lines captures the users' past experiences of selecting particular media content. Once the set of K-lines of part I is formed, part II processes this set and morphs in into a *K-line mesh*. A K-line mesh is an undirected graph whose nodes and edges are formed by connecting the K-lines produced in part I in such a way that it captures the collective experiences of the users' across all training sessions of part I. Finally, part III describes the retrieval mechanism, i.e., how to navigate the K-line mesh and retrieve media content that conforms to users' particular affective states.

Part I: Training Phase.

The purpose of this phase is to form a set of K-lines, such as the one shown in Figure 3. This is done by a process of data collection and formation of K-nodes connected into K-lines, as described in Algorithm A, below.

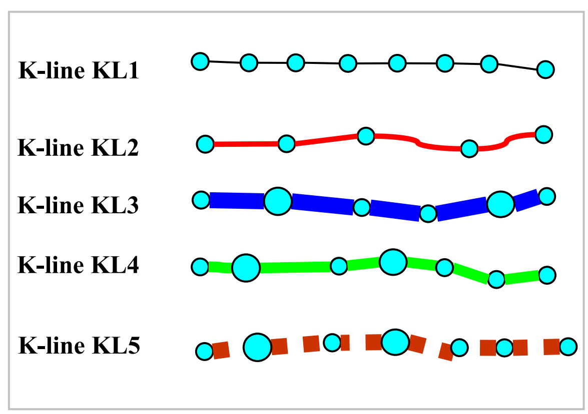

Figure 3. Five K-lines

Each K-line of Figure 3 contains several nodes (the blue-shaded circles), connected in a linked-list manner. Each node is a K-node and has the structure described in Section 2.

Algorithm A

```
1. Frequency = F;
2. nextKlineID = 1;
3. repeat {
       a. createKline( nextKlineID );
       b. nextKlineID++;
4. }until (system or user quits);
```

(method)

```
createKline( k ) {
1.  count = 0;
2.  repeat {
3.  user selects media content MC;
4.  if (isKlineIntersection(k, MC)){
        a. user enters AV_user;
        b. count = 0;
5.  }
6.  elseif ( count % F == 0 ) {
        a. user enters AV_user;
        b. count = 0;
7.  }
8.  else { AV_user= null; }
9.  createKnode( AV_user, null, MC, null ,null);
10. count++;
11. } until (system or user quits) ;
12. } // end createKline( k )
```

(method)

createKnode(AV_{user}, AV_{calc} ,MC,minAV,maxAV**)**

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

(method)

isKlineIntersection (k, MC))

- **Input:** a K-line ID k and MC.
- **Returns**: *True*, if MC 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.

Part II: Formation of K-line mesh

At the end of Part A, there are several K-nodes with their AV_{user} initialized, but there are also many K-nodes with AV_{user} = null (those are the K-nodes for which the user is not prompted for an AV). Also, at the end of Part A, the AV_{calc} and the values minAV and maxAV in the R-frame are not initialized for any of the K-lines. The purpose of Part II is to assign values for all values AV_{calc}, minAV, and maxAV, for all K-nodes throughout the K-line collection. Part II is performed automatically, i.e., without any user interference, and it is an iterative process, as illustrated by Algorithm B.

Algorithm B

repeat {

- **Step B1:** Initialize AV_{calc} for all K-nodes.
- **Step B2:** Initialize the *R-frame* for all K-nodes.
- **Step B3:**: Update the AV_{calc} values of step B1.

}

until (no significant changes are observed in the R-frame values**)**;

Step B1: **Initialize AV_{calc} for all K-nodes of all K-lines.**

In this step, we calculate AV_{calc} for all K-nodes of every K-line. Denote by p_{jx} the desired AV_{calc} value of the x-th K-node (x=1, 2, …) of the j-th K-line (j=1, 2, …). The sought values in p_{jx} are calculated as

$$p_{jx} = A + B \cdot M_x + C \cdot M_x^{\,2} \qquad \textbf{(1)}$$

where A, B, and C in expression (1) are given by the following expressions:

