Citation for published version:
Serra, J, Orvalho, V & Cosker, D 2016, Behavioural facial animation using motion graphs and mind maps. in *MIG '16, Proceedings of the 9th International Conference on Motion in Games.*, 2994270, Association for Computing Machinery, New York, U. S. A., pp. 161-166, 9th International Conference on Motion in Games, MIG 2016, San Francisco, USA United States, 10/10/16. https://doi.org/10.1145/2994258.2994270

DOI:
10.1145/2994258.2994270

Publication date:
2016

Document Version
Early version, also known as pre-print

Link to publication

University of Bath

Alternative formats
If you require this document in an alternative format, please contact:
openaccess@bath.ac.uk

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Download date: 08. Jul. 2024

Behavioural Facial Animation using Motion Graphs and Mind Maps

José Serra[1]*   Verónica Orvalho[1]   Darren Cosker[2]

[1]Instituto de Telecomunicações & Faculdade de Ciências da Universidade do Porto
[2]University of Bath

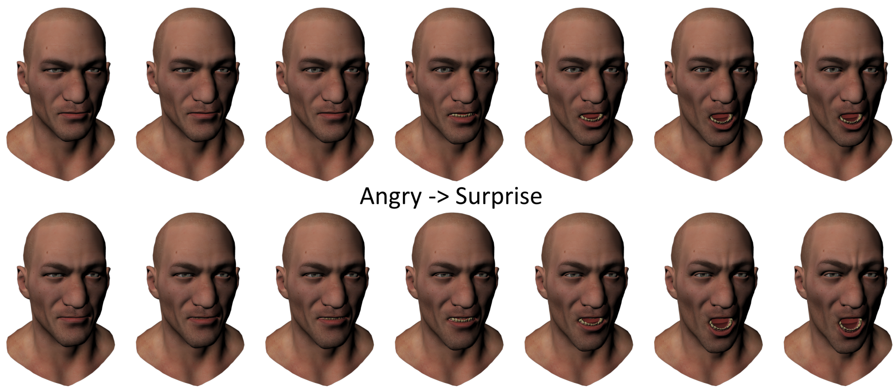

Figure 1: *Our motion graph based approach can easily generate two similar, but different, facial animations (e.g. transition between angry and surprise). All the poses and temporal dynamics were automatically chosen. This approach is combined with mind maps as behaviour controllers to create more engaging characters with less effort.*

Abstract

We present a new behavioural animation method that combines motion graphs for synthesis of animation and mind maps as behaviour controllers for the choice of motions, significantly reducing the cost of animating secondary characters. Motion graphs are created for each facial region from the analysis of a motion database, while synthesis occurs by minimizing the path distance that connects automatically chosen nodes. A Mind map is a hierarchical graph built on top of the motion graphs, where the user visually chooses how a stimulus affects the character's mood, which in turn will trigger motion synthesis. Different personality traits add more emotional complexity to the chosen reactions. Combining behaviour simulation and procedural animation leads to more emphatic and autonomous characters that react differently in each interaction, shifting the task of animating a character to one of defining its behaviour.

Keywords: Facial Animation, Procedural Generation, Behavioural Simulation

Concepts: •**Computing methodologies** → **Procedural animation;** *Motion processing;*

1 Introduction

Achieving life-like characters depends on a character being able to express facial behaviours appropriately when a stimulus occurs. However, due to time and cost constraints, animation of secondary characters in games relies heavily on a small array of animations. After several interactions, the repetitions become noticeable and may even hinder the immersion of the player. Traditionally, behaviour is controlled by scripting or decision trees, while facial animation is generated via manual or performance driven approaches. Scripting/trees require extensive behaviour discrimination and their relation with stimuli, making the definition of complex interactions very slow. On the animation side, key-framing produces very good quality animation, but is an extremely slow process and, while performance-driven techniques reduce the creation time, they require an actor and specialised equipment. Procedural animation, i.e. motion generated by an algorithm, is an alternative capable of producing unique results with a low cost, however it is seldom used in facial animation. Still, these methods do not address repeated choice of actions. Behavioural animation builds on top of procedural methods, adding a decision layer where each character becomes autonomous and chooses the animation based on the stimulus. Behaviour facial animation techniques allow controlling the dozens of characters required by videogames, enabling them to react uniquely to most varied stimuli.

