HAL
archives-ouvertes.fr

Optimal Marker Set for Motion Capture of Dynamical Facial Expressions

Clément Reverdy, Sylvie Gibet, Caroline Larboulette

▶ **To cite this version:**

Clément Reverdy, Sylvie Gibet, Caroline Larboulette. Optimal Marker Set for Motion Capture of Dynamical Facial Expressions. MIG2015 8th ACM SIGGRAPH Conference on Motion in Games , Nov 2015, Paris, France. pp.31-36, 10.1145/2822013.2822042 . hal-01279483

HAL Id: hal-01279483
https://hal.archives-ouvertes.fr/hal-01279483

Submitted on 26 Feb 2016

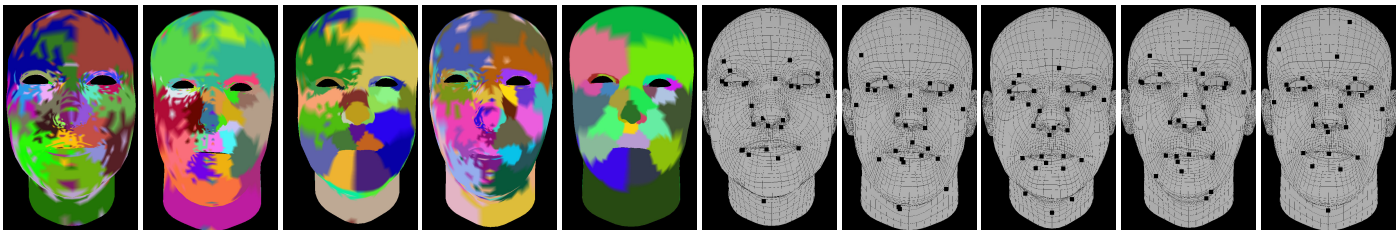
HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Optimal Marker Set for Motion Capture of Dynamical Facial Expressions

Clément Reverdy* Sylvie Gibet† Caroline Larboulette‡

University of South Brittany



(a) (b) (c) (d) (e) (a') (b') (c') (d') (e')

Figure 1: *Clusters and corresponding marker sets automatically determined by applying our K-means clustering algorithm (with $K = 30$ clusters) on the* range-of-motion *sequences of actors A (a,a'), B (b,b'), C (c,c'), D(d,d') and combined sequences of actors B+C+D (e,e').*

Abstract

We seek to determine an optimal set of markers for marker-based facial motion capture and animation control. The problem is addressed in two different ways: on the one hand, different sets of empirical markers classically used in computer animation are evaluated; on the other hand, a clustering method that automatically determines optimal marker sets is proposed and compared with the empirical marker sets. To evaluate the quality of a set of markers, we use a blendshape-based synthesis technique that learns the mapping between marker positions and blendshape weights, and we calculate the reconstruction error of various animated sequences created from the considered set of markers in comparison to ground truth data. Our results show that the clustering method outperforms the heuristic approach.

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation;

Keywords: Facial animation, clustering, K-means, Gaussian Process regression

1 Introduction

The animation of virtual characters has numerous applications, from entertainment such as video games or movies, to other serious game applications involving interaction with avatars for communication or education purposes. To make these avatars appear more attractive and realistic, special care must be taken at several levels of animation, e. g., behavior, body and hand movement, and facial animation. Using captured motion on real actors provides the ability to animate virtual characters with credible behavior and thus reinforce their comprehension and acceptability. Furthermore, data-driven approaches go beyond the production of realistic animations: they allow for a detailed analysis that might help to extract and understand some relevant key postures and dynamical features, and make possible the use of pre-recorded motion, through editing and synthesis operations. However, this approach is still challenging the computer animation community, in particular for multi-channel animation involving the simultaneous control of facial expression, hand and body movements in expressive or linguistic tasks. In this paper we focus on expressive facial animation with the aim to further linguistically edit our data within a concatenative synthesis framework dedicated to virtual signers [Gibet et al. 2011].

