The University of Bradford Institutional Repository

Author(s): Sheng, Y., Willis, G., Ugail, H. and Gonzalez Castro, G.

Conference paper: PDE-based Facial Animation: Making the Complex Simple

Publication year: 2008

Conference: ISVC 2008

eISSN: 1611-3349

Publisher: Springer Verlag

Publisher's site: *http://www.springerlink.com/content/105633/*

PDE-Based Facial Animation: Making the Complex Simple

Yun Sheng[1], Phil Willis[1], Gabriela Gonzalez Castro[2], and Hassan Ugail[2]

[1] Department of Computer Science, University of Bath, United Kingdom
[2] School of Informatics, University of Bradford, United Kingdom

Abstract. Direct parameterisation is among the most widely used facial animation techniques but requires complicated ways to animate face models which have complex topology. This paper develops a simple solution by introducing a PDE-based facial animation scheme. Using a PDE face model means we only needs to animate a group of boundary curves without using any other conventional surface interpolation algorithms. We describe the basis of the method and show results from a practical implementation.

1 Introduction

Commercial applications have boosted research in facial animation. In the games industry, many demanding applications require simulating realistic human interactions, and therefore also require realistic facial animation technology, often for talking characters. In the film industry, 3D computer-generated impersonators can be employed in dangerous scenes where it would be inappropriate to use humans. Facial animation technology can be used to increase the degree of realism of the 3D impersonators. In medicine, in order to eliminate unnecessary surgical accidents, facial animation technology can be used in both craniofacial surgical planning and facial tissue surgical simulation ahead of the surgery. Facial animation technology can also be exploited to reduce transmission bit-rates during videoconferencing, by sending only animation parameters rather than the video sequence.

Since Parke's pioneering work [1], research has continued into improved visual realism, improved expressiveness and into the ability to control the face effectively. As concluded in [2], mainstream facial animation techniques fall the following five fundamental categories:

1. *Key-frame animation* is the earliest attempt to animate a 3D face. The idea is that, given at least two key frames with variant facial expressions, intermediate expressions between the key frames can be produced by interpolation [1].
2. *Performance-driven animation* is carried out by capturing motion data of 3D mimics with some special interactive input devices, such as the motion capture cameras [3], data gloves, monkey suits [4], etc.
3. *Muscle-based animation* simulates muscle movements anatomically with a 3D face wireframe. Notable examples include the FACES (Facial Animation, Construction and Editing System) [5], Terzopoulos et al.'s physically-based model [6] and Zhang's multilayer physical model [7].

G. Bebis et al. (Eds.): ISVC 2008, Part II, LNCS 5359, pp. 723–732, 2008.

4. *Pseudomuscle-based animation* simulates muscle actions using geometric deformation operators, such as the boundary element approach [8], FFD (Free Form Deformation), etc.
5. *Direct parameterisation* utilises a set of animation parameters, such AUs (Action Units) or MPEG-4 FAPs (Facial Animation Parameters) parametrically to represent facial animations.

Direct parameterisation for facial animation is the most commonly used animation approach because this class of methods provides users with a parametric description of the face geometry, with a high degree of adaptability to the application. These direct parameterisation methods differ from each other, usually depending on the complexity of the face model geometry. For those topologically-simple face models, for example the CANDIDE-3 face model with just above one hundred vertices [9], facial animation can be parameterised relatively intuitively [10]. This is because it only requires associating a small group of vertices with each animation parameter; and determining how these vertices move with respect to the change of parameter values. For complex face models, especially ones acquired with a 3D laser scanner which yields a very large numbers of vertices, direct parameterisation is not feasible.

One of the commonly used solutions is to identify a set of control points in the point cloud, chosen to encompass the most salient facial features on the face. Facial animation is started by parameterising this relatively small number of control points. These control points are used to move the surrounding vertices by other surface interpolation algorithms, such as the radial basis function and Dirichlet free-form deformations [11]. This raises the problem of how to make the motion of the control points influence the surrounding vertices smoothly and effectively. There is as yet no recognised best solution to this.

To resolve this, we propose partial differential equations (PDE) to parameterise facial animation. Having been studied for decades, it is known that the PDE method is an effective and powerful tool in CAD (Computer-Aided Design) [12], capable of parameterising complex geometry with a relatively small group of parameters. This method adopts a boundary value approach, whereby a surface is designed by defining a number of space curves. Typical applications of the PDE method are in CAD, where there is little need to animate a surface in a controlled fashion. This animation aspect of the PDE method is one we explore here.