$$A = \begin{vmatrix} \mathrm{M}^{(0)} & \Phi^{(1)} & \Phi^{(2)} \\ \mathrm{M}^{(1)} & \Phi^{(2)} & \Phi^{(3)} \\ \mathrm{M}^{(2)} & \Phi^{(3)} & \Phi^{(4)} \end{vmatrix} \cdot \left(\Delta^{|0,4|}\right)^{-1}$$

$$B = \begin{vmatrix} \Phi^{(0)} & \mathrm{M}^{(1)} & \Phi^{(2)} \\ \Phi^{(1)} & \mathrm{M}^{(2)} & \Phi^{(3)} \\ \Phi^{(2)} & \mathrm{M}^{(3)} & \Phi^{(4)} \end{vmatrix} \cdot \left(\Delta^{|0,4|}\right)^{-1}$$

$$C = \begin{vmatrix} \Phi^{(0)} & \Phi^{(1)} & \mathrm{M}^{(1)} \\ \Phi^{(1)} & \Phi^{(2)} & \mathrm{M}^{(2)} \\ \Phi^{(2)} & \Phi^{(3)} & \mathrm{M}^{(3)} \end{vmatrix} \cdot \left(\Delta^{|0,4|}\right)^{-1}$$

$$\Delta^{|0,4|} = \begin{vmatrix} \Phi^{(0)} & \Phi^{(1)} & \Phi^{(2)} \\ \Phi^{(1)} & \Phi^{(2)} & \Phi^{(3)} \\ \Phi^{(2)} & \Phi^{(3)} & \Phi^{(4)} \end{vmatrix} \neq 0 .$$

The right hand sides of the above expressions are determinants which are produced by applying Cramer's

Rule on the system of equations (2), shown next. (Cramer's Rule is a standard result in linear algebra which gives the solution of a system of linear equations [19])

$$A \cdot \Phi^{(w)} + B \cdot \Phi^{(w+1)} + C \cdot \Phi^{(w+2)} = \mathrm{M}^{(w)}$$

$$\text{where, } \Phi^{(w)} = \sum_{i=1}^{n} i^{w}, \text{ and } \mathrm{M}^{(w)} = \sum_{i=1}^{n} M_i \cdot i^{w},$$

$$\text{for } w = 0, 1, 2. \qquad \textbf{(2)}$$

The system of equations shown in expression (**2**) is produced by applying the Least Squares Method (LSM) on each of the K-lines resulting from Part I. The LSM is a mathematical method introduced by Gauss circa 1794 and it is since updated and typically used for regression analysis. The interested reader can find descriptions of this method in various textbooks, including [20, 21, 22]. In the above system of equations, M_i is the AV_{calc} value of the i-th K-node that is populated with a AV_{calc} within its K-line. Note, n is less than or equal (in most cases, less) than the total number of K-nodes of this K-line, since not all K-nodes have a user assigned AV_{calc} at the end of Part I. The LSM process yields a curve such as the one expressed by equation (**1**), and such that the distances between that curve and the 2-dimensional points (i, M_i) are minimized. This means that the left hand side, E, of expression (**3**) below, attains its minimum value.

$$E = \sum_{i=1}^{n} \left[M_i - (A + B \cdot M_i + C \cdot M_i^{2}) \right]^2 \qquad \textbf{(3)}$$

This, in turn, means that the derivatives of E with respect to A, B, and C, are zero, which yields the system of linear equations (with A, B, and C being the unknowns) given in expression (2) above, and consequently, the solution of the expressions for A, B, and C, as shown above.

Step B2: Initialize R-frame.

At the end of step B1, every K-node in our K-lines has associated with it either one AV (the AV_{calc}, calculated by the interpolation process) or two AVs (the AV_{calc} calculated by the interpolation process and the AV_{user} as originally entered by the user). The purpose of this step is to create a *range* of appropriate values by initializing the minAV and maxAV fields of the R-frames of all K-nodes. Recall, the meaning of the R-frame is that a media content MC is appropriate for delivering it to the user, if the user's AV is within the range [minAV, maxAV]. At the beginning of this step (step B2), none of the R-frames throughout the K-line collection formed in step B1 has its minAV and maxAV values initialized. The first thing done by this step is to initialize all those values. Moreover, once those values are initialized, we face an additional issue: it is possible that the [minAV, maxAV] ranges of K-nodes that point to the same media content, are *different*. This is because all R-frames are associated with K-nodes hosted in *different* K-lines. As such, the interpolation process of step B1 produces, in general, different AV_{calc} values for those K-nodes. As a result, those different AV_{user} values will contribute in producing different minAV and maxAV values during this part of step B2. Note, such K-nodes are at the intersection of their hosting K-lines. Figure 4 illustrates this situation.