There are two possibilities to capture facial motion: either marker-based or markerless techniques. Different reasons may be argued to prefer one over the other solution. As our future aim is to animate signing avatars through motion capture, our application requires to capture full-body movements, including body and hand motion, facial expression and gaze direction. In this particular case, marker-based motion capture (MoCap) is necessary because the different body channels need to be captured simultaneously, each channel conveying a specific meaning.

One challenge in facial marker-based motion capture is the choice of the marker layout. Numerous empirical facial MoCap layouts have already been proposed to capture facial expressions and control facial animation systems. However little work has been done to determine and evaluate the best marker set for motion acquisition and animation.

In this paper, we propose a method to compute an optimized marker set for facial motion acquisition and synthesis control. Our aim is not so much to produce animations with high accuracy and realism as this is done in [Le et al. 2013], but to be able to easily capture new sequences of facial and full-body motions, avoiding as much as possible the intervention of skilled animators. That is why we seek the best trade-off between complexity (we try to minimize the number of markers required) and the quality of the produced animation (which should best render the facial expressiveness). In addition we want to remain as independent as possible of the synthesis method to facilitate the reutilization of the corpus in different contexts. We propose a dual heuristic / automated approach: after analyzing the capability of different empirical maker layouts used in previous studies to produce credible facial animations, we automatically determine marker layouts by using a clustering technique, and evaluate these marker layouts by using the same synthesis technique.

*e-mail:clement.reverdy@univ-ubs.fr

†e-mail:sylvie.gibet@univ-ubs.fr

‡e-mail:caroline.larboulette@univ-ubs.fr

In the remainder of the paper, we first review the related work. Then section 3 presents the methodology we have used to create our animations and to evaluate the efficiency of each marker set we have tested. Section 4 presents the data set on which the algorithms are applied. Section 5 presents the results obtained for empirical marker sets, while section 6 defines a novel approach to automatically compute optimized marker layouts and shows the corresponding results. Finally section 7 concludes and proposes directions for future work.

2 Related Work

Performance-Based Motion Capture. Marker-based facial motion capture usually consists in following the $3D$ positions of markers disposed on an actor's face with a network of cameras. This method presents the advantage of allowing a capture with a very high frame rate (≥ 120 fps) and a good precision (≤ 1 millimeter). However, since the number of markers that can be put on the actor's face is limited, this method can only output a sparse representation of the face. So if this technique efficiently captures large scale deformations, it is not sufficient for fine scale details like wrinkles.

In order to get a dense and direct mesh representation of the actor's face, several markerless methods relying on structured light [Zhang et al. 2008] or stereo cameras [Bradley et al. 2010] [Beeler et al. 2011] have been developed. These methods allow for the acquisition of a series of high resolution triangle meshes at 30 to 42 fps. However these methods are more sensitive to light conditions than the traditional marker-based motion capture techniques. Moreover, applications that necessitate simultaneous capture of body and facial motion are incompatible with the capture conditions required by these methods.

Marker Set Optimization. Amazingly, whereas the marker placement is a decisive choice, only few studies have been dedicated to this issue. Private companies usually exploit empirical marker layouts which are often homemade. In [Le et al. 2013] a method is proposed to automatically determine an optimal marker set, the optimization process relying on the minimization of the reconstruction error from the ground truth with respect to the chosen animation synthesis method - so that the marker set found can be optimized for this synthesis method in particular.

The problem of marker optimization is also similar to some of the issues related to mesh compression where the shape of the whole mesh may be determined from a subset of its vertices [Sorkine and Cohen-Or 2004], [Meyer and Anderson 2007], [Southern and Zhang 2011]. Our approach is similar to [Southern and Zhang 2011] and [Sattler et al. 2005] who have experimented K-means-based methods for mesh compression.

Animation Synthesis from Marker-Based Motion Capture. Facial animation by blendshapes from motion capture is a well-known technique. Following this method, several facial key shapes, designed most of the time by an animator, are blended to produce appropriate facial animations. Blendshapes present the advantage of providing both a level of abstraction and a compact representation which allows easiness of editing and retargeting to other blendshape-based facial models. However, it is still necessary to find a mapping between marker positions and blendshape coefficients. The quality of the animation is also dependent on the choice of key shapes.

