Automatic markers' influence calculation for facial animation based on performance capture

Damian Pęszor[1,2], Konrad Wojciechowski[2], and Marzena Wojciechowska[2]

[1] Institute of Informatics, The Silesian University of Technology, Akademicka 16, 44-100 Gliwice, Poland
[2] Polish-Japanese Academy of Information Technology, Koszykowa 86, 02-008 Warsaw, Poland

Abstract. Marker-based performance capture is a technique that enables acquisition of expression and mimicry of human face. This data can be used to propel facial animation system, be it bone driven or similarly dependant on position of points in space. Every model that is to be animated has to be analyzed in order to select level of influence each marker has over each vertex of said model. This process can be quite tedious if done manually. In this paper we present an approach for automatic calculation of markers' influence based on position of vertex on human face's surface obtained by acquisition using structured light-based scanner or similar approach.

Keywords: marker influence, vertex weight, facial animation, performance capture, face scan, bone driven animation, fiducial points

1 Introduction

Performance capture is a technique of acquiring data about deformation of surface of actor's face in order to capture his mimicry for later use in facial animation. While there are many specific methods to implement performance capture, most of them are based on tracking reflective markers that are either painted on face's surface or attached to it. Once obtained, performance capture data is used to create realistic facial animation. There are few techniques to create facial animation, each having different flaws and benefits. Texture based animation, while quite common in two dimensional applications, looks poorly in three dimensional environment. Physiological models based on tissue, muscle and skeleton of human face give very good results, but level of complexity and thus computational cost are big enough to make this approach unsuitable for most real time animation. Blending of different poses has low computational cost and is applicable to 3D environment, but is model-specific and therefore requires huge amount of work for every animated mesh. Bone driven animation is most commonly used because of low computational and memory cost. This technique is able to imitate subtle transformations of facial skin and is usable for not only single mesh but also those with similar morphology. The biggest problem with bone driven animation is the need to assign a weight for each vertex-bone pair that

will represent influence this bone has over given vertex. Since every marker can be considered the end of the bone, this issue can also be seen as defining specific marker's influence over vertices. If markers are positioned on facial surface with anthropometric features in mind rather than in simple grid arrangement, one can find fiducial points on animated model. Those points can then be used to obtain vertices' weights that are specific for animated model, thus eliminating both major disadvantages of bone driven animation.

2 Facial Mesh

The method presented in this paper is designed to calculate influence of markers placed on human face. It can be, however, used on different surfaces that have underlying structure, given that points on this surface can be detected similarly to fiducial points on facial mesh or can be manually selected. In case of artificial models of human face, the quality of the model determines if it can be analyzed the same way as realistic model used in this research. In case of low quality models (ones that do not preserve realistic curvatures of facial surface) or models that are representation of damaged, deformed, artistic or non-human faces, manual detection of fiducial points might be needed.

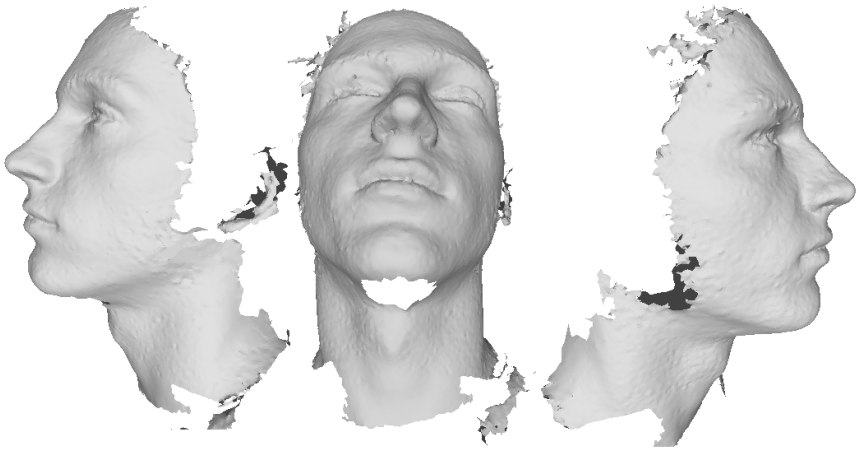

Fig. 1. Mesh as acquired by 3dMDface System.

