Synesthetic Generation of Sound Clouds by Applying Social Computing

María Navarro-Cáceres*, Lucía Martín*, Javier Bajo†, Yves Demazeau‡ and Juan Manuel Corchado*
*Department of Computer Engineering and Automatics. University of Salamanca. Salamanca, Spain
Email: {maria90,luciamg,corchado}@usal.es
†Universidad Complutense of Madrid. Madrid, Spain
Email: jbajope@usal.es
‡Univ. Grenoble Alpes, CNRS, LIG
F-38000 Grenoble, France
Email: Yves.Demazeau@imag.fr

***Abstract*—The aim of social computing is to analyse the concept of social nature and design digital systems that share information between machines and users. The insights given by social computing can be applied to easily construct creative systems. As a case study, this paper presents a social machine implemented as a virtual organization where humans and machines collaborate in a creative process to transform a picture into a musical sound cloud. The prototype built from this model is evaluated by experts who rate the sounds produced following tonal music criteria.**

Keywords*–*Synesthesia; Social Computing; Tonality; Music Generation; Sound Cloud

I. Introduction

Human beings are considered social creatures, always looking for interaction and communication with other people, and making decisions based on their social context. Social information given by such social contexts provides the basis for the inference, planning and coordination of any activity. However, this concept of a social environment cannot be translated into digital systems. In the digital world, we are socially blind [1]. Thus, the emergence of social machines has served to solve this problem and facilitate interaction and communication among people, to computerize aspects of human society, and to forecast the effects of technologies on social behavior [2].

Some authors have computed models of social intelligence based on social and psychological theories. Mission Rehearsal Exercises [3] or Tactical Language Training [4], [5] have implemented agents that develop social skills, such as leadership, foreign languages and culture in an artificial society. For example, the Sims 2 [6] is a popular game that models a virtual world with a social community. We can also consider interactive social robots, such as Teddy Bear, which was made by MIT Media Lab [7].

In the business area, the most widely used applications are recommendation systems, which suggest products, services and information to potential consumers. Companies, such as Amazon or Netflix, are adopting these systems [8] to improve customer loyalty. One approach is collaborative filtering to predict future sales by using historical sales transactions [9]. In the public sector, some government applications apply social computing to detect terrorist, criminal or other similar organizations [10], [11]. Social computing has also been applied to support decision making in health policy or state intervention [12].

With regards to music generation, some form of interaction between humans and machines is quite common. Martin *et al.* [13] presented the prototype software Toolkit to enable non-technical users to design artificial and intelligent agents to perform electronic music in collaboration with a human musician. Pachet *et al.* [14] developed The Continuator, a system able to interact with users to create a jazz improvisation in real time. Thorogood *et al.* [15] also present a system to generate soundscapes based on tweets about recent news items. There are examples of interaction with people to create different sounds or compositions through interactive evolution, such as Functional Scaffolding for Musical Composition technique [16] or neural nets [17]. Despite these proposed models of person-computer interaction, social machines have not yet been applied to this field to transform image into music.

In this work, we propose a system that supports communication among large groups of people over computer networks to generate creative content. In particular, this article focuses on uploading images by users to create a musical composition by translating colors into sound, imitating a neurological phenomenon known as synesthesia. The system is designed as a social machine described as an agent-based Virtual Organization (VO) where humans and machines collaborate in a creative process to transform a picture into a musical sound cloud. Agents start with an iterative process to extract sound from color and then generate a sound composition, denominated here as sound cloud, applying a swarm algorithm and following musical criteria such as consonance, distance between notes and distance to the main key. The prototype built from this model is evaluated by experts who rate the sound cloud or fragment produced by considering novelty and quality according to tonal music criteria.

A new architecture for creativity scenarios and an overall view of the system, based on social computing, is detailed in Section II, while the technical description of the workflow is given in Section III. Section IV presents the experiment carried out with the preliminary results obtained. Finally, Section V discusses the implications of the proposal and future work.

 ISBN: 978-1-61208-533-3

II. Social Machine with Virtual Organizations

Our aim is to develop a social machine capable of generating music from images provided by the users. An introductory description will be provided to give the reader a general idea about the system developed.