In [Weise et al. 2011], the authors propose a method to provide a mesh representation of the user's face and its corresponding blendshapes [Weise et al. 2008; Li et al. 2009; Li et al. 2010] from data captured via a depth sensor camera. Then, blendshape weights are computed by minimizing a cost function taking into account both the geometry and the texture of the model.

Other methods such as the *least squares mesh* technique [Sorkine and Cohen-Or 2004] or the *thin-shell model* [Le et al. 2013; Botsch and Sorkine 2008] may be used. Both techniques are based on the same principle: the positions of all vertices of a mesh are directly estimated from the positions of a subset of these vertices.

In this paper, we do not focus on fine scale details but on easiness of data acquisition and genericity. For this reason we have chosen to rely on a blendshape animation system to compute facial deformations. Moreover, we argue that the system described in [Weise et al. 2011] is able to provide data of sufficient quality to both serve as training data for this algorithm but also to serve as *ground truth* for the evaluation of our results.

Cross-Mapping of Facial Data and Blendshape Parameters. The process of cross-mapping MoCap data and blendshape parameters is not trivial as it is a one-to-many mapping due to the fact that multiple blendshape weights combinations may lead to the same facial configuration. Traditional approaches identify pairs of MoCap data and blendshape parameters that are carefully selected and designed by the animator [Deng et al. 2006]. These pairs are then used in a learning process that determines the selection of corresponding blendshape parameters from new MoCap data input values. Other current methods largely rely on radial basis functions and kernel regression to achieve these steps [Cao et al. 2005; Deng et al. 2006; Liu et al. 2008]. However, such methods have several drawbacks: a number of localized basis functions have to be chosen prior to the learning process, and the result is conditioned by the quality and density of input data. Thus, noisy input often yields bad estimates, this being known as the classical over-fitting problem.

In our work, we need to simultaneously record body, manual and facial data at high frequency rates. Such technical difficulty may result in noisy positions of the facial markers. Therefore, the mapping between motion capture data and blendshape parameters is done via a machine learning algorithm. We consider the problem as a Bayesian inference problem. Instead of incorporating explicit basis functions (such as radial basis functions), we use a Gaussian Process regression technique to describe a distribution over functions that map the MoCap data and the blendshape parameters [Rasmussen and Williams 2005].

3 Methodology

We use two different approaches to find the best possible marker set: an empirical approach detailed in section 5 and an automatic clustering approach detailed in section 6. The empirical approach evaluates different facial MoCap marker layouts that have been used in previous work. The automatic approach takes as input the vertex positions of the facial mesh for all of the frames of a training sequence and computes through a clustering technique the optimized marker layout for a given number of clusters. Both approaches are evaluated through the same synthesis technique that achieves the mapping from motion capture data to blendshape weights.

3.1 Animation Synthesis

For each marker layout empirically or automatically determined, the synthesis can be decomposed into three steps: (i) for a long sequence of facial animation referred as *range-of-motion* sequence, we first take as training examples *markers-weights* pairs (X_i, Y_i) that describe the marker positions and the corresponding blendshape weights at each frame i; (ii) for this training data, a Gaussian

Process regression technique (GP) learns the mapping between the MoCap data and the weights; (iii) for new MoCap test data, the new blendshape weights are estimated from the same GP regression technique (see figure 2).