2.1 Mesh acquisition

While artistic representations of human face are often simplified and therefore easy to animate, those that are realistic can prove to be a difficult subject to

animation. The more realistic the face, the easier it is to spot parts of model that move differently than natural mimicry. Therefore, for the purpose of this paper, meshes of real faces have been acquired and analyzed. Depending on technique used (stereo vision, structured light, time of flight, etc.), various type of errors can be introduced in acquired mesh. Lens distortion, lighting differences, interference from another camera can all lead to errors. Faces used in this research were obtained using 3dMDface system, a structured light based scanner, at The Institute of Theoretical and Applied Informatics of Polish Academy of Sciences in Gliwice. Being structured light based, this equipment produces errors on surfaces that are highly reflective, such as hair, glossy skin or eyes, due to incorrect determination of depth. Most of expressive facial area is however well preserved and therefore can be used as a basis for animation. Eight different actors were scanned for a total of forty models representing neutral facial expression, sample of which are presented in Fig. 2.

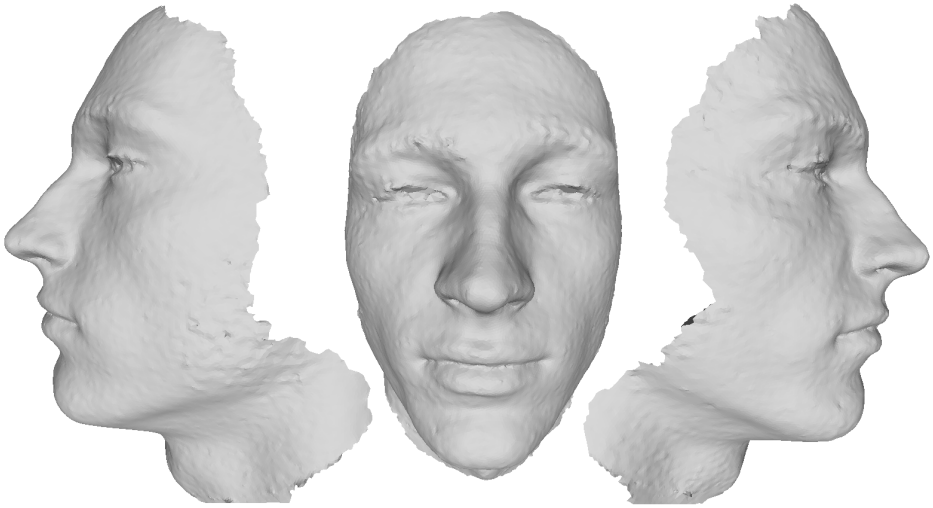

Fig. 2. Example of preprocessed mesh

2.2 Mesh preprocessing

Meshes obtained using structured light or other methods are prone to various types of errors. First of all, those meshes often contain more than single continuous surface, therefore any surfaces out of face's expressive area should be removed. It is typical for the surface that has biggest number of vertices to be the one that contains facial expressive area, so only this surface has to be preserved. Oftentimes scanned models also contain topological errors like non-manifold edges and vertices which have to be removed in order to correctly

estimate fiducial points. Furthermore, due to the noise present, edges of the analyzed surface might be composed of vertices that do not represent skin affected by mimicry. Those can be safely removed in order to make sure that they are not affected by facial expressions' animation. Depending on how face is oriented to the scanner, facial hair and other factors, it is probable, that resulting mesh will contain gaps or holes. Those need to be filled with respect to preserving curvatures of human head, for which [1] was used.

3 Model Division

In order to calculate influence of each marker over each vertex, one has to be able to distinguish markers that affect given vertex from those which do not. All vertices of expressive facial area lie between fiducial points that can be found on face structure, therefore finding those fiducial points that correlate to marker placement is essential to calculate markers' influence on every vertex. In some cases, such as testing marker arrangement prior to performance capture session, manual selection of fiducial points can be useful. In most cases, however, this process can be automated.