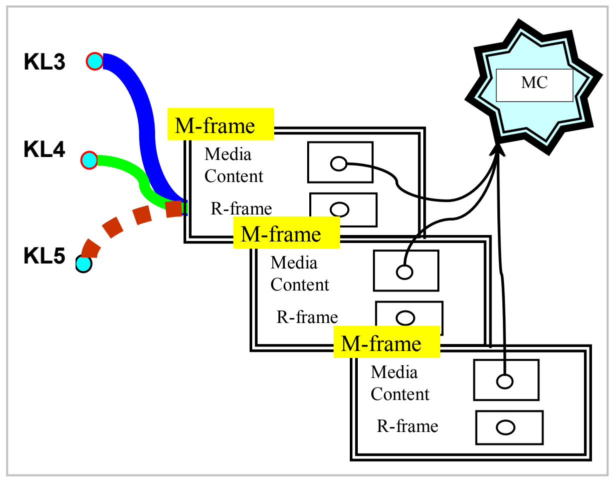

Figure 4. Three different K-nodes, at K-line intersection.

To remedy the above concerns, we define an *inter-K-line R-frame width*, W_{R}. W_{R} is a number that represents the ideal width of a R-frame and serves as a guideline that all intervals [minAV, maxAV] throughout the K-line collection try to achieve. If the size of an interval [minAV, maxAV] is bigger (or, smaller) than W_{R}, we shrink (or, expand, respectively) [minAV, maxAV] so that its resulting size is equal as much as feasible (subject to the conditions below) to W_{R}. The shinkage and expansion of any interval [minAV, maxAV] results to an interval I_{R}. I_{R} is subjected to the following conditions:

1. I_{R} should include all values AV_{user} of the K-nodes that are at the intersection of the K-lines under consideration.
2. I_{R} should not be narrower than W_{R}.

3. Preferably, I_R should be symmetric with respect to the interval $[\mu_1, \mu_2]$, where $\mu_1 = \min\{AV_{user}, AV_{calc}\}$ and $\mu_2 = \max\{AV_{user}, AV_{calc}\}$. That is, each each of the left and right boundaries of I_R should be equidistant from the corresponding left and right boundaries of $[\mu_1, \mu_2]$.
4. condition (3) should not contradict any of conditions (1) and (2).

Step B3: update the AV_{calc} values of step B1

In step B3 we adjust, as necessary, the AV_{calc} values calculated in step B1. Specifically, we adjust the AV_{calc} value of each K-node to fall inside the corresponding R-frame interval [minAV, maxAV] of that K-node. Note, at the end of step B2, the R-frame interval [minAV, maxAV] of each K-node is initialized. For any K-node N, denote:

- $N.AV_{calc}$: the AV_{calc} value of node N.
- $N.\min AV$: the minAV value of the R-frame of node N.
- $N.\max AV$: the max value of the R-frame of node N.

We adjust $N.AV_{calc}$ as follows:

- if $N.AV_{calc}$ < $N.\min AV$, then $N.AV_{calc} = N.\min AV$.
- If $N.AV_{calc}$ > N.maxAV, then $N.AV_{calc}$ = N.maxAV.

In other words, for every K-node, we adjust the AV_{calc} value of that K-node to fall inside the corresponding [minAV, maxAV] interval of the R-frame of that K-node. The motivation for doing so is to improve on the estimations of the AV_{calc} values that result from step B1. Note, the quality of the AV_{calc} values resulting from step B1 depends on the quality of the curve calculated in that step (as described in step B2 On the other hand, the R-frame interval [minAV, maxAV] associated with a K-node and a piece of media content, as calculated during step B2, represents the overall (collective) knowledge of all K-nodes that are at the intersection of their respective K-lines. Therefore, if a LSM-calculated value AV_{calc} is outside the corresponding R-frame interval values, this constitutes evidence that AV_{calc} is possibly off and thus it would be prudent to adjust it.