We present a novel end-to-end behavioural facial animation system inspired by the motion graphs [Kovar et al. 2002] and the use of mind maps as behaviour controllers [Fernandes et al. 2012]. While motion graphs allow for generation of unique, on the fly, facial animation, mind maps serve as the character' brain, choosing when each motion should be displayed. Mind maps consist in a hierarchical graph that encodes behaviour using 2 types of nodes: emotions and actions. Whenever a stimulus occurs, the character's mood is updated following the stimulus connections, with additional influences of personality traits such as base personality. The new mood is used to trigger the synthesis of a new animation, via the motion graphs. Each facial region has its own motion graph, created from the analysis of a labelled and landmarked motion DataBase (DB). The poses from all samples are compared using a Euclidean distance based metric. Dijkstra's algorithm [Dijkstra 1959] is used to choose a path in each graph, whose analysis allows recovering the motion details. Our method is capable of storing almost the

*e-mail:jserra@dcc.fc.up.pt
Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org. © 2016 ACM.
MiG '16,, October 10-12, 2016, Burlingame, CA, USA
ISBN: 978-1-4503-4592-7/16/10...$15.00
DOI: http://dx.doi.org/10.1145/2994258.2994270

same information as the DB in significantly less space. Uniqueness is achieved via a combination of noise in the chosen path and independently calculated region paths. Mind maps also control the facial movements after the character is target of a stimulus or is idling.

Our method is particularly suited for controlling and animating a large number of secondary characters. This happens because:

- Motion graph based approach is capable of representing the training DB with considerably less storage cost and a low ratio of compression to information lost;
- Fully automatic behavioural animation system that allows characters to react differently to stimulus, without the need to both specify all animations nor all pairs of stimulus-reaction;

2 Related Work

Behavioural animation was first introduced by Reynolds [Reynolds 1987] and refers to a system where each character is responsible for animating itself according to a set of rules or guidelines [Millar et al. 1999]. Behavioural methods fit within the procedural animation field, which includes techniques to generate animation using a pre-configured algorithm. Effectively controlling the animation without directly manipulating the model's transformation/deformation. Procedural techniques can be loosely divided into: *constraint/rule*-based [Perlin 1997], where rules impose limits and variations on the generated motion; *statistical/knowledge*-based [Kovar et al. 2002; Heck and Gleicher 2007], with the motions learnt from examples; and *behaviour*-based, where the cognitive/emotional process is emulated [Xue et al. 2007]. In the last, the character's behaviour is modelled to decide the animation, while statistical and constraint-based methods focus on the actual motion. Hybrid techniques also exist [Arya and DiPaola 2007; Bidarra et al. 2010; Sagar et al. 2014].

Motion graphs fit within *statistical* procedural animation, and have been widely used for body animation [Kovar et al. 2002; Heck and Gleicher 2007; Casas et al. 2012]. Kovar et al. [Kovar et al. 2002] define motion graphs as a directed graph capable of encoding motion data in a way that synthesising movements is done by traversing the graph. The movements can be encoded in the nodes or the edges. Motion graphs are different from move trees [Mizuguchi et al. 2001] as the latter is created manually, while the former is automatic [Kovar et al. 2002]. However, applications of motion graphs to the face are scarce, with the exception of [Zhang et al. 2004]. Here, each node contains a face mesh obtained from a training sequence, with the graph created by connecting all nodes. The user specifies the source and destination nodes, with the path chosen by minimizing an L2-norm based similarity metric. Variations are added by traversing the graph several times for different parts of the face. Our approach shares some concepts with Zhang et al. [Zhang et al. 2004]. However, it extracts meaningful connections from the samples instead of connecting all nodes. This allows reducing considerably the graph size. Like Zhang, we also traverse each region graph, however, we additionally vary the chosen paths for coherent noise. Finally, the use of expression labels eases the creation of new animations.

Several behavioural animation works appeared in the context of virtual humans [Jung et al. 2011]. Arya and DiPaola [Arya and DiPaola 2007] present an approach that uses 4 independent spaces: knowledge, personality, mood and geometry. The first is controlled via a XML-based language compatible with MPEG-4, the second and third have behavioural psychology bases, while the last relies on a hierarchy to define the motions. Another hierarchical approach is presented by Perlin [Perlin 1997], where the lowest level relies on coherent noise applied in the joints, while the highest allows specifying the blended emotions. Bidarra et al. [Bidarra et al. 2010] model the character's internal state in PAD model [Mehrabian and Russell 1976] based space. On the animation side, they place short facial animations in this space, which are chosen based on proximity to the current mood. [Xue et al. 2007] use fuzzy logic to combine existing expressions using parameters from 3 layers: social, emotional and physiological. The expression is generated by blending, but timing needs to be provided. Motion graphs use the training sequences to learn the timing details. Sagar et al. [Sagar et al. 2014] combine statistical and behavioural models to create a generative model of facial expressions. They simulate, using e.g. recurrent neural network models, the different neurobiological systems, which trigger the motor circuits and drive a physically based facial model.