3.1 Estimation of marker placement

Most of the methods for finding fiducial points on human face are based on colour features present in two dimensional frontal pictures. Few methods that are capable of finding fiducial points in three dimensional mesh either project it to two dimensional space in order to apply 2D based algorithms, or at least use colour data instead of mesh structure. This allows to find some fiducial points but is not suitable for those points that describe structure of the face in a way that could be useful in terms of retargeting the mimicry to face of different actor than the one whose performance was captured. [2] presents a method mainly based on curvatures of three dimensional mesh. Few improvements were made to this method; all fiducial points were manually selected on neutral model to compensate for points that are difficult to find using curvatures. Initial two dimensional window of 96x96mm used for estimation of nose tip of mesh composed of vertices $V = (v_1, v_2, ..., v_n)$ has been replaced with sphere. This sphere is located in neutral face's nose tip p_{nt} of radius r such that:

$$r = \frac{\| p_{nt} - \frac{(\sum_{i=1}^{n} v_i)}{n} \|}{2} l \tag{1}$$

Next, nose tip was differentiated from surroundings using gradient based method by traversing from vertices closest to centroid of the model to farthest. This allowed to place the sphere in farthest point from centroid while reducing it's radius 5 times. At this point curvature estimation based on paraboidal fitting [3] is much more likely to correctly point to nose tip if it's calculated using more than one ring neighbourhood. Similar approach was used to find other

fiducial points. Points that do not have specific curvature can be found using their correspondent position on neutral model with relation to those fiducial points which were already found. Position of marker m_j with normal N on analyzed mesh can then be found as a vertex v_i such that:

$$\arg\min_i \frac{\|v_i - m_j\|((v_i - m_j) \cdot N)}{\|v_i - m_j\|\|N\|} \tag{2}$$

3.2 Triangular division of the model

Having found fiducial points one can divide the model into triangles based on those found points. All fiducial points should be connected into a topologically correct mesh with few rules in mind. First, every triangle should represent possibly flat area - this way all edges can be extended into planes that will bisect facial model easily. Second, each triangle should be possibly equilateral in order to avoid uneven distribution of influence. Third, triangles should be possibly small, so distance from marker to vertex that is affected by said marker will be small as well. Since surface of the mesh is not coplanar with triangles connecting markers, there is no clear association between vertex and triangles near it. Obtaining such association, however, is needed in order to limit the number of markers that have influence over specific vertex to those that are near it. Those markers represent surfaces that are affected by same transformation as the one sampled in analyzed vertex. Two methods were designed to obtain this association.

Geometric method is based on spatial coordinates of analyzed vertex with relation to marker's position. Each markers' based triangle represents a plane in three dimensional space. Edge of triangle therefore represents intersections of two planes. Using spherical linear interpolation [4] of normals, one can obtain a third plane intersecting through the edge with same angle to both of triangles' planes. Each vertex lies on either side of this plane and eventually - inside of pyramid based on edges of one of triangles. In some cases, e.g. in vertices lying on cheeks, vertex can be inside two or more pyramids - one correct, and others on far side of the face. Distance between the triangle's plane (alternatively, it's centroid) and vertex can then be used to verify which triangle should be associated with specific vertex. This reveals the main problem of the method - the pyramids might extend beyond expressive area thus resulting in non-expressive surface being dependant on markers' movement. The remedy for that is to cut out of the model any non-expressive areas or use graph-based method instead.

Graph-based method uses mesh structure to obtain vertex-triangle association. Each marker is associated with vertex that is topologically connected to every other marker through some other vertices. One can therefore assume that with marker arrangement based on anthropometric features, shortest path between markers corresponds to smallest distance. Using A* [5] algorithm to obtain paths between neighboring markers results in finding edge of markers' based triangle on actual surface model similarly to extending planes in geometric method. Instead of testing for side of plane, breadth first search is used to

decide on which side of the path vertex is located. Due to the nature of triangular mesh, one has to select one of middle vertices of the edge's path and start the search from this vertex to the direction of opposite marker. All vertices found between three edges' path can be associated to triangle composed of those edges. This method ensures that no vertices from outside of expressive area are associated with markers. The main flaw of this method is that it is dependant on topology of the mesh rather than surface itself, which can lead to problems in case of structured light induced errors or markers placed with low density.