The PDE method has successfully been used to model the human face with many merits [13], when the 3D face model constructed by the PDE method can be referred to as the PDE face. The approach provides users with a great degree of freedom to individualise the 3D face by adjusting a small set of facial boundary curves acquired beforehand, by scanning a real person. This boundary curve characteristic can benefit the direct parameterization in facial animation. Animation of a complex face model can be implemented by adjusting the position of the boundary curves before each reconstruction of the PDE face. In other words, these facial boundary curves are parameterised for animating the PDE face that will be rendered later. Depending on the demand of the application, the reconstructed PDE face could be either simple or highly complex. The PDE method is inherently capable of generating smooth facial animations with a complex face model, by adjusting only a relatively small number of boundary curves. Hence, we generalise the major advantages of the PDE-based facial animation method over conventional ones as follows:

- The PDE-based facial animation method provides animators with a user-friendly and effective interface, requiring the adjustment of a relatively small number of parameters, i.e. the facial boundary curves.
- The PDE-based method is capable of generating facial animations with a complex face model by adjusting a group of boundary curves only, eliminating the need for the animator to use control points to interpolate other vertices of the 3D face model.
- The PDE-based method allows one animation scheme (for boundary curves) to be used for different face models in variant resolution.
- Unlike conventional animation methods, the PDE-based method inherently results in smoothly animated facial surfaces without additional trimming.
- This technique allows both modelling and animation to be performed on the *same* underlying model, and thus there is no added layer used to animate the model.

The rest of the paper is structured as follows: Section 2 briefly reviews the PDE face developed in [13]. Section 3 is dedicated to the justification of why the PDE-based facial animation scheme is better for complex face model animation. In Section 4, we introduce the implementation of our direct parameterisation scheme with a selected group of FAPs. In Section 5, the results of the PDE-based facial animation are demonstrated by simulating four universal facial expressions with our newly-developed interface, called the PDE face generator. Finally, the conclusion is drawn in section 6.

2 PDE Face

The PDE face inspired by the Bloor-Wilson PDE (BWPDE) method [14] differs from conventional face models [13]. The BWPDE adopts the fourth order PDE with four boundary curves used to derive a solution for the PDE in surface design. The PDE face is constructed by resolving a series of BWPDEs, i.e. a combination of fourth order PDEs. The PDE face provides users with a great degree of freedom to mimic a 3D face with a relatively small number of parameters, i.e. the facial boundary curves.

The PDE face employs 28 boundary curves acquired from a laser-scanned 3D face model, which was captured by scanning a human head with the eyes open and the mouth closed so that its neutral configuration is portrayed naturally. The 28 boundary curves cover the entire facial area from the chin to the forehead, describing the most salient facial features. In the PDE face, the parametric 3D facial surface $S(u, v)$ is defined as solutions to a series of fourth order elliptic PDEs, each of which has a consistent form:

$$\left(\frac{\partial^2}{\partial u^2}+\frac{\partial^2}{\partial v^2}\right)^2 S(u,v)=0 \tag{1}$$

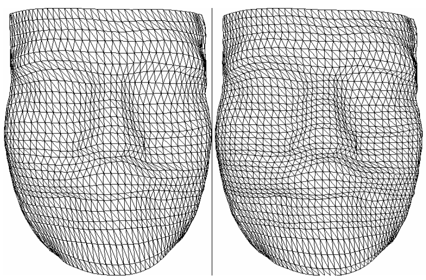
where u and v are the parametric surface independent variables, limited to $0 \le u \le 1$ and $0 \le v \le 2\pi$. The above PDE can be resolved by a group of analytic functions. More details about how to resolve the above PDE can be found in [13]. Given that a fourth order PDE is in use, four boundary conditions are necessary to find the particular solution to

Equation (1) and therefore, the solution to nine different PDEs with 28 boundary curves is required to produce the whole face model. Continuity in the model is guaranteed by prescribing at least one common boundary condition for adjacent patches.