This is exactly what is done in this step! Note, the updated AV_{calc} values are used for the next iteration of Algorithm B, in step B1.

Steps B1, B2, and B3 or algorithm B are repeated until two successive iterations produce negligible changes in their R-frame intervals, throughout the K-line collection. When this happens, the system is considered to be trained and the K-line mesh to be fully formed. Figure 5 shows a K-line mesh formed at the end of Algorithm B.

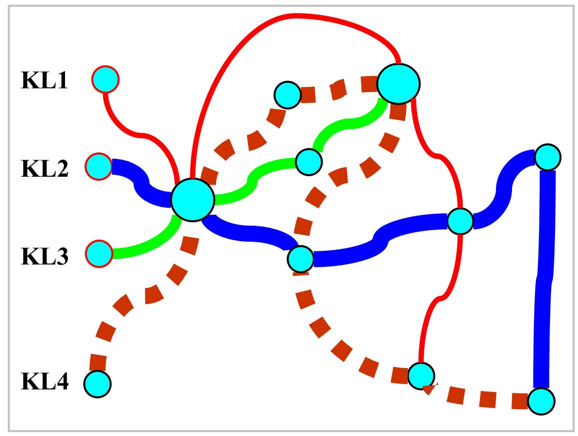

Figure 5. A K-line mesh

Each of the nodes at the intersections of K-lines in Figure 5 are K-nodes whose M-frames point to the same piece of media content and their R-frames have identical [minAV, maxAV] intervals.

Part III: K-line mesh navigation

We now describe the process of retrieving media content from the formed K-line mesh. The process is divided into 4 steps.

Step 1: The system acquires an affective value AV_{input} from the user.

Step 2: The system chooses a K-line that is a closest match to the input AV_{input}. To determine the closest matching K-line the system calculates the distance between AV_{input} and each of the 1st K-nodes of the existing K-lines and chooses the K-line that produces the shortest Euclidean distance between AV_{input} and the 1st K-node of each of the K-lines. This distance is given by the expression

$$dist(KL, AV_{input}) = \frac{1}{2} \cdot \sqrt{(p+q-2 \cdot w)^2} \quad \textbf{(4)}$$

where p and q are the values of AV_{user} and AV_{calc} of the 1st K-node of K-line KL and w is the value of AV_{input}.

Step 3: Once the best matching K-line is determined, the system starts from the 1st K-node of that K-line and retrieves the media content pointed to by that node. Then it traverses that K-line sequentially and retrieves only the media content of those nodes for which the R-frame intervals [minAV, maxAV] contain the afective value AV_{input}. Figure 6 illustrates two pieces of media content retrieved during Step 3.

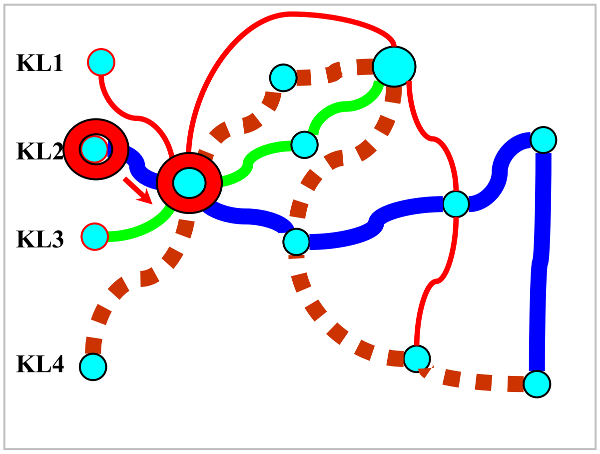

Figure 6. K-line mesh navigation

In Figure 6, K-line KL2 is the closest matching K-line (as found in step 2) and then the first and second K-nodes of that K-line are retrieved. (the retrieved nodes are marked with heavy (red) circles and the progress of the retrieval algorithm is indicated with the (red) arrows along the edges of the K-line mesh).