Using graph method to cut outer edge of expressive area before using geometric method proves to give best results. Graph method can also be used to improve estimation for those vertices that are contained by two or more pyramids in geometric method.

4 Influence calculation

Each vertex is positioned on surface in three-dimensional space. Each part of surface is based on triangle and therefore each vertex can be projected onto this triangle. Similarly to the method used to divide the model, one can define a plane as extension of triangle's edge by using normal to the triangle's surface. In all cases except the edge of the expressive area, there will be a neighboring triangle on the other side of analyzed edge. Since both triangles can have different orientation, there is a portion of space that is placed between triangles' normals' based planes. To cover for that, neighboring triangles' normals are interpolated using spherical linear interpolation, so that plane based on interpolated normal would split space equally for both triangles.

Having three planes containing the space, each described by a point (marker) $\{A, B, C\}$ lying on it and it's unit normal n_i, a point of intersection p_{int} of all three can be found.

$$p_{int} = \begin{vmatrix} n_{00} n_{10} n_{20} \\ n_{01} n_{11} n_{21} \\ n_{02} n_{12} n_{22} \end{vmatrix}^{-1} [(A \cdot n_0)(n_1 \times n_2) + (B \cdot n_1)(n_2 \times n_0) + (C \cdot n_2)(n_0 \times n_1)] \quad (3)$$

A line between point p_{int} on top of pyramid and vertex v_i inside of it can be extended to intersect with the plane of triangle that is a basis of said pyramid and has normal n and a marker-related point A lying on it. The intersection found is projection v_i' of v_i onto the base triangle.

$$v_i' = (v_i - p_{int}) \frac{(A - p_{int}) \cdot n}{(v_i - p_{int}) \cdot n} + p_{int} \quad (4)$$

Having found this point on triangle's surface, barycentric coordinates are calculated with relation to triangle's vertices.

$$
\begin{aligned}
a &= \frac{(y_B - y_C)(x_{v'} - x_C) + (x_C - x_B)(y_{v'} - y_C)}{(y_B - y_C)(x_A - X_C) + (x_C - x_B)(y_A - y_C)} \\
b &= \frac{(y_C - y_A)(x_{v'} - x_C) + (x_A - x_C)(y_{v'} - y_C)}{(y_B - y_C)(x_A - X_C) + (x_C - x_B)(y_A - y_C)} \\
c &= 1 - a - b
\end{aligned} \tag{5}
$$

Barycentric coordinates could be identified with influence each of surrounding three markers has over the vertex. This, however, would result in differences between vertices that are near of each other, but are assigned to different triangles due to the edge separating them. Those differences, visible as sharp, unwanted features of the model, need to be smoothed. It is also worth to note, that in most cases, especially when the marker arrangement is not dense, markers on neighboring triangles do affect how skin is deformated inside of the analyzed triangle, so those also need to be taken into account.

Surface of triangle is therefore based on it's three vertices and three another points. This is exactly how cubic Bézier triangle is constructed. [6] Surface of cubic Bézier triangle composed of n points $p_i | i \in \mathbb{N} \wedge i \in [0, n-1]$ each having barycentric coordinates of a, b, and c related to triangle's vertices α^2, β^2 and γ^2 respectively and influenced by control points $\alpha\beta$, $\alpha\gamma$ and $\beta\gamma$ is described with following equation:

$$
p_i(a,b,c) = (\alpha a + \beta b + \gamma c)^2 = \alpha^2 a^2 + \beta^2 b^2 + \gamma^2 c^2 + 2\alpha\beta ab + 2\alpha\gamma ac + 2\beta\gamma bc \tag{6}
$$