Most approaches rely on some sort of manually created hierarchy [Perlin 1997; Arya and DiPaola 2007; Xue et al. 2007; Bidarra et al. 2010]. Motion graphs significantly reduce the configuration step on the animation side, as they are automatically created. We additionally combine motion graphs and mind maps [Fernandes et al. 2012], previously presented from a behaviour control standpoint. Mind maps provide a visual counterpart to scripting techniques such as [Arya and DiPaola 2007; Sagar et al. 2014]. We extend the original approach by connecting it to motion graphs and introducing chained actions for more complex reactions. On the other hand, Sagar et al. [Sagar et al. 2014] present, arguably, the most advanced approach, however this requires a complex set-up. Mind maps or scripting require considerable less knowledge and expertise, providing just enough emotional and cognitive complexity. Also, as it simulates the brain, unexpected behaviours might arise, which is undesirable in a game. For more details on character behaviour simulation, we forward the reader to [Vinayagamoorthy et al. 2006; Jung et al. 2011].

3 Method Pipeline

Our approach for behavioural animation is divided in two stages: 1) set-up, where the motion graphs are generated from the analysis of a DB (A) and the behaviour is defined via the mind maps interface (B), and 2) behaviour choice and motion synthesis, where a stimulus is analysed in the mind map and triggers synthesis via motion graphs (fig. 2). In the set-up stage, the motion graphs of each facial region are created (sec. 4.1) by comparing all poses in the DB and merging them when similar enough (sec. 4.2), following user defined thresholds. Also in the set-up, the author defines the behaviour (sec. 6) by creating the mind map, with its emotional and action layers, and personality traits, such as initial mood and personality. Finally, in-game events will cause new motions to be generated. First, the stimulus triggers an update in the mood that, in turn, triggers a new action to be chosen (sec. 6.1). These actions have associated expression labels, which are used to find paths in the region graphs (sec. 5). Each path node is then used to reconstruct the poses, thus generating the final landmarks' movements.

4 Motion Graph Generation

Creating the region motion graphs starts with a dynamic 2D/3D sparse DB, whose samples have been aligned to remove influences of identity and head movements. Motion graphs are created by comparing all poses/frames of each sample with all others. Merging occurs if the similarity (sec. 4.2) value between two frames/nodes is smaller than a user-defined threshold. This controls the trade-off between motion quality (smoothness) and flexibility of the graph (compactness). The DB labels provide the control mechanism that drives motion synthesis. The proposed method also relies in region graphs, instead of the holistic approach, to allow more ac-

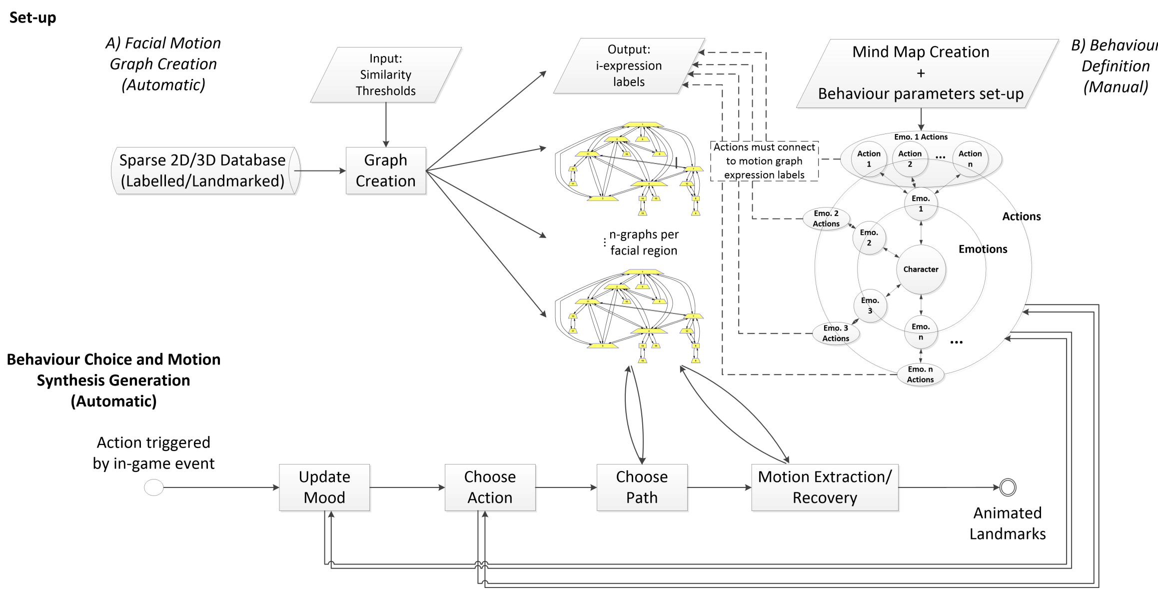

Figure 2: *Overview of the behaviour animation pipeline. First, the DB is analysed to create a graph per facial region and the character's mind map is defined. Afterward, game stimulus force the character's mood update, which is used to choose an animation. The motion graphs are traversed to recover the final motion.)*

curate similarity values, preventing inter-regional influences. The regions chosen for this paper were: eyes, eyebrows, nose, mouth, with the cheeks and jaw grouped into another. These were chosen empirically, however, other configurations are possible.