Step 4: This step is performed as part of step 3. As the iterator moves from the current K-Node to the next K-node during step 3, the system checks if the currently visited K- node is at the intersection of two or more K-lines. If this is so, then the system decides to which K-line among the intersecting K-lines to move next. This decision is made similar to step 2 above. That is the system calculates the distance between the AV_{input} and each of the K-nodes that are candidates to be selected next. Then, it resumes by following the K-line that contains the K-node that produces the minimum distance. Figures 7 and 8 illustrate the progress of the retrieval algorithm after an intersection of K-lines has been encountered in Figure 6.

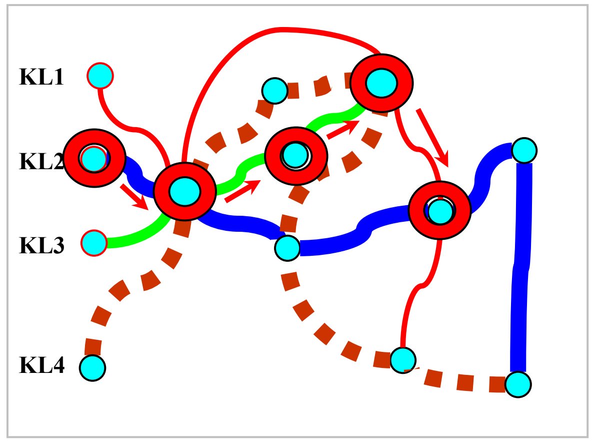

Figure 7. K-line mesh navigation progress I

As illustrated in Figure 7, after the retrieval algorithm encounters a K-line intersection in Figure 6 (the intersection of all 4 K-lines KL1, KL2, KL3, and KL4) it decides to continue and retrieve the next MC along K-line KL3. Apparently, the retrieved K-node in K-line KL3 contains a MC which produces the closest match between AV_{input} and the MCs of all other candidate nodes which are immediate successors of the retrieved K-node of Figure 6.

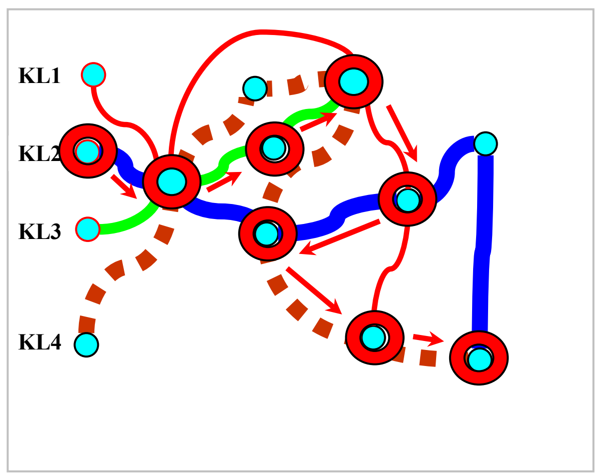

Figure 8. K-line mesh navigation progress II

Figure 8 is continuation of Figure 7 and shows the progress of the retrieval algorithm after Figure 7. Similar to Figure 7, in Figure 8 the algorithm encounters another K-line intersection and it decides to move accordingly, as indicated by the progress arrows.

4. Evaluation

We evaluate the method described by Algorithm M of section 2 for a media collection.

The setting of our evaluation is a *normal every-day routine* environment. That is, our evaluation is performed when the user is under non-extreme conditions, and is done when a user experiences media content while performing a normal everyday-life routine type of activity, such as relaxing, or browsing the web. We choose this setting because we want to avoid (as much as possible) emotional and physiological fluctuations that may be induced by *extreme unrelated* types of activity and thus contaminate our experiments. (An example of "extreme" activity that could cause unusual fluctuation of emotion-based (and, most likely, physiology-based) attributes is when a user reads a RSS-feed that announces that her stock portfolio declined by 30% over the last hour. Another example of an "extreme" activity is when the user is engaging in some form of fitness exercise, or vigorous household maintenance activity – those would most likely cause escalation of the intensities of *all* the physiology-based attributes.)