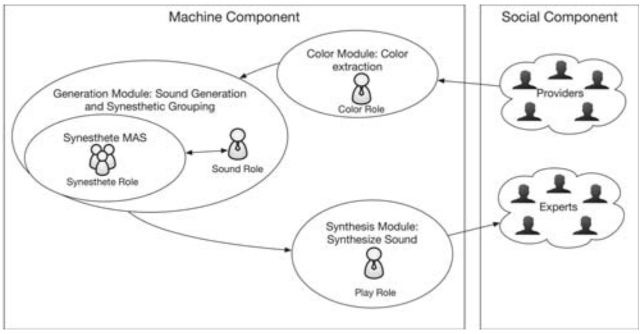

Figure 1. Social Machine schema, with human and machine components highlighted.

Fig. 1 represents a social model where human providers and experts (the social component) collaborate with the intelligent system (machine or software component). The social and software part of the system are highlighted in the squares. Each circle represents a particular stage in the workflow. Providers are focused on providing information about images, which is the input of the system. The color of the image is extracted in Hue, Saturation, Luminance (HSL) codification in the color extraction module. The HSL data of each color is then transformed into individual musical notes, and a swarm algorithm is applied to search consonant sounds, taking a sound as an individual particle with an associated color HSL, all of which occurs in the Sound Generation Module. The best sounds are selected and ordered in the Synesthetic Grouping Module following tonal music criteria. Finally, the sound is synthezised and played (Synthesizer Module) so that the Experts can evaluate the quality of the musical compositions.

The different steps described must be implemented using different intelligent modules. With an agent-based VO it is possible both to make a distributed system and easily integrate intelligent components, even combining different technologies or languages.

When developing the definition of the model used by the VO, it is necessary to analyze the needs and expectations of potential system users. The result of this analysis is the set of roles involved in the proposed model. Fig. 1 also shows the main roles identified in each previously described module; each role is represented by an agent picture.

- Provider: Represents the first part of the social machine. In this case, the user will be both the provider of the input image/picture and the listener of the final result.
- Color: Extracts colors from the image that will be associated to the Synesthete Agents.
- Synesthete: Transforms color into sound. To do so, each color is associated to a Synesthete agent. All the synesthete agents are particles immersed in a swarm algorithm that permits them to navigate through the space and change their knowledge about sounds as they are moving. At the end, different agents will be grouped according to their affinity with regards to musical aspects.
- Sound: Decides the order in which the groups of sounds corresponding to groups of synesthete agents will be played according to different parameters, such as consonance or melody leading.
- Play: Transforms the numerical notes into Musical Instrument Digital Interface (MIDI) information that can generate and play physical music.
- Experts: Given that creative process and products are hard to validate, various musical experiments have been rated following an expert evaluation. In this particular case, the output generated by the machine is evaluated by the expert interaction. The evaluation consists of rating each fragment generated on a scale from 1 (very bad) to 5 (very good). The social community can apply this form to study the quality of the compositions extracted by the system.
- Supervisor: The supervisor is a common agent in every VO. An agent who exercises this role will have overall control of the system. It analyzes the structure and syntax of all messages in and out of the system. As it is a technical agent not related with the main work, it is not represented in Fig. 1.

Section III explains the concrete implementation given for each role designed in the VO.

III. Machine Component Description

This section details the working flow of the machine component describing the algorithms and techniques implemented. It is divided into four subsections that describe the four stages of the system. Section III-A details the color extraction of the image provided by the users and the generation of the synesthete agents, which is essential to create music. Section III-B explains the interaction model among the synesthete agents. Section III-C describes a Sound Agent that groups and orders the sound created by the synesthete agents. Finally, Section III-D presents the synthesis process to play music, carried out by the Synthesizer Agent.

A. Color Extraction

The Color Role receives the digital image provided by the human, and then extracts the colors of the picture. In the first step, the Color agent creates a grid of cells as shown in Fig. 2.

The number of cells in the grid is set beforehand by the user, and must be an integer number lower than the number of pixels of the image. The color of each cell will correspond to the mean color between all the colors existing in the area studied. The HSL properties of the color are used to transform

Figure 2. The color extraction process shown in this figure consists of three main stages.

the color into sound to instantiate the Synesthete agents (Fig. 2).

Once the color has been extracted, Synesthete agents are created and placed into a 2D space in random positions. These agents can at a future time change their positions encoded as coordinates (x, y), where x and y are real numbers. Each Synesthete agent has the following properties:

- Color: this property consists of three attributes following the HSL model.
- Position: Considered a bidimensional position (x, y) as previously explained.
- Sound: Consists of note names in terms of loudness and pitch, as we will explain below.
- Associated Sound: Sound vector of the nearest agents.
- Velocity: An array with two vectors: One for the velocity in the X axis and another for the velocity of the Y axis.
- Sounding: A Boolean variable to store the decision about whether the sound is good enough to be played.