We have selected the Cohn-Kanade (CK and CK+) data set [Kanade et al. 2000; Lucey et al. 2010] because, while not originally intended for motion synthesis, it is has sparse landmarks (lightweight), has expression labels and multiple samples per label. [Zhang et al. 2014] presents an alternative, however it is not free. CK/CK+ expression labels are associated to the emotions of: *happiness, sadness, disgust, surprise, anger, fear, contempt* plus *neutral*. All samples follow the structure: neutral-to-peak expression, with each pose containing 68 landmarks. We clean the DB both manually and by fitting a sigmoid curve to each landmark displacement, per sample. The curve's parameters are optimized via least squares. We additionally reduce the number of landmarks (fig. 3) to lower the overhead (fig. 3) and the errors in similarity calculation. Procrustes analysis is applied for pose alignment, first by removing effects of translations from each sample's neutral pose and then regionally between samples to reduce effects of proportions.

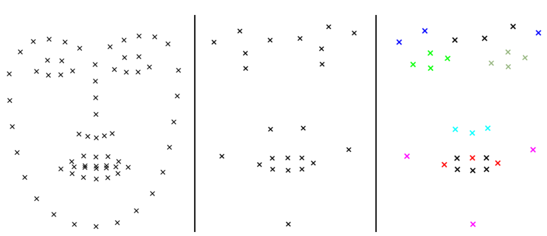

Figure 3: *From left to right: original DB landmarks (68); reduced set used in graph creation; "Non-rigid" alignment markers of the face regions, grouped by colours.*

The core structure of our approach is a directed graph, where each node contains the landmarks displacement and each edge has a similarity value. The displacement was chosen, as opposed to the actual positions, to mitigate errors in alignment and effects of each person's proportions. The graph source contains the average of the neutral poses which combined with each node's displacement obtains the final poses. Each graph node also contains all expression labels from the samples whose nodes were merged in it. Finally, we store information in the nodes for better motion recovery that includes the average number of consecutive merged nodes and their respective landmarks' averaged velocities. Consecutive refers to, when merging a sample graph into the final graph, several sample nodes directly connected by one edge, might be merged into the same graph node. Destination nodes, i.e. nodes that contain a peak expression from the original samples, store the average number of frames from neutral to peak and respective standard deviation;

4.1 Graph Creation

Graph creation is the same irrespective of the region. First, each sample is converted to a motion graph, i.e. sample graph and then the sample graphs are merged into the final graphs. This is done by identifying similar nodes that serve as transition points. Both steps are similar with the main difference being that one works with frames from the samples, while the other already compares nodes. Each step has its own threshold.

At this point, it should be noted that all sequences used to create the motion graph need to start with the same/equivalent pose. This initial pose is the common denominator, allowing the displacement to be obtained. A sample motion graph is then generated following these steps for each pose:

- Calculate the displacement between current pose and the sample's first pose;
- Add this value to the average of all samples's neutral poses;
- This new pose is used to calculate the similarity to all the current sample-graph nodes;
- If the similarity value is lower than the user-defined threshold, we merge the new pose with the sample graph node with the lowest similarity; Otherwise, we create a new node and append it to the sample graph. In both cases a connection is established to the previous iteration node.

Creating the final graph is an iterative process seeded with a random sample graph. Merging a sample graph with the final graph again follows the same comparing-and-merge approach.

4.2 Similarity Metric

The chosen similarity metric, eq. 1, takes into account both spatial location of the landmarks and their instantaneous velocity. When comparing two different poses a and b, we define that each pose/frame P_a is composed by n number of landmarks $< PaL_1 ... PaL_n >$. As such the distance between two poses for the same landmark i is given by $dist(PaL_i, PbL_i) = \sqrt{\sum_{j=1}^{dimensions}(PaL_{i,j} - PbL_{i,j})^2}$. The instantaneous velocity of a landmark i for the P_a is given by $\vec{V}(PaL_i) = PaL_i - Pa_{prev}L_i$, where $Pa_{prev}L_i$ is the position of L_i in the pose/frame immediately before P_a. On top of this, we calculate the velocity's influence, v_{infl}, which is a scalar that represents how similar are the velocities of a landmark i in two poses. This is given by $v_{infl}(PaL_i, PbL_i) = 1 - |\vec{V}(PaL_i) \cdot \vec{V}(PbL_i)|$. v_{infl} varies between $[0, .., 1]$: 0, if the two vectors are the same, independently of the direction; 1, if they form $+/-90°$. If both vectors are close to opposite, we argue they represent onset or offset phases, thus the influence should be the same. This is achieved using the absolute value. The final similarity value is obtained via:

$$sim(P_a, P_b) = \sum_{i=1}^{n} \left(dist(PaL_i, PbL_i) + \lambda v_{infl}(PaL_i, PbL_i) dist(PaL_i, PbL_i) \right) \quad (1)$$

λ controls the influence of the velocity (smoothness) in the result.

5 Motion Synthesis

New motion is created by traversing the graph, choosing the nodes relevant to the desired facial behaviour. We associate nodes that contain peak, or apex, expression to the expression labels. We assume these are the most desirable to animate a character. In a long animation, the current sink node becomes the source for the next iteration, with only the first source manually specified. A path is chosen, for each region graph, using Dijkstra's algorithm [Dijkstra 1959] that minimizes the path similarity values. However, the paths cannot be used directly as the temporal dynamics were partially lost.We use the information contained in each node to recover as much information as possible. This is done by determining the core poses and temporal dynamics (step 1), extending these to achieve a more realistic motion (steps 2 and 3) and smoothing the final sequence. These are now described in more detail:

1. Use average of consecutive merged nodes to control the number of poses generated by the node. This is crucial to adequately recover non-linear aspects of motion. If only one pose is generated. the node's displacement is used directly to create the pose. With more poses, the velocity is used to distribute the poses evenly around the central point and in the direction of the next/previous nodes.

2. Use the average peak nodes duration as the length of neutral-to-peak sequences. The current displacement is stretched or shrunk using linear interpolation. This reduces inaccuracies related to the nodes having information from multiple samples. For peak-to-peak transitions, we estimate the length by finding the neutral-to-source and neutral-to-target lengths and using the path's node closest to the neutral pose as a weighting factor.

3. Normalise the displacements of each facial region via linear interpolation with the longest path as reference.

4. Smooth the sequences using the Savitzky-Golay window-based filter [Orfanidis 1996] and sigmoid fitting (as in sec. 4). These smoothing approaches complement each other, as the first removes drastic motions and the second removes any left zigzag. This tends to occur when generating sequences not in the DB or when path noise is introduced (sec. 5.1).

We additionally use the graph to generate idle movements, which are crucial to create realistic characters [Egges and Magnenat-Thalmann 2005]. We find the neighbours of the nodes associated to the current expression, and randomly hop between these. We limit the hops to neighbours that share the expression label of the current expression. These movements are learned purely from the similarity of expressions. As a result, the author needs to manually provide the hops occurrence interval and the time remaining in each hop.

5.1 Introducing Variation

Synthesizing animations that share a common label, but are slightly different, is a crucial requirement for our method. Our approach inherently introduces noise, as each facial region produces an independent motion from the respective graph. We force additional variations by using the standard deviation stored per peak node to define a normal distribution and randomly sampling this space to vary the sequence duration. The path is also varied by changing the Dijkstra's results (sec. 5). Instead of using the original nodes, we randomly replace them with their neighbours.

6 Mind maps as behaviour interface

Mind maps have been previously presented as character behaviour controllers [Fernandes et al. 2012], however we extend this technique by connecting it with motion graphs. Specifying a character's behaviour has two steps: creating the mind map and defining the mood and personality properties. A mind map is a hierarchical graph where the central node represents the character and connects to an emotional nodes layer, which in turn, connects to an action/stimulus nodes layer. Emotional nodes can be any kind of emotion, psychological states, e.g. bored, or other mind states. These determine the representation of mood and personality traits, which is defined by a weighted array with one entry per emotional node. An action node can be received, akin to a stimulus, and performed. Performing actions connect with emotions or other performing actions, while stimuli only connect with emotions. Chain reactions, e.g. a character receiving a gift that first is surprised and then happy, are obtained by connecting performing actions. Each emotional node should also have a flagged performing action, which triggered as the mood decays (sec. 6.1). Also, each edge connecting actions to emotions has a weight, which can be seen as the stimulus influence in the mood or the emotional threshold to trigger an action. Multiple edges per stimulus allow more complex behaviour that extends "black or white" actions of the original approach.