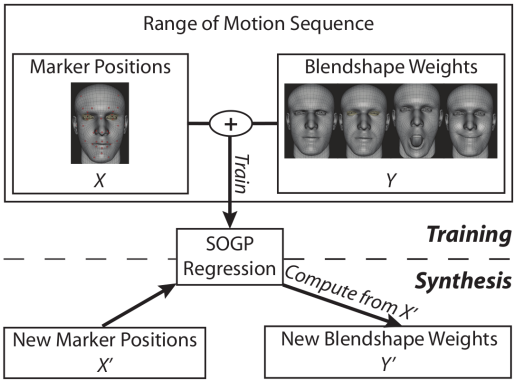


Figure 2: *Animation synthesis overview.*

Facial MoCap Representation. In traditional marker-based facial motion capture, captured data can be represented by a sequence of N facial poses over time $X = (X_1, ..., X_i, ..., X_N)^T$. Each facial pose X_i at frame i is encoded by a $3 \times K$ dimensional vector, K being the number of markers, and x_i^k the 3D position of the k^{th} marker: $X_i = (x_i^1, x_i^2, ..., x_i^K)$.

Blendshape Representation. In blendshape animation, a mesh is represented by a neutral shape and a set of basic deformations of this shape, where each basic deformation applied to the neutral shape represents a particular pose (e.g., one of the Action Units of the Facial Animation Coding System [Ekman and Friesen 1978]). Hence the shape B_i of the mesh at frame i is defined as a linear combination of these basic deformations:

$$B_i = B_0 + \sum_{l=1}^{L} w_i^l B^l \tag{1}$$

where B_0 is the neutral shape, L is the number of basic deformations, B_l is the l^{th} basic deformation and w_i^l its associated weight. Let $Y_i = (w_i^1, w_i^2, ..., w_i^L)$ be the vector of the L weights at frame i. We can express the blendshape animation sequence as the vector $Y = (Y_1, ..., Y_i, ..., Y_N)^T$.

Animation by Gaussian Process Regression. Formally, we consider a set of N observations $\{(X_i, Y_i), i = 1...N\}$, where X_i denotes the input vector (facial marker positions at frame i), and Y_i denotes the output vector (blendshape weights at frame i). In a GP model, we make the assumption of a double stochastic process on the distribution f and the noise ϵ_i:

$$Y_i = f(X_i) + \epsilon_i \tag{2}$$

and for all the observations:

$$Y = f(X) + \epsilon \tag{3}$$

The Gaussian Process is defined as:

$$f(X) = GP(\mu(X), K(X, X')) \tag{4}$$

where $\mu(X)$ is the mean function and $K(X, X')$ the covariance function. Based on the set of input-output observations, the Bayesian approach computes the posterior distribution of the real process f using the prior and the likelihood [Rasmussen and Williams 2005].

One major drawback prevents GP from being applied to large datasets: the computation of the covariance matrix which is highly costly. To overcome this limitation, we used a derived method called *Sparse Online Gaussian Process* (SOGP) which combines a sparse representation (using a smaller subset of input data) and an online algorithm of the posterior process [Csató and Opper 2002].

3.2 Quality Measurement

To evaluate the different marker sets, we compare the animations produced via our animation synthesis method using virtual markers with the animations produced by a reference animation system presented in section 4 and considered as the ground truth data. The quantitative measure of quality is based on the root mean squared error (RMSE) calculated from the positions of the vertices of the mesh for the ground truth data and the synthesized ones :

$$RMSE = \sqrt{\frac{1}{N} \cdot \sum_{n=1}^{N} \sum_{p=1}^{P} \|v_n^p - \hat{v}_n^p\|^2} \tag{5}$$

where N is the number of frames, P the number of vertices, v_n^p the position of the p^{th} vertex at the n^{th} frame, and $\hat{v}_n^p$ the position of the same vertex at the same frame in the ground truth mesh.

RMSE results presented in this study are computed over all frames of all test sequences performed by the concerned actor except the *range-of-motion* sequence (see section 4).

3.3 Notations

In section 5, we consider two categories of empirical marker sets: the first category, named *STAR*, denotes state-of-the-art marker layouts that have proven to be satisfactory for target applications; the second category, named *MAN*, denotes manual marker layouts manually defined from a given existing marker set. In section 6, each clustering experiment uses a marker set named *K-means*. Moreover, each K-means experiment is based on a *range-of-motion* sequence performed by an actor Φ. Therefore, a given marker set will be labeled *STAR*, *MAN*, or *K-means_*Φ, Φ representing the actor, $\Phi \in \{A, B, C, D\}$. It may be optionally followed by the number of markers. Given one marker set, the corresponding synthesis can be defined by its training sequence, applied on the *range-of-motion* of actor Ψ, and its test sequence applied on the test sequences of actor Ω. The synthesis will be labeled *train_*Ψ*, synth_*Ω.