For step M1 of Algorithm M (our methodology from section 2), we choose as AA the following set:

AA = { *Amusement, Excitement, Sadness, Happiness, Anger, Relief, Embarrassment, Pleasure, Satisfaction, Pride, Heart Rate, Systolic Blood Pressure, Diastolic Blood Pressure* } .

To satisfy step M1 of algorithm M, the set AA is partitioned into two sets AAE and AAP, as follows:

- **AAE** = { *Amusement, Excitement*, *Sadness, Happiness, Anger, Relief, Embarrassment, Pleasure, Satisfaction, Pride* },
- **AAP** = { *Heart Rate, Systolic Blood Pressure, Diastolic Blood Pressure* }

All affective attributes of AAE are emotion-based attributes, borrowed from the most recent Ekman's emotion classification [23]. The three attributes of set AAP are, in our opinion, among the most (if not, *the most*) representative and least evasive to measure physiology-base attributes in humans. Notwithstanding the century-old debate of whether or not, and how much, the emotional and physiological state of being interact and influence each other (see, for example, [24]), we believe that when a typical person performs non-strenuous types of activity, the disparity between sets AAE and AAP is large enough to warrant those forming a *partition* of set AA. This is important because, based on our earlier discussion in section 2, we like the attributes that we choose as emotion-based attributes to have as little interference as possible with the ones that we choose as physiology-based attributes. As much as possible, a chosen physiology-based attribute should not be a disguised emotion-based attribute, or visa-versa. We believe that the chosen attributes for sets AAE and AAP have these characteristics. Examples of attributes that would probably not be as appropriate to use are "skin temperature" (mentioned earlier in section 2), or "skin resistance". These two attributes are usually considered to be physiology-based attributes; nevertheless as it is also commonly accepted, they both have high correlation with emotional states as well.

For step M2 and step M3 of algorithm M, we use the set AAE to form and evaluate systems ASE_i , and the set AAP to form and evaluate systems ASP_j, respectively. The algorithm described in Section 3 is used to form and evaluate the K-line meshes for those systems. Note, during part I (algorithm A) and part II (algorithm B) of the method described in section 3, the AV_{user} values and the AV_{calc} values are initialized and modified at various stages. In order for the method to be fair for both sets AAE and AAP (and the corresponding systems ASE_i and ASP_j) we use the *same scale* of values for the intensities of both the emotion-based and the physiology-based attributes. Following our previous works ([11, 12, 25, and 26]) with emotion-based attributes, we use a scale 0 to 5 for the intensity values of the attributes of set AAE. This necessitates that the same scale is used for the attributes of set AAP. Note, all 3 attributes of set AAP are heterogeneous, in the sense that their typical intensities come both, in different units of measurement and different value ranges. Table 1 shows the corresponding units of measurement and some commonly accepted intensity value ranges for these attributes.

Table 1. Typical ranges of the AAP set attributes and their units of measurement.

Attribute	*Typical range*	*Unit*
Heart Rate (HR)	**60-120**	**bps**
Diastolic BP (DBP)	**60-100**	**mmHg**
Systolic BP (SBP)	**100-180**	**mmHg**

We convert any intensity value V of any of the three attributes of Table 1 to a value within the range [0..5] by multiplying V by a constant c_{HR} , or c_{DBP} , or c_{SBP} , where

$$c_x = \frac{5}{\max\left\{p_x\right\}}, \; x = HR, or\; DBP, or\; SBP \, ,$$

and p_x is an intensity value recorded for attribute x. Note, if V_x is an intensity value for the physiology-based attribute x, then $V_x \cdot c_x \in [0..5]$.

Similar to previous works ([10, 11, 12, and 25]) we choose a collection of songs as our test bed media collection. Our collection consists of 300 songs, of which 75 are randomly chosen to build systems ASE_i and ASP_j, mentioned in steps M2 and M3 of algorithm M of Section 2. We use the algorithm described in Section 3 to build and evaluate all those systems. The collection as well as the algorithm used to construct the affective systems ASE_i and ASP_j are the same as the ones we use in [26]. Note, we choose only 75 songs (out of the 300 songs available in the collection) because according to our earlier findings from [25, 26], a fairly small subset of the entire collection is enough to provide extremely accurate indications for the suitability of the affective attributes used to build an affective system of this nature. Table 2 shows the results of our experiments.