The mood represents the current emotional state of the character. It is updated after a stimulus occurs, which then triggers a performing action (sec. 6.1). Personality traits influence the mood variation and include: personality, an array to which the mood decays; background context, representing the influence of the environment; and affection matrix, holding the moods towards other characters. While the starting mood and the personality are mandatory, the background context and affection matrix are not. After an update, the mood starts decaying to the personality, whose duration needs to be specified. The dynamics of the variation are controlled by a curve that can be e.g. linear, quadratic or sigmoid.

6.1 Triggering Animation

Synthesis of animation occurs whenever a stimulus happens and in certain intervals when the mood is decaying. While the former is akin to the perception-decision-action loop, the latter provides insights on the character's mood. Whenever a stimulus occurs: 1) it is found in the mind map and the emotion weights obtained, 2) these weights are used in conjunction with the personality traits to update the current mood (eq. 4), 3) the highest value of the mood then specifies the emotional node from where the performing action is selected, based on edge weight closest to the mood value (fuzziness is added for similar edge values), 4) finally the expression label of the performing action controls the synthesis on motion graphs. For actions triggered by the mood decay, only steps 3 and 4 are relevant. However in step 3, we choose the representative node of emotional node associated to the highest mood value. Synthesis is also slightly different, as the behaviour should only hint at the current mood. Instead of generating the motion from all path nodes (sec. 5), we use only the initial nodes. When the chosen emotional node does not have a representative node, there is no synthesis. Idle movements are added whenever the character is not receiving any stimulus or in the intervals of mood decay animations. Additionally, as the characters interact, it is possible that a character faces a stimulus not present in its mind map. When this happens the character will copy the stimulus and respective emotional weights from the mind map of the stimulus source. For world stimuli, e.g. an explosion, the particular stimulus should contain the emotional node and edge weights to be copied.

Updating the mood consists in a simple sum of all properties. The equation 4 was split purely for better readability. The *mood* represents the current mood, *ActionEmot* is the array formed by the stimulus edge weights, with λ representing how the affection towards another character, $Affect$ matrix, can change the perception of the stimulus. *Personality* represents the influence of the base personality in the received action, *BGContext* deals with the influence of the current location, with α representing the importance of current stimulus over the personality traits. Finally β represents how much the received stimulus can change the current mood.

$$Stimulus = (1-\lambda)ActionEmot + \lambda Affect \tag{2}$$

$$PTraits = \frac{Personality + BGContext}{2} \tag{3}$$

$$mood = |(1-\beta)mood - \beta(\alpha Stimulus + (1-\alpha)PTraits)| \tag{4}$$

7 Results & Discussion

Both motion graphs and mind maps were implemented in *Matlab*© and tested in a laptop with an i7-4720HQ and a NVidia GTX 970M. The landmarks sequences are imported in *Autodesk Maya 2011*© and applied to a 3D blendshaped face model using direct manipulation [Lewis and Anjyo 2010] that converts 2D landmarks movements into 3D blendshape animation. Nevertheless, the 3D animation results of the accompanying video should serve only as a rough approximation of how an animation could look like. Accompanying video shows results of using motion graphs *per se* and combined with mind maps, from where fig. 4 was extracted. The model used in all animations is the same, as this allows seeing purely the differences caused by our method. We now discuss early results on both motion graphs and mind maps.

The motion graphs used were created from the analysis of 70 sequences from ~20 subjects of the CK/CK+ [Kanade et al. 2000; Lucey et al. 2010] DB. We have compared the training samples against the synthesized equivalent and found the proposed technique is capable to synthesizing similar motion. The compression ratio refers to the memory footprint gains compared to a graph with a node per pose. As a reference, for a ratio of 90%, the difference in length is 0.98 ± 1.84 frames, and the difference in pose pixels is 2.86 ± 2.3. For a ratio of 78%, the respective differences are 0.071 ± 0.39 and 2.12 ± 1.81 and for 57.53%, we have the differences of 0 ± 0 frames and 1.52 ± 1.51 pixels. As for the generation times, we have results in the order of ± 1 seconds. It is also