The color of each cell (in HSL model), produced by the Color agent, is transformed into a sound by these Synesthete agents. Both pitch and loudness can be linked with certain color properties. This relation can be established in different ways. A social interaction could be used in this stage to select the color for each note; however, for the scope of this study, a standard relation proposed by Sanz [18] was followed. First, Hue is automatically associated to Note name following the Lagresille system [18], where each set of color tones corresponds to a specific note. Then, Saturation is related to Loudness. We can consider this association to be logical, thus the more intense we see the color, the more intense the sound should be. The Saturation is translated into values of Loudness from 1 to 5, where 1 is very low and corresponds to a 0 of saturation and 5 is very high volume and corresponds to a saturation of 100%. Finally, Luminance refers to the Octave. As the luminance value is increased, the sound has a higher pitch, and will therefore be more acute. In order to preserve a balance between the coherence of different notes but also diversity in octaves, this is mapped from octave number 2 to octave number 6 according to the MIDI codification.

For instance, the first cell of Fig. 2 corresponds to a HSL codification of (242, 61, 52). That is translated into G note according to Lagresille System. The loudness corresponds to the value 3.44. The Luminance corresponds to the third octave, according to the integer mapping carried out.

B. Notes Grouping

The behavior of the synesthete agents that implement the Synesthete Role is based on a particle swarm optimization (PSO) algorithm [19]. Thus, the movement is regulated by attraction forces capable of modifying their position and velocity following a fitness function. The swarm allows the association of several agents with similar features following a fitness function, which will now be briefly described. The steps followed in this algorithm are:

1) Each agent has a position P in the 2D space, and can produce one sound from the color associated.
2) Each agent a_1 searches its neighbor agents a_2, a_3,..., a_N in the space based on Euclidean distance, and exchanges information with them to measure the quality between these sounds. This process, explained below, provokes an attraction force between the agents. The strength depends on the level of quality of the sounds.
3) These steps are repeated until a sound balance is found, upon algorithm convergence. Sound balance means that the particles do not update their positions significantly over the iterations. This indicates that the sounds are balanced in their right positions according to the quality function analyzed here.

In the final state, an agent organization with groups of pleasant sounds will be obtained.

As mentioned above, each agent rates its quality according to a fitness function. This function considers two musical factors to evaluate the quality of the sounds: consonance properties following the tonal standards and loudness, according to (1).

$$F(a, n) = \sum_{i=0}^{M} (C(x, n_i) + L(x, n_i)) \tag{1}$$

where $L(a, n_i)$ considers the loudness of the sound corresponding to agent a and compares it with the loudness of n_i sound, and $C(x, n_i)$ measures the quality of the intervals between the sound of a and the sound of n_i (i-neighbor). C comprises a combination of consonance, distance to the main key (which is selected according to the most common note in the space) and distance to the n_i musically speaking, all based in the Fourier transform (FFT) of each sound. Due to its complexity, C function is not fully described here, but analysed in our previous article [20] and based on the Tonal Interval Space proposed by Bernardes *et al.* [21], which allows us to create tonal music. TIS is defined as a 12-D space, where geometrical distances captures musical properties. To do so, the FFT is extracted from each note initially codified as a chroma vector. The set of the first six components of the FFT vector, considering real and imaginary part, comprises the Tonal Interval Vector (TIV), which are the coordinates of the 12-D space. That permits to encode not only notes, but also chords or keys. Bernardes *et al.* [21] and Navarro *et al.* [20] demonstrate that Euclidean distances and other geometrical measures taken in such space, captures some musical properties. In particular, the following measures were considered here:

- Consonance between two notes n_1 and n_2: In the TIS, this value is measured as the distance between the corresponding TIVs of n_1 and n_2.
- Belonging to the main key: In the TIS, we measured the degree of membership of one note to the key by calculating the angle between the projection of TIS corresponding to the key codification and the projection of TIS corresponding to the note n_i.
- Voice-leading: This part allows us to analyse voice leading between two notes considering not only the consonance, but also the number of semitones between them.

C will be a linear combination between the first measure (consonance between two notes) and the voice-leading.