4 Data Set

Since our synthesis process is based on learning algorithms, we need to get data on which these algorithms can be first trained and on which to rely as ground truth data to evaluate our resulting animations.

4.1 Ground Truth Data

We decided to use the Faceshift commercial software [fac 2012] as a reference system to easily collect the data required by the learning algorithms. We chose this system for several reasons. Firstly, it offers a flexible mesh and a set of blendshapes that match the face of the actor. The basic blendshapes are based on key shapes from the FACS coding system, thus producing dynamical facial animations that are quite similar to those of the human face. Moreover, this system provides a good approximation of the large scale facial deformations. Secondly, the technique employed by Faceshift produces pairs of selected virtual markers on the actor's face and blend-

shape weights. This allows us to learn the mapping of the *markers-weights* pairs, using our GP regression model, thus avoiding the tedious work of manually tuning the blendshape weights [Deng et al. 2006]. Using this system is a good way to collect a large amount of training data with different subjects.

To simulate each marker set with Faceshift (virtual markers), each marker has been positioned on a vertex of the actor's mesh. The training sequences consist of time series (about $1min30$ to $2min$ at 30 fps) of facial expressions performed by each actor.

For each actor, the mesh and the corresponding set of blendshapes optimized for this actor have been computed and exported. Regardless of the actor, the produced facial mesh has the same number of vertices (12021) and the same topology (connectivity matrix). The performances are captured via a depth sensor camera and the animations produced by Faceshift are output as the sequences Y of blendshape weights. These sequences are considered as a fair enough approximation of the ground truth for the purpose of this study.

4.2 Corpus

Our corpus contains the facial expressions of four non-deaf actors named A, B, C and D. There is one *range-of-motion* sequence per actor, designed for training and normalizations goals. Each actor/tress was instructed to freely explore the deformation capability of his/her face, making grimaces during $1min30$ to $2min$. The purpose of this sequence is to capture the facial deformation space specific to each actor. Apart from the training sequence, each actor has performed between 25 and 30 sequences lasting from $2s$ to $6s$ and recorded at a frequency of 30 fps. Each sequence has been performed once. For each sequence, the actor was instructed to watch a video and then mimic the facial expression he has seen. The videos are small sentences in French Sign Language with emotional content performed by deaf people.

4.3 Preprocessing

Each sequence X (input MoCap data) of a given marker set performed by a given actor is centered and scaled by respectively the mean vector and the standard error vector computed on the training *range-of-motion* sequence of this actor with the same marker set. The positions of the virtual markers for each sequence (both training and testing) have been centered and divided by the standard deviation of the training sequence of the corresponding actor.

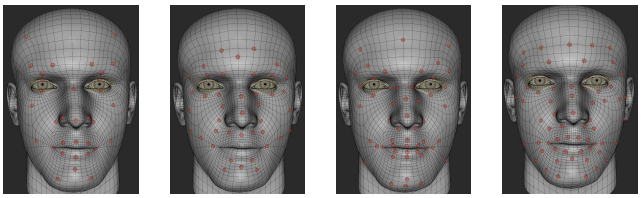
5 Empirical Marker Sets

In this section, we consider the following two categories of empirical marker sets: *STAR* and *MAN*. The results are presented with respect to the notation introduced in section 3.3.

5.1 State of The Art (STAR) Marker Sets

In our first approach we have tested different known marker sets of different sizes (see figure 3):

- MPEG4 (53 markers): the set of Facial Feature Points used in the MPEG4 standard;
- Face Robot (35 markers): the set of markers used in the commercial software Autodesk Face Robot (Softimage);
- SignCom (41 markers): the set of markers used in [Gibet et al. 2011] designed for French Sign Language (FSL) capture;



facerobot 35 markers | signcom 41 markers | mpeg4 53 markers | sign3d 60 markers

Figure 3: *STAR marker sets.*

- Sign3D (60 markers): the set of markers used in [Lefebvre-Albaret et al. 2013] designed for FSL capture.