Table 2. Suitability values

	$\sigma - values$	$\pi - values$
Pride	**11.9**	
Pleasure	**21.3**	
Embarrassment	**23.1**	
Happiness	**24.3**	
Satisfaction	**26.6**	
Anger	**31.8**	
Relief	**32.1**	
Amusement	**33.3**	
Sadness	**38.1**	
Excitement	**38.4**	
HR		**19.4**
Diastolic BP		**22.3**
Systolic BP		**24.7**

The left-most column of Table 2 shows the affective attributes from the sets AAE and AAP. The numbers in the right-most two columns of Table 2 are percentages that show the suitability values obtained for each attribute. Higher percentage values mean higher user satisfaction from the evaluation of the built systems. The $\sigma - values$ in the middle column of Table 2 are the suitability values of the emotion-based attributes; the $\pi - values$ in the right-most column of Table 2 are the suitability values of the physiology-based attributes. The same $\sigma - values$ are reported in our previous work in [26], except that are reproduced here in sorted order. For easier comprehension the results of Table 2 are also shown in Figure 9.

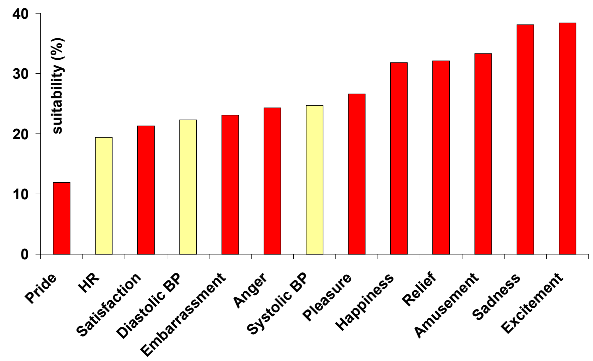

Figure 9. Results of Table 2.

In Figure 9 the darker (red) bars are the suitability values of the emotion-based attributes; the lighter (yellow) bars are the suitability values of the physiology-based attributes. As we can easily observe, the suitability values of the emotion-based attributes overwhelm the suitability values of the physiology-based attributes.

Note, based on our past work and recommendations in [26], the best performing emotion-based attributes would be chosen in building a production-grade affective system of the sort that is built here, and the worst-performing attributes would be rejected for being used in such a system. Also, based on our results in [12], the *cumulative* suitability derived from more than one attribute used in an affective system exceeds the individual suitability values attained by using *single* attributes. (Suitability values of more than 74% are reported in [12] for a combination of emotion-based attributes and this is so even while it is doubtful that all those attributes contribute positively in the system under discussion).

The *average* suitability (about 22%) of *all* physiology-based attributes is very close to reported performance of systems in which *no* affective attributes at all are taken in account. In such systems, playlists are assembled by random song selection, without any consideration of the user's affective state. Such results are reported in both [10, 12].

5. Conclusion

We present a method (algorithm M, in section 2) for evaluating the impact of emotion-based and physiology-based attributes in an affective computing system. We implement our method and use it to evaluate affective computing systems built for a variety of emotion-based attributes (10 emotion-based attributes from Ekman's most recent emotion classification) and the most representative physiology-based attributes of "heart rate", "diastolic blood pressure", and "systolic blood pressure".

Our findings indicate that in the vast majority of cases the emotion-based attributes are far more meaningful in terms of creating systems that correlate well with users' interaction and also such attributes contribute positively in increasing users' satisfaction during usage of such systems. The same cannot be said for the physiology-based attributes! In fact, the systems built with physiology-based attributes perform close to the *random* playlist generating paradigm, i.e., in the neighborhood of 20%. This is indication that physiology-based attributes have much less impact than emotion-based attributes when building affective media systems that are to be used under normal, non-strenuous, everyday-life activities.

Based on our findings, we conclude that: (a) in building affective media systems to be used during normal, everyday-life, non-strenuous activities of a person, the use of emotion-based attributes is far more preferable than the use of physiology-based attributes, and (b) the contribution of physiology-based attributes is, at best, very inconclusive (since such systems that employ physiology-based attributes perform very close to random).