Fig. 1 shows different views of the boundary curve points for a PDE face, where each of 28 horizontal curves is defined by 46 of these points. To ensure that the generated face passes through all the facial boundary curves, the curves are substituted into the PDEs as boundary conditions. This give the generated face a high fidelity and some of the facial curves are also associated with essential facial features.

(a) (b) (c)

Fig. 1. Discrete representation of 28 boundary curves of PDE face. (a) Front view (b) Profile (c) 45-degree rotational view.



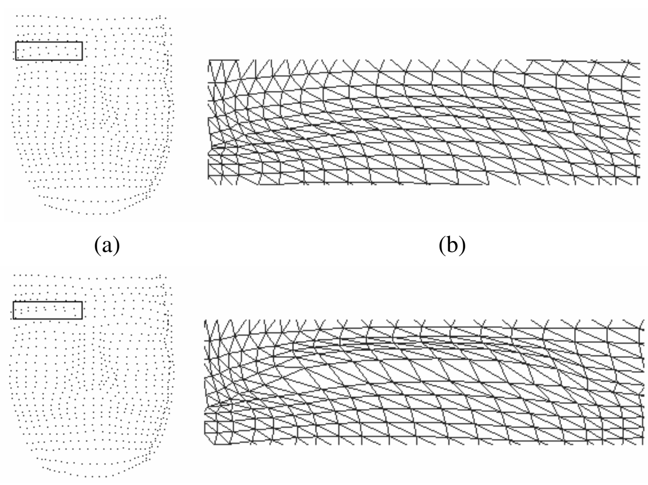
(a) (b)

Fig. 2. PDE face. (a) Using a 28 by 41 *uv* mesh (b) Using a 37 by 41 *uv* mesh.

The resolution of the *uv* mesh determines that of the resulting PDE face wireframe. The more *uv* samples, the smoother the generated face surface, but the more computationally complex. This is a useful trade-off which can be exploited according to the display and specific application. A complex PDE face model can be built by simply increasing the resolution of its *uv* mesh. Fig. 2 illustrates neutral views of a PDE face produced with the same facial boundary curves, but with two different *uv* meshes. Thus, by modifying the resolution of the *uv* mesh, it is possible to use the PDE with a relatively small number of boundary curve points to generate the facial animation with a 3D face in high resolution. This will be demonstrated in the next section.

3 PDE-Base Facial Animation

As stated previously, conventional direct parameterisation facial animation schemes for complex face models generally employ control points to help interpolate the large number of vertices in the model. This leads to demanding interpolation algorithms. Our technique uses boundary curves to animate the 3D face, instead of manipulating directly a set of control points selected from the vertices of the 3D model. The boundary curves used are not vertices in the resulting PDE model and thus, interpolation schemes to recalculate the modified position of vertices is not required. In other words, the vertices of the animated PDE face are positioned by the PDE algorithm using the modified boundary curves as input, rather than by the surface interpolation schemes used in the conventional approaches. As described in the preceding section, by modifying the resolution of the *uv* mesh it is feasible to use a relatively small number of boundary curve points to animate a complex 3D face in high resolution. Moreover, the inherent attributes of the PDE method guarantee a smooth reallocation of the vertices of the animated face. Thus, unlike conventional methods in direct parameterisation for complex face models, the PDE-based facial animation will result in a smooth displacement of the vertices without any extra surface interpolation algorithms.



(a) (b)

(c) (d)

Fig. 3. Illustration of the PDE-based facial animation. (a) Discrete facial boundary curves before animation. (b) Active area of the PDE face before animation. (c) Discrete facial boundary curves after animation. (d) Active area of the PDE face after animation.

Fig. 3 shows a subjective example of the PDE-based facial animation. Fig. 3(a) shows the frontal half of the facial boundary curve points before an animation takes place, with a rectangle highlighting the active area on the boundary curves. Fig. 3(b) shows the enlarged region of the constructed PDE face before animation, corresponding to the active area underlined in Fig. 3(a). Fig. 3(c) shows that, in the rectangle, the lower curve moves towards the upper one in animation with other curves remaining fixed. The impact on the animation can be seen in Fig. 3(d), which gives rise to a stretch of a group of vertices on the generated PDE face. It is worth noting that the magnitude of the vertex stretch is gradually weakening from the bottom up, producing a smooth change in the face surface. Moreover, for the sake of visibility, the resolution of the *uv* mesh used here is set to 55 by 41. Thus, the PDE face we see here consists of 2255 vertices. However, depending on the need, the complexity of the PDE face can be arbitrarily increased by enlarging the resolution of the *uv* mesh.