The VO behavior is inspired from Particle Swarm Optimization (PSO) behavior. This algorithm proposed by Kennedy [19] is an example of swarm intelligence where each individual is moving freely through space considering three factors: the inertia weight component, the cognitive component, and the social component. To begin, (i) inertia force is related to the physical inertia and depends on the previous force applied to the particle; (ii) cognitive components refer to the attraction forces between particles or groups of particles and, finally, (iii) the social component is related to the exchange of information among particles. Within the algorithm, the particles have several premises to accomplish within system S:

- Stay near the neighbor particles. This rule prevents particles from straying too far from the center of the system.
- Move towards the gravity center. Each particle is attracted by other particles depending on certain parameters previously established. Thus, attraction forces are fundamental in this type of model.
- Avoid collisions between particles. In this case, repulsive forces are needed if the distance between two particles is too small.

The next position p_t of a particle a in the swarm depends on the current position p_{t-1}, the current velocity v_{t-1}, the best position at the current time pb_i, and the best position found by any of its neighbors pb_n, following (2) [19]:

$$\vec{p_t} = f(\vec{p_{t-1}}, \vec{v_{t-1}}, \vec{pb_i}, \vec{pb_n}) \tag{2}$$

PSO needs to be adapted to solve our specific problem. The particles in the algorithm are represented by the Synesthete agents in the VO. The three factors that provoke the particle movement are adapted to our creative system. Thus, attraction forces are related to inertia and cognitive components, while the exchange of information between agents is the social component. The cooperative attitude in a VO is also essential to achieve the goal of the whole system.

In this case, the communication allows the agent to know about its neighbors colors, sounds and position. These agents can have cooperative and non-cooperative behavior. The cooperative interactions are based on an attraction function. The intensity of the attraction forces depends on the fitness function $F(x, y)$, explained in (1). This function permits the modification of the position of each particle according to the quality measures. In contrast, non-cooperative interactions refer to a repulsion function. This repulsion function is activated only if the agents' positions are very near each other, in order to avoid collisions following the theory presented in Blackwell *et al.* [22].

The algorithm starts when each agent searches its neighbors. To do so, a ratio is established so that the agent selects who its neighbors can be. The agents carry out an interaction process to exchange information, and finally decide the best position according to the attraction force generated. The force for Agent a_i depends on the values obtained by applying the fitness function according to its neighborhood.

Equation (3) represents the calculations to get the next position p_{t+1} of a given agent a_i at iteration t. This is a linear combination of the current velocity v_{t+1} and the previous position p_t.

$$\vec{p_{t+1}} = \vec{p_t} + \vec{v_{t+1}} \tag{3}$$

Note that all of the values referring to position and velocity are vectors. The particles a_i velocity are given by (4).

$$\vec{v_{t+1}} = \sum_{k=1}^{N} F * \vec{v_t} + (\vec{p_{kt}} - \vec{p_t}) \tag{4}$$

where k represents the k neighbor agent present in particle a_i. The three different components described previously (inertia, cognitive and social) are each represented by one of the three terms in (4). The fitness function F regulates the effect of the momentum (velocity) component. The vector $(p_{kt} - p_t))$ allows the movement of particles towards the best position found by all the neighbor agents.

Within each iteration, every particle moves in a direction that is determined by the influence that its neighbors have over it. In our case, unlike the general PSO algorithm, there is no global best position for the whole system in the intermediate steps. The particles move around the search space based on these equations for a number of iterations until, if all goes well, they all converge. The convergence criterion is achieved when the positions of particles are not noticeably modified. The global best can then be taken as the final solution produced by the algorithm. At the final point of the algorithm, we also expect diverse subgroups to be generated by the attraction forces.

In order to play the full composition, each agent has the ability to decide whether to sound; in other words, to modify the property of "sounding", which is a Boolean value according to musical quality factors. To achieve this, a threshold is established so that if the values for the fitness function are not above this threshold value, they are not candidates to create the melody by the Sound Role (described later), and consequently, the "sounding" property is set to 0.

C. Sound Cloud Generation

The sound cloud is generated by the Sound Agent when the Synesthete agents have been grouped. The Sound agent decides the playing order for each group. This order depends on sound expectedness, the position of the synesthete agents, and the group they belong to.

The Sound agent extracts the positions of those agents whose attribute “sounding” is set to 1, meaning that the sound can be played. It studies the subgroups that exist by applying a clustering algorithm, and according to the mean position of each cluster, it decides the sub-swarm each agent belongs to.