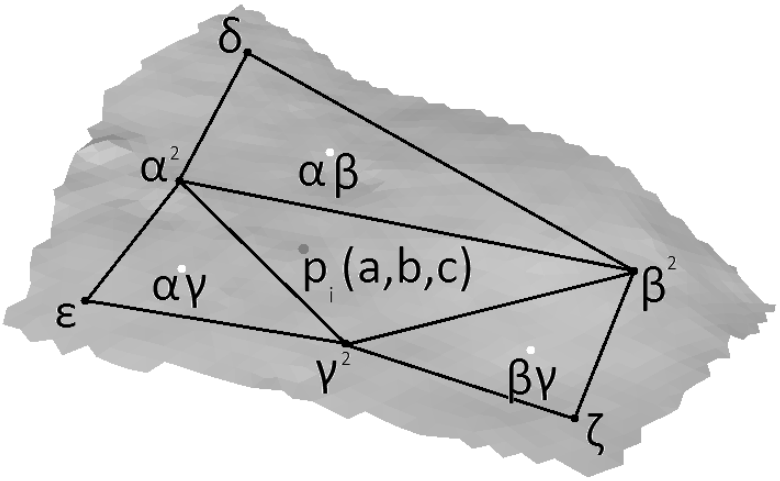

Fig. 3. Example of surface. Black dots represent markers, white dots represent additional control points of Bézier triangle and grey dot represents example point

Since projected barycentric coordinates, vertices of triangle and control points are all known, one could assume that equation can be used directly. This is however not the case, because appropriate control points (vertices of neighboring triangles) will not result in obtaining surface expressed by points v_i due to the difference between points' projection and points themselves. Instead of directly calculating change in barycentric coordinates that would cover for that, another method is used. First, barycentric coordinates are assumed to be correct and control points are estimated using nonlinear least squares fitting [7] in order to find a surface approximating actual points' positions. Once obtained, each control point weighted by barycentric coordinates is replaced with position of third vertex of neighboring triangle $(\delta, \epsilon, \zeta)$ with new weight. Similarly, weight of each triangle's vertex is calculated:

$$\begin{aligned} A &= a^2 \\ B &= b^2 \\ C &= c^2 \\ D &= \frac{2\alpha\beta ab}{\delta} \\ E &= \frac{2\alpha\gamma ac}{\epsilon} \\ F &= \frac{2\beta\gamma bc}{\zeta} \end{aligned} \tag{7}$$

This can be used as weighting term for corresponding markers, so the difference between marker's position in neutral frame and in current frame f can be expressed as:

$$\begin{aligned} p_{i,f}(A,B,C,D,E,F) = p_{i,0}(A,B,C,D,E,F) + A(\alpha_i - \alpha_0) \\ +B(\beta_i - \beta_0) + C(\gamma_i - \gamma_0) + D(\delta_i - \delta_0) + E(\epsilon_i - \epsilon_0) + F(\zeta_i - \zeta_0) \end{aligned} \tag{8}$$

Various ways of applying offset using e.g. normal to surface can be used instead to further enhance wrinkles and similar details of facial surface.

5 Exemplary results

The results of using weights obtained by automatic calculation of influence presented in this paper can be seen in Fig. 4. One can notice that using weights obtained in this matter produces unrealistically smooth surfaces (as seen on cheek in bottom-right picture) or a valley-like surface in the position of marker (as seen in top-right picture), which do not accurately correspond to how real skin reacts. This can be however compensated by reducing animation impact through using appropriate initial vertices' positions. This is also a good way to reproduce wrinkles and similar features.