4 FAPs-Driven Animation

We adopt MPEG-4 Facial Animation Parameters (FAPs) for our direct parameterisation scheme. In MPEG-4, the motion of the most low-level FAPs is standardised only by associating each of them with a unique facial feature point, the definition of which is also standardised by the MPEG-4 Facial Definition Parameters (FDPs). The standard leaves users a great degree of freedom to define their proprietary face models as long as the models comply with the MPEG-4 requirement for some vertices on the model to correspond to MPEG-4-defined feature points. We choose a group of points from the PDE facial boundary curves, such as *right corner of left eyebrow*, *left corner of inner lip contour*, etc, to ensure compliance with the MPEG-4 definition. However, these selected points are not the only ones affected by FAPs. In order to achieve a smooth animation, groups of boundary curve points anatomically-related to those already selected are also chosen.

Since the animation of the eyebrows, eyes and month contributes most to facial expressions, there are the three facial features studied in this paper. Furthermore, the FAPs considered here are only the low-level ones.

4.1 Eyebrows

For the eyebrows, FAP31-36 are implemented. Inspired by [15], we adopt weighted sinusoids to parameterise the eyebrow animation with the FAPs. Fig. 4 illustrates the bidirectional animation trajectory of one of the boundary curves representing the left eyebrow in a sinusoidal manner. The middle horizontal dotted line in black stands for one of boundary curves in a neutral position, i.e. with all the relevant FAPs at the zero level. This segment of boundary curve is acted on by three FAPs. The three FAPs have different motion effects. When only FAP33 functions, the whole curve bends up and down sinusoidally, as depicted by the darkly dotted sine curves directed by the black dashed arrows. When we want to raise the left inner or outer eyebrow, FAP31 or FAP35 will be switched on, respectively, causing the rise of the left eyebrow corners in an inverse sinusoidal way, as highlighted in grey. By doing so, we not only

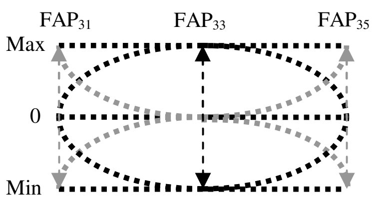


Fig. 4. Eyebrow animation in a sinusoidal fashion

ensure a smooth simulation of the eyebrow animation, but also guarantee that all the boundary curve points involved can meet at a horizontal line, i.e. the uppermost horizontal dark dotted line in Fig. 4 when all the three FAPs are set to maxima.

The above sinusoidal animation of the left eyebrow can be formatted by Equation (2) and (3):

$$y'_{j,i} = y_{j,i} + FAP_{33} \cdot \sin\left(\frac{\pi(t_2 - i)}{(t_2 - t_1)}\right) \cdot \left(y_{j,i} - y_{j+1,i}\right) + FAP_{31}\left(1 - \sin\left(\frac{\pi(t_2 - i)}{(t_2 - t_1)}\right)\right) \cdot \left(y_{j,i} - y_{j+1,i}\right), \qquad i \in ((t_2 - t_1)/2, t_2] \tag{2}$$

$$y'_{j,i} = y_{j,i} + FAP_{33} \cdot \sin\left(\frac{\pi(t_2 - i)}{(t_2 - t_1)}\right) \cdot \left(y_{j,i} - y_{j+1,i}\right) + FAP_{35}\left(1 - \sin\left(\frac{\pi(t_2 - i)}{(t_2 - t_1)}\right)\right) \cdot \left(y_{j,i} - y_{j+1,i}\right), \qquad i \in [t_1, (t_2 - t_1)/2] \tag{3}$$

where $y_{j,i}$ and $y'_{j,i}$ indicate the y coordinate values of the ith ($t_1 \le i \le t_2$) point on the jth boundary curve before and after animation, respectively. t_1 and t_2 delimit the extent of the facial boundary curves for the left eyebrow. Equation (2) formulates the right half of boundary curve animation for the left eyebrow with FAP31 and 33, while Equation (3) formulates the left part of boundary curve animation the left eyebrow with FAP 33 and 35. Similar formulae can be applied to the right eyebrow with FAP32, 34 and 36.