The first group Sg_1 and the first note n_1 is randomly chosen. The agent then applies an expectedness measure to evaluate the probability of each sound being played in a composition following the selected value n_1. We study this probability by using the difference in loudness between the notes and rules of classical music based on the Tonal Interval Space [21], such as voice-leading and the belonging to the key above mentioned.

To select the first note of the following sub-group Sg_2 chosen randomly, we have to evaluate each note in the Sg_2 compared with the last note chosen in Sg_1. Again, the note or group of notes with the best values will be the next chord in the progression. From here, the process will be repeated until all the sounds are selected from the swarm system.

The rhythm is out of the scope for this first experiment, thus the sound will be constant along the melody.

D. Synthesizer Role

As we continue to advance in this section, the last step in our VO aims to synthesize the results proposed by the previous algorithm. The numbers for the pitches and the loudness obtained need to be interpreted so that an instrument or a synthesizer can play them. The MIDI format transforms agents properties into MIDI data [23]. This is the main task of the Synthesizer Role. Once this task is accomplished, the Role agent extracts the MIDI info and transforms it into audio information so that the computer can reproduce it.

IV. Results and Discussion

The evaluation presented in this paper aims to investigate whether sounds with low fitness values are judged more consonant than sounds with higher penalty values; in other words, if the fitness function measures the social acceptance of the music generated.

We made a preliminary experiment deploying a social network in a specific web for a number of people. There, the members can login to upload any image and listen to the results. Curiously, almost all the images used in the social network displayed for a number of people, were personal images, reflecting some events in their personal lives, such as travels, monuments or family photos. We finally selected three picture that we considered as anonymous enough, as shown in Fig. 3.

Figure 3. Collection of pictures applied to the system to extract sound cloud music.

For the purposes of this work, we considered 43 musical experts who evaluated the individual melodies in terms of tonal musical quality and their adaptation to the image they derive from. The evaluation consists of rating each fragment presented on a scale from 1 (very bad) to 5 (very good). Fig. 4 shows the results obtained in the evaluation. The fragments can be listened in the following url: goo.gl/GpLgHw. With each fragment, the image was shown in order to validate both parameters at the same time.

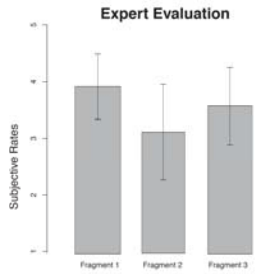

Figure 4. Evaluation results for each fragment.

In the plot, the mean punctuation of these 43 experts is shown for each image. In this case, we expected each musical fragment to be valued with a scale from 1 (very bad) to 5 (very good). Among the compositions proposed, one was well evaluated and the rest were evaluated as a fair to good sound cloud. Fragment 1 corresponds to Fig. 3a, Fragment 2 to Fig. 3b and Fragment 3 to Fig. 3c. Fragment 1 has a mean of 3.92, considered as almost Good composition according to quality and adaptation to the image. The error is about 0.51, meaning the values have been oscillating between 3.41 and 4.43, both considered above Fair rates.

Fragment 2 obtains a mean evaluation of 3.11, considered as Fair composition according to quality and adaptation to the image. However, the mean error is 0.96, meaning the values have been oscillating between 2.15, considered as bad rate and 4.43,considered as good rates. This oscilation might be due to personal preferences for the adaptation of the melody to the image, or the worse quality in musical terms comparing to the Fragment 1.

Fragment 3 gets a mean rate of 3.75 considered as almost Good composition according to quality and adaptation to the image. The error is about 0.72, meaning the values have been

TABLE I. COMPARISON BETWEEN OUR PROPOSAL, JANUS SYSTEM AND KIRKE & MIRANDA'S WORK

	Our Work	JANUS	Kirke's
Creative Products	X	-	X
Interaction	X	X	X
Open System	X	X	-
Growth Capacity	X	X	-
Social Character	X	-	-
Public Participation	X	-	-
Uses a MAS	X	X	X
BDI Architecture		X	X
Swarm Computing	X	-	-
Combines reasoning with swarm	X	-	-
VO support	X	X	-
Compatible with web services	X	X	-
Executable in different SO	X	X	X
Support to experts to interact with the system	X	X	X
Charge Balance	X	X	-
Provides a user interface	X	-	X
Provides a logging tool	-	X	-

oscillating between 3.03 and 4.47, both considered above Fair rates.