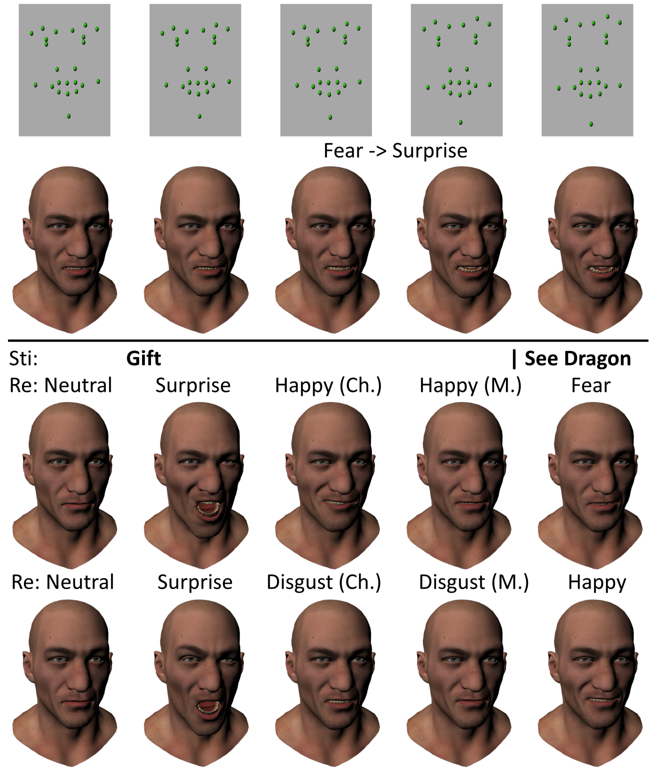

Figure 4: *From top to bottom: landmark sequence generated from Fear to Surprise mapped to a 3D facial mode. 3rd and 4th sequences show only the key poses generated from the same motion graph, but different mind maps for the same input stimuli,* ***Sti****.*

important to note that high compression ratios can lead to peak expressions of different labels being merged in the same peak node. Our method method produces results comparable to the face graphs of Zhang et al. [Zhang et al. 2004]. Both approaches also have issues, e.g. face graphs produce motions that occasionally have distinguishable phases and pass through a clear not relevant mid expression Our approach sometimes produces distinguishable region timings. Also in both methods, occasionally there is an exaggerated pose jump. Our method's main advantage is compression and shorter path finding, as their method takes ~1.9 seconds. The chosen DB additionally impacts the results as it still contains some jitter even after cleaning, and only has motions from neutral-to-peak. This issue is "*bypassed*" by appending to the end of a sample, its own reversed copy. Nevertheless, facial behaviours have different onsets and offsets [Ambadar et al. 2009]. As for the main limitations, the presented method relies heavily on well defined peak expressions, which is not always the case. Using sigmoid fitting to smooth the results can also lead to excessive removal of fine details.

Mind maps work primarily as a reactive framework that allows the author to control the behaviours without fully describing all possible situations. On the other hand, the character will lack behaviours when no stimulus occurs. They are also not suited to represent the complex cognitive and emotional processes of the brain (to which we forward to [Sagar et al. 2014]), nor controlling the decisions to achieve a certain goal, e.g. speech based interactions. While it would be possible, the required chain of actions and edge weights would be so fine grained that would make the mind map conception too complex. For more intricate interactions, scripting languages [Arya and DiPaola 2007] or decision tree are more suitable. Still, these approaches and mind maps can complement each other, with the latter adding emotional complexity. Thus, simplifying character definition by removing the need to specify all possible mood variations, while also visually defining the reactive part of the character.

8 Conclusion

In this paper, we propose a behavioural animation pipeline to control and synthesise character reactions to external stimulus and internal mood changes. Mind maps control the behaviours choices and motion graphs the synthesizes of animations on the fly. The bridge between mind maps and motion graphs is established via the expression labels, which are used to find destination nodes in the graph. The number of sequences our method can generate is almost limitless, due to independent region graphs and by introducing small variations in the minimum path. Finally, the presented approach is capable of encoding the motions of the training DB in a much smaller memory footprint.

Our priorities for the **future** include solving the issue of of disconnected movements between the regions, which is required to achieve more credible motions. We would also like to study how to control more complex interactions such as speech using mind maps. The presented method is particularly useful for animating non-playable characters (NPC) that interact with the player or between themselves, in crowds of both videogames and films. The same motion graph can be used in several characters, which coupled with different mind maps and personality traits, allows to significantly reduce the character creation cost, while making it more believable. Motion graphs and mind maps have a wide range of applications, and while their use is not a new idea, their application in facial animation remains under-exploited.

9 Acknowledgements

This work is funded by Fundação para a Ciência e Tecnologia/FCT (SFRH/BD/79905/2011), POPH/FSE program, and EPSRC grant CAMERA EP/M023281/1. We thank *Faceware*© for allowing the use of the Victor model. We also thank Ozan Cetinaslan and Pedro Mendes for all the feedback and Ozan for the direct manipulation code. Finally, we would like to thank the reviewers for the priceless feedback.

References

AMBADAR, Z., COHN, J., AND REED, L. 2009. All smiles are not created equal: Morphology and timing of smiles perceived as amused, polite, and embarrassed/nervous. *Journal of Nonverbal Behavior 33*, 1, 17–34.