4.2 Eyes and Mouth

The MPEG-4 standard defines 12 FAPs for the eyeballs, pupils and eyelids. However, since the PDE boundary curves do not cover much detail in the eyes, we have only implemented FAP19 and 20 here, which dominate the eyes opening and closing in a linear manner.

Similarly, according to the characteristics of the PDE boundary curves, we have implemented four FAPs for animation of the mouth corners, which are FAP53, 54, 59 and 60. For raising the lip corners, governed by FAP59 and 60, we use the same strategy as developed above for animation of the eyebrow corners, which is sinusoidal. As for stretching the lip corners, governed by FAP53 and 54, a linear interpolation is applied to those boundary curve points representing the lips.

4 Results

To offer an intuitive interaction with the PDE face, we have developed a PDE face generator, a snapshot of which is given in Fig. 5. This uses the boundary curves, shown as points in the left hand window, to synthesise a textured PDE face shown in the right hand window. Animation is by moving the sliders.

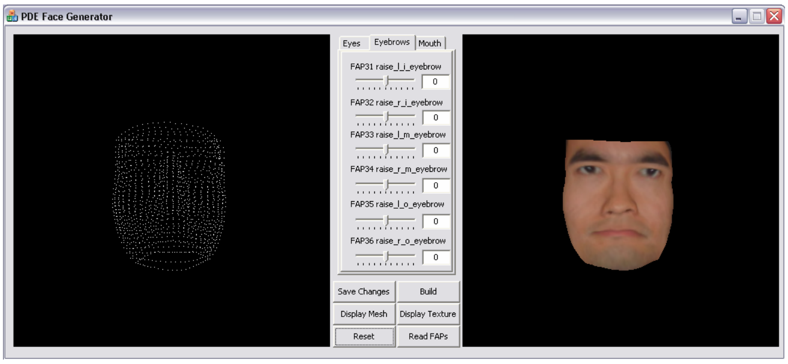
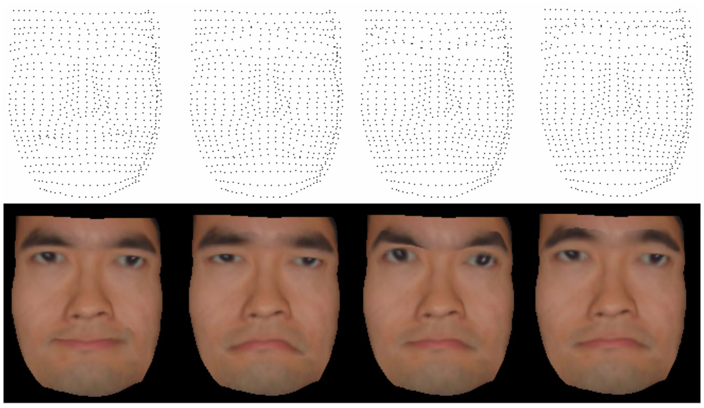


Fig. 5. PDE face generator with the boundary curves and a textured PDE face in neutral



(a) (b) (c) (d)

Fig. 6. FAPs-driven PDE facial expressions. (a) Joy (b) Sadness (c) Anger (d) Fear.

The texturing method adopted here is named *uv*-mesh texture mapping [13]. This method, assisted by the mesh warping technology, associates the texels of the texture map with the vertices of the *uv* mesh in the PDE face so that it eliminates the 3D-to-2D association routinely required in texture mapping, simplifying the process. Moreover, this method allows synthesis of the textured 3D face without the need that the facial expression in the texture map must exactly match that of the 3D face model.

Research in facial expression has concluded there are six universal categories of facial expressions [2]. We show the names and corresponding descriptions of four of these facial expressions in Table 1, along with the names of the FAPs involved. Fig. 6 shows the four synthesised PDE facial expressions, as well as the animated boundary curves in the upper row of the figure. For the sake of visibility, only the frontal halves of the facial boundary curves are shown here. However, the complete boundary curves are required in PDE face generation.