It is necessary to consider that the sound cloud is not expected to be a full tonal piece, although it follows the main tonal standards, but a soundscape or an ambient piece inspired by a painting. Additionally, all the rates obtained are subjective evaluations, which depends on the social culture, mood and personal preferences and perceptions. Therefore, the results for the same fragment can be very different between individuals. However, in view of the present results, our social machine is able to provide acceptable compositions that in some way reflect the colors of a pictorial work.

We also present a comparison among three systems to highlight the advantages of our work. In particular, Kirke's work, a creative work based on MAS [24] along with the JANUS System [25], a framework that works with VOs and MAS for general purposes. The qualitative comparison is shown in Table I. It is worth noting that the study developed by Kirke *et al.* [24] uses emotions and MAS to generate a musical melody. The use of emotions to create new music enhances the systems originality; however, while it interacts with the users, it is not a proper application for social communities. Moreover, scalability and flexibility are not included in their work, as they did not design the system following a VO methodology.

JANUS [25] is a multiagent platform that was specifically designed to deal with the implementation and deployment of multiagent systems. It is based on an organizational approach and is focused on supporting the implementation of the concepts of role and organizations as first-class entities. This consideration has a significant impact on agent implementation and allows an agent to easily and dynamically change its behavior. This feature is also shared with our system, supporting the VO design along with the musical composition. Although BDI architecture is not considered in this specific work because the agents designed do not require such architecture beforehand, it is a key feature to consider in a future work, to make a general framework that supports creative process. Apart from this, our system shares the majority of the features corresponding to a VO framework, adding a social component essential for the success of a composition system.

V. CONCLUSIONS

A social machine was developed to transform colors into music. This model was implemented with a VO to create a flexible system that can be adapted to new social contexts given by different social situations. The social component is divided into two parts: the social providers that give the pictures to the system to extract the color, and the social experts, who evaluate the quality of the music generated.

The machine component contains a workflow of four stages, all of which are based on VOs and implemented by a multiagent system. The first step consists of the color extraction of the image provided by the social community. The color properties are used to give the initial parameters that the synesthete agents use to rate the quality of the sound generated in order to move throughout the space. The movement originated in the space follows diverse rules according to a modified swarm algorithm designed for this particular work.

In order to generate a consistent music composition, an agent is developed to evaluate the probability of each sound based on the previous sounds existing in the composition. This final output is synthesized and played for an expert community. Rhythm component is primitively developed, so we will consider improving this aspect in future work.

The opinions of these experts were used to evaluate the acceptance of the music created and their accordance to the images given as an input. In particular, we considered three images with their corresponding musical fragments that were rated by 43 experts. The community of experts agrees that the quality is acceptable for this approach of our model in view of the mean results obtained, all of them above Fair good compositions. However, the number of images and fragments is not enough to extract strong conclusions. Therefore, we plan to extend this test in a future work, adding a larger number of images and analysing for the one hand the music quality and on the other hand, the adaptation of the music to the image.

However, this opinion does not affect the machine result, as the interaction is limited to the user choosing an input (picture) and then evaluating the musical result. Therefore, the machine does not learn, removing a quite important part of the social environment, such as experts' opinions. Thus, we propose futuare work to add a feedback option to the system in order to automatically incorporate expert evaluations to improve our system. This loop will permit to be plunged into an interactive evolution, in which the machine will store each experience (image, melody and overall rates) and will consider it to train a model and extract new melodies from new cases, adapted to the social evaluations of previous experiences.

ACKNOWLEDGMENTS

This work has been partially supported by the Spanish Government through FPU program FPU2013/2071.

References

[1] T. Erickson and W. A. Kellogg, "Social translucence: an approach to designing systems that support social processes," ACM transactions on computer-human interaction (TOCHI), vol. 7, no. 1, 2000, pp. 59–83.

[2] F.-Y. Wang, "Toward a paradigm shift in social computing: the acp approach," Intelligent Systems, IEEE, vol. 22, no. 5, 2007, pp. 65–67.

[3] W. R. Swartout et al., "Toward virtual humans," AI Magazine, vol. 27, no. 2, 2006, p. 96.

[4] C. Girard, J. Ecalle, and A. Magnan, "Serious games as new educational tools: how effective are they? a meta-analysis of recent studies," Journal of Computer Assisted Learning, vol. 29, no. 3, 2013, pp. 207–219.