ARYA, A., AND DIPAOLA, S. 2007. Multispace behavioral model for face-based affective social agents. *J. Image Video Process. 2007*, 1 (Jan.), 4–4.

BIDARRA, R., SCHAAP, R., AND GOOSSENS, K. 2010. Growing on the inside: Soulful characters for video games. In *IEEE Symposium on Computational Intelligence and Games*, 337–344.

CASAS, D., TEJERA, M., GUILLEMAUT, J.-Y., AND HILTON, A. 2012. 4d parametric motion graphs for interactive animation. In *Symposium on Interactive 3D Graphics and Games*, ACM, New York, NY, USA, 103–110.

DIJKSTRA, E. 1959. A note on two problems in connexion with graphs. *Numerische Mathematik 1*, 1, 269–271.

EGGES, A., AND MAGNENAT-THALMANN, N. 2005. Emotional communicative body animation for multiple characters. In *International Workshop on Crowd Simulation (V-Crowds)*, 31–40.

FERNANDES, T., SERRA, J., ÓRDO NEZ, J., AND ORVALHO, V. 2012. Mind maps as behavior controllers for virtual characters. In *CHI - Extended Abstracts*, ACM, 2291–2296.

HECK, R., AND GLEICHER, M. 2007. Parametric motion graphs. In *Symposium on Interactive 3D Graphics and Games*, ACM, New York, NY, USA, 129–136.

JUNG, Y., KUIJPER, A., FELLNER, D. W., KIPP, M., MIKSATKO, J., GRATCH, J., AND THALMANN, D. 2011. Believable Virtual Characters in Human-Computer Dialogs. In *Eurographics - State of the Art Reports*, N. John and B. Wyvill, Eds.

KANADE, T., COHN, J., AND TIAN, Y. 2000. Comprehensive database for facial expression analysis. In *proc. of 4th IEEE Int. Conf. on Automatic Face and Gesture Recognition*, 46–53.

KOVAR, L., GLEICHER, M., AND PIGHIN, F. 2002. Motion graphs. In *In proc. of SIGGRAPH*, 473–482.

LEWIS, J. P., AND ANJYO, K. 2010. Direct manipulation blendshapes. *IEEE Comput. Graph. Appl. 30*, 4 (July), 42–50.

LUCEY, P., COHN, J., KANADE, T., SARAGIH, J., AMBADAR, Z., AND MATTHEWS, I. 2010. The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 94–101.

MEHRABIAN, A., AND RUSSELL, J. 1976. The three dimensions of emotional reaction. *Psychology Today 10*, 3, 57–61.

MILLAR, R., HANNA, J., AND KEALY, S. 1999. A review of behavioural animation. *Computers & Graphics 23*, 1, 127 – 143.

MIZUGUCHI, M., BUCHANAN, J., AND CALVERT, T. 2001. Data driven motion transitions for interactive games. In *Eurographics 2001 - Short Presentations*.

ORFANIDIS, S. 1996. *Introduction to Signal Processing*. Englewood Cliffs, NJ, Prentice Hall.

PERLIN, K. 1997. Layered compositing of facial expression. In *In proc. of SIGGRAPH*, ACM Press, 226–227.

REYNOLDS, C. W. 1987. Flocks, herds and schools: A distributed behavioral model. In *In proc. of SIGGRAPH*, ACM, 25–34.

SAGAR, M., BULLIVANT, D., ROBERTSON, P., EFIMOV, O., JAWED, K., KALAROT, R., AND WU, T. 2014. A neurobehavioural framework for autonomous animation of virtual human faces. In *In proc. of SIGGRAPH Asia*, SA '14, 2:1–2:10.

VINAYAGAMOORTHY, V., GILLIES, M., STEED, A., TANGUY, E., PAN, X., LOSCOS, C., AND SLATER, M. 2006. Building Expression into Virtual Characters. In *in Eurographics - State of the Art Reports*, ACM, 21–61.

XUE, Y.-L., MAO, X., LI, Z., AND DIAO, W.-H. 2007. Modeling of layered fuzzy facial expression generation. *In Digital Human Modeling: International Conference on Digital Human Modeling*, 243–252.

ZHANG, L., SNAVELY, N., CURLESS, B., AND SEITZ, S. M. 2004. Spacetime faces: High resolution capture for modeling and animation. *ACM Trans. Graph. 23*, 3, 548–558.

ZHANG, X., YIN, L., COHN, J. F., CANAVAN, S., REALE, M., HOROWITZ, A., LIU, P., AND GIRARD, J. M. 2014. Bp4d-spontaneous: a high-resolution spontaneous 3d dynamic facial expression database. *Image and Vision Computing 32*, 10, 692 – 706. Best of Automatic Face and Gesture Recognition 2013.