Table 1. Description to implementation of FAPs-driven PDE facial expressions

Expressions	Descriptions	FAPs
Joy	The eyebrows are relaxed. The mouth corners are pulled back toward the ears.	53,54, 59 and 60
Sadness	The inner eyebrows are bent upward. The eyes are slightly closed. The mouth corners are pulled downward.	19, 20, 31, 32, 59 and 60
Anger	The inner eyebrows are pulled downward and together. The eyes are wide open. The lips are pressed against each other or opened to expose the teeth.	19, 20, 31, 32, 35, 36, 53 and 54
Fear	The eyebrows are raised and pulled together. The inner eyebrows are bent upward.	31-36

6 Conclusion

Direct parameterisation has been the most widely used technology in facial animation. For those face models with a very large number of vertices, most of the direct parameterisation techniques adopt a number of control points to interpolate other vertices of the 3D model. Such an implementation is, however, rather complicated. In this paper, we proposed a new solution by exploiting a PDE-based facial animation scheme. The idea behind this is that the PDE method is capable of generating animations of a complex face model, by adjusting only a small group of facial boundary curves, without involving conventional control points and surface interpolation. The complexity of the synthesized PDE face is controlled by the resolution of the *uv* mesh. This means that the PDE-based method allows one animation scheme to be used for different face models in various resolutions, without any other extra process. This paper also developed a FAPs-driven animation scheme, using sinusoids to simulate the motion of the eyebrows and mouth. Finally, the experiments showed some encouraging results by using the PDE-based facial animation method to simulate emotional facial expressions.

Since the existing boundary curves do not include the detail of the mouth, some facial animations, such as mouth opening and closing etc, cannot be implemented with the current model. Hence, our future research will focus on this.

References

1. Parke, F.: Computer Generated Animation of Faces. In: Proc. of ACM Annual Conference, Boston, United States, vol. 1, pp. 451–457 (1972)
2. Parke, F., Waters, K.: Computer Facial Animation, Wellesley, Massachusetts (1996)
3. Ju, E., Lee, J.: Expressive Facial Gesture from Motion Capture Data. In: Proc. of EUROGRAPHICS 2008 Conf., Crete, Greece, April, pp. 381–388 (2008)
4. Animazoo, `http://www.animazoo.com`
5. Patel, M., Willis, P.: The Facial Animation, Construction and Editing System. In: Proc. of EUROGRAPHICS 1991 Conf., pp. 33–45 (1991)

6. Terzopoulos, D., Waters, K.: Analysis and Synthesis of Facial Image Sequences Using Physical and Anatomical Models. IEEE Trans. on Pattern Analysis and Machine Intelligence 15(6), 569–579 (1993)
7. Zhang, Y., Prakash, E., Sung, E.: A New Physical Model with Multilayer Architecture for Facial Expression Animation Using Dynamic Adaptive Mesh. IEEE Trans. on Visualization and Computer Graphics 10(3), 339–352 (2004)
8. Hui, K., Leung, H.: Expression Modeling–a Boundary Element Approach. Computers & Graphics 30(6), 981–993 (2006)
9. Ahlberg, J.: CANDIDE-3: an Updated Parameterised Face, Report No. LiTH-ISY-R-2326, Linkoping University, Sweden (January 2001)
10. Sheng, Y., Sadka, A., Kondoz, A.: Automatic Single View-based 3D Face Synthesis for Unsupervised Multimedia Applications. IEEE Trans. on Circuits and Systems for Video Technology 18(7), 961–974 (2008)
11. Lee, W., Magnenat-Thalmann, N.: Fast Head Modeling for Animation. Image and Vision Computing 18, 355–364 (2000)
12. Gonzalez Castro, G., Ugail, H., Willis, P., Palmer, I.: A Survey of Partial Differential Equations in Geometric Design, the Visual Computer 24(3), 213–225 (2008)
13. Sheng, Y., Willis, P., Gonzalez Castro, G., Ugail, H.: PDE Face: a Novel 3D Face Model. In: Proc. of the 8th IASTED Int. Conf. on Visualization, Imaging, and Image Processing, Palma, Spain, September, pp. 408–415 (2008)
14. Ugail, H., Bloor, M., Wilson, M.: Techniques for Interactive Design Using the PDE Method. ACM Trans. on Graphics 18(2), 195–212 (1999)
15. Gonzalez Castro, G., Ugail, H., Willis, P., Sheng, Y.: Parametric Representation of Facial Expressions on PDE-based Surfaces. In: Proc. of the 8th IASTED Int. Conf. on Visualization, Imaging, and Image Processing, Palma, Spain, September, pp. 402–407 (2008)