[5] M. Si, S. C. Marsella, and D. V. Pynadath, "Thespian: Modeling socially normative behavior in a decision-theoretic framework," in Intelligent Virtual Agents. Springer, 2006, pp. 369–382.

[6] P. Zaphiris and A. A. Ozok, "Human factors in online communities and social computing," Handbook of Human Factors and Ergonomics, Fourth Edition, 2012, pp. 1237–1249.

[7] W. D. Stiehl et al., "Design of a therapeutic robotic companion for relational, affective touch," in Robot and Human Interactive Communication, 2005. ROMAN 2005. IEEE International Workshop on. IEEE, 2005, pp. 408–415.

[8] F.-Y. Wang, K. M. Carley, D. Zeng, and W. Mao, "Social computing: From social informatics to social intelligence," Intelligent Systems, IEEE, vol. 22, no. 2, 2007, pp. 79–83.

[9] Z. Huang, D. D. Zeng, and H. Chen, "Analyzing consumer-product graphs: Empirical findings and applications in recommender systems," Management science, vol. 53, no. 7, 2007, pp. 1146–1164.

[10] E. Ferrara, P. De Meo, S. Catanese, and G. Fiumara, "Detecting criminal organizations in mobile phone networks," Expert Systems with Applications, vol. 41, no. 13, 2014, pp. 5733–5750.

[11] D. López Sánchez, J. Revuelta, and F. De la Prieta, "Twitter user clustering based on their political preferences and the louvain algorithm," in Proceedings on Practical Applications of Agents and Multi-Agent Systems. In Press. Springer Verlag, 2016.

[12] J. Bajo, J. F. De Paz, G. Villarrubia, and J. M. Corchado, "Self-organizing architecture for information fusion in distributed sensor networks," International Journal of Distributed Sensor Networks, vol. 2015, 2015, pp. 2–10.

[13] A. Martin, C. T. Jin, and O. Bown, "A toolkit for designing interactive musical agents," in Proceedings of the 23rd Australian Computer-Human Interaction Conference. ACM, 2011, pp. 194–197.

[14] F. Pachet, "The continuator: Musical interaction with style," Journal of New Music Research, vol. 32, no. 3, 2003, pp. 333–341.

[15] M. Thorogood, P. Pasquier, and A. Eigenfeldt, "Audio metaphor: Audio information retrieval for soundscape composition," Proc. of the Sound and Music Computing Cong.(SMC), 2012, pp. 277–283.

[16] A. K. Hoover, P. A. Szerlip, and K. O. Stanley, http://maestrogenesis.org/, 2012, online; accessed December 10, 2016.

[17] J. Ye and S. Chen, https://www.cs.swarthmore.edu/~meeden/cs81/s14/papers/AndyLucas.pdf, 2014, online; accessed December 10, 2016.

[18] J. C. Sanz, Lenguaje Del Color: Sinestesia cromática en poesía y arte visual. H. Blume, 2009.

[19] J. Kennedy, "Particle swarm optimization," in Encyclopedia of Machine Learning. Springer, 2010, pp. 760–766.

[20] M. Navarro-Caceres, M. Caetano, G. Bernardes, and J. M. Corchado, "Iterative generation of chord progressions in the tonal interval space with an artificial immune system," Expert Systems with Applications, 2016, p. In review.

[21] G. Bernardes, D. Cocharro, M. Caetano, C. Guedes, and M. Davies, "A multi-level tonal interval space for modelling pitch relatedness and musical consonance," Journal of New Music Research, 2016.

[22] T. Blackwell, "Swarm music: improvised music with multi-swarms," Artificial Intelligence and the Simulation of Behaviour, University of Wales, 2003.

[23] W. B. Hewlett and E. Selfridge-Field, "Midi," in Beyond MIDI. MIT Press, 1997, pp. 41–70.

[24] A. Kirke and E. Miranda, "A multi-agent emotional society whose melodies represent its emergent social hierarchy and are generated by agent communications," Journal of Artificial Societies and Social Simulation, vol. 18, no. 2, 2015, p. 16.

[25] N. Gaud, S. Galland, V. Hilaire, and A. Koukam, "An organizational platform for holonic and multiagent systems," in Proceedings of Sixth international Workshop on Programming Multi-Agent Systems, 2008, pp. 104–119.