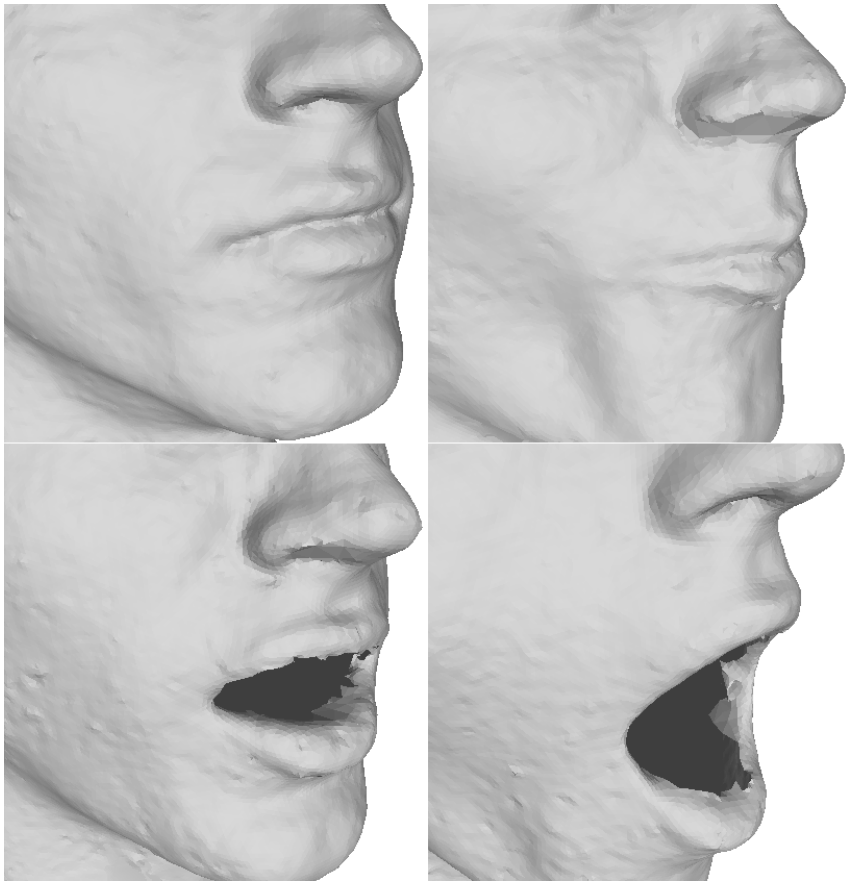

Fig. 4. Surface with calculated markers' influence, original and distorted by movement of markers

6 Conclusion and further works

The method presented in this paper allows to automatically estimate markers' influence over vertices of human facial model. While this seems to result in acceptable weights estimates, it is still clear that using simple offset, as typical in bone-driven animation, will not produce satisfactory results in presenting skin deformations. A different approach, based on distortions dependant on interpolated normals could possibly be applied in order to improve the way distortions are animated, which will be part of future research. Still though, this work provides easy and fast way to estimate how surface is affected by changes in markers' positions and therefore is useful in both further research in animation automatization and practical applications. Apart of producing animation itself, this can be also useful to test markers' positions on facial surface and it's effects on possible expressions on mesh used before comitting to expensive performance

capture procedure without need to manually select weights for each possible marker arrangement.

Acknowledgments. This project has been supported by the National Centre for Research and Development, Poland.(project INNOTECH In-Tech ID 182645 "Nowe technologie wysokorozdzielczej akwizycji i animacji mimiki twarzy." - "New technologies of high resolution acquisition and animation of facial mimicry.")

References

1. Zhao W., Gao S., Lin H. "A Robust Hole-Filling Algorithm for Triangular Mesh" in *The Visual Computer: International Journal of Computer Graphics*, Springer-Verlag New York, Inc. Secaucus, USA, Vol. 23 Issue 12, November 2007, pp. 987-997.
2. Shalini Gupta, Mia K. Markey and Alan C. Bovik "Anthropometric 3D Face Recognition" in *International Journal of Computer Vision*, Springer US, 2010, Volume 90, Issue 3, pp. 331-349.
3. T. Surazhsky, E. Magid, O. Soldea, G. Elber, E. Rivlin "A comparison of Gaussian and mean curvatures estimation methods on triangular meshes" in *Proceedings of IEEE International Conference on Robotics and Automation*, Vol. 1, 2003, pp. 1021-1026.
4. Shoemake K. "Animating rotation with quaternion curves" in *SIGGRAPH '85 Proceedings of the 12th annual conference on Computer graphics and interactive techniques*, pp. 245-254, July 1985
5. Hart P. E., Nilsson N. J., Raphael B. "A Formal Basis for the Heuristic Determination of Minimum Cost Paths" in *IEEE Transactions on Systems Science and Cybernetics*, IEEE, Vol. 4, Issue 2, July 1968, pp. 100-107
6. Farin, G. "Curves and Surfaces for CAGD", 5th ed., Academic Press. ISBN 1-55860-737-4
7. Bates D. M., Watts D. G., "Nonlinear Regression Analysis and Its Applications", New York, Wiley, 1988, ISBN 978-0-471-81643-0