We welcome these findings since they provide an educated guideline that further research on the impact of the emotional state of a person is likely to be fruitful in the area of affective computing. This is important especially at times that we witness the continuous popularity rise of the ubiquitous computing paradigm. It is comforting to know that some form of built-in emotion-driven intelligence in the myriad of computing devises that aspire to manage and even dictate so many aspects of our lives, is not totally beyond our grasp.

In the future we are interested to investigate the impact of emotion-based attributes on different types of media content, especially in written and spoken text. There has been some reported research on annotating text based media with emotion-based attributes (e.g., [27]). However, we are not aware of any affective systems like the ones we describe here, whose media type is text. Seeing that a fairly large portion of a person's daily life is consumed in verbal communication (such as listening to news, talking with peers, etc.), and/or reading, we believe that such systems would be useful.

6. References

[1] Mark Weiser, "Ubiquitous Computing", http://www.ubiq.com/hypertext/weiser/UbiHome.html (last accessed Dec. 5, 2008).

[2] Marvin Minsky, "Music, Mind, and Meaning", MIT-AI Laboratory Memo 616, Fall 1981, http://web.media.mit.edu/~minsky/papers/MusicMindMeaning.html (last accessed Dec. 10, 2008).

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

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

[5] H. Mandviwala, S. Blackwell, C. Weikart, and J. Van Thong, "Multimedia content analysis and indexing: evaluation of a distributed and scalable architecture", ITCom 2003, Volume 5242 -- Internet Multimedia Management Systems IV, pp. 137-145, 2003.

[6] Lloyd Rutledge, Jacco van Ossenbruggen, and Lynda Hardman, "Structuring and presenting annotated media repositories", Proceedings of the 13th International World Wide Web conference on Alternate track papers & posters, ACM press, pp. 466 – 467, 2004.

[7] F.-F. Kuo and M.-K. Shan, "A personalized music filtering system based on melody style classification", Proc. IEEE Int. Conf. on Data Mining, pp. 649-652, 2002.

[8] Han-Saem Park, Ji-Oh Yoo, and Sung-Bae Cho, "A Context-Aware Music Recommendation System Using Fuzzy Bayesian Networks with Utility Theory", International Conference on Fuzzy Systems and Knowledge Discovery, FSKD 2006, Springer-Verlag LNAI 4223, pp. 970 – 979, 2006.

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

[10] S. Dornbush, Jesse English, Tim Oates, Zary Segall, and Joshi Anumpam, "XPod: A Human Activity Aware Learning Mobile Music Player", Proc. Workshop on Ambient Intelligence, 20th International Joint Conference on Artificial Intelligence, 2007.

[11] 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, pp. 148-153, 2008.

[12] 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, Vol. 2, pp. 46-61, 2008.

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

[14] Glenn Elert, "Temperature of a Healthy Human (Skin Temperature)", http://hypertextbook.com/facts/2001/AbantyFarzana.shtml, (last accessed Dec. 7, 2008).

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

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

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

[18] Marvin Minsky, "A Framework for Representing Knowledge – FRAMES", MIT-AI Laboratory Memo 306, June 1974, http://web.media.mit.edu/~minsky/papers/Frames/frames.html, (last accessed Dec. 10, 2008).

[19] "Cramer's rule", http://en.wikipedia.org/wiki/Cramer's_rule. (last accessed Dec. 1, 2008).

[20] Otto Bretscher, Linear Algebra With Applications, 3rd ed., Upper Saddle River, NJ, USA, Prentice Hall, 1995.

[21] 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.

[22] T. Kariya, H. Kurata, Generalized Least Squares, Wiley, 2004.

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

[24] W. James, "What is emotion?" Mind 9, pp. 188-205, 1884. Also available at http://psychclassics.yorku.ca/James/emotion.htm (last accessed Dec. 5, 2008).

[25] 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, pp. 461-466, 2008.

[26] Anestis A. Toptsis and Alexander Dubitski, "Affective Space Calibration in Action-rich Media Affective Systems", Journal of Convergence Information Technology, (in press), 2008.

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