We have then quantified the RMSE error between the ground truth data and the synthesized data. As shown in figure 4, this first experiment does not yield any significant quantitative results in terms of errors. Nevertheless, the spatial placement of markers (figure 3) is different from one marker set to another. Accordingly, errors on the synthesized data are distributed differently along time and space (face regions) for the various marker sets. For example (see figure 5), data synthesized with the sign3D marker set has a higher error rate on the eye region than the Face Robot marker set.

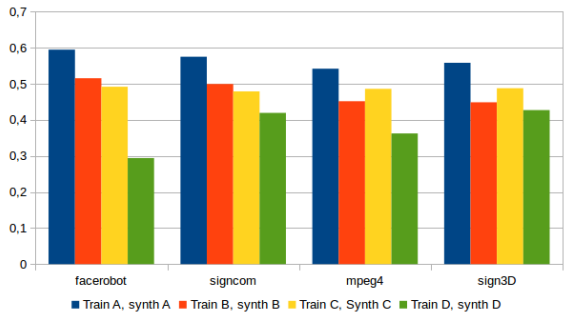


Figure 4: *RMSE of synthesis for the STAR marker sets and the same actors in training as in testing sequences.*

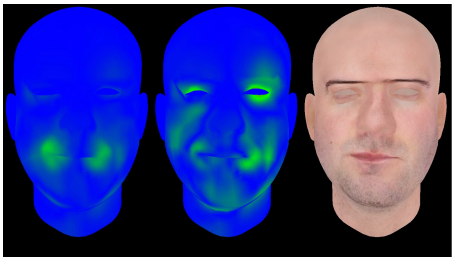


Figure 5: *Left: RMSE with Face Robot marker set; center: RMSE with Sign3D marker set; right: original animation sequence.*

5.2 Manual (MAN) Marker Sets

In order to analyze the influence of the number of markers on the synthesis quality, we manually established a set of manual marker layouts. We initially took a mix of 62 markers belonging to the STAR marker layouts, and then successively took off some markers from it. We thus formed four marker sets of sizes 62, 46, 30 and 15 (see figure 6 (b,c,d,e)). We removed in priority the markers that visually appeared the less useful.

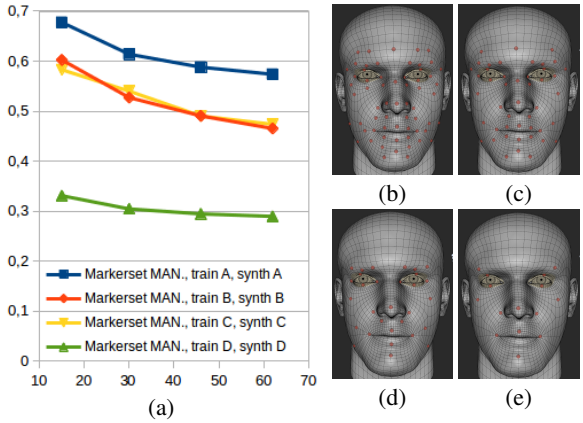


Figure 6: *(a): RMSE on manual marker sets, synthesis achieved with the same actor in training and test; (b, c, d, e): the MAN marker sets with respectively* 62, 46, 30 *and* 15 *markers.*

As expected, figure 6 (a) shows a constant decrease of the RMSE according to the number of markers. However, beyond 35 or 40 markers, we can observe on the basis of the RMSE error computed along the whole sequence, that the gain is not significant.

6 Automatic Determination of Marker Sets via Unsupervised Clustering

6.1 K-means Clustering Method

The key idea is to apply a clustering technique on the vertices of a training data set that mostly covers a large part of the human facial expression space. Let V be the matrix that represents the sequence of the mesh deformation where $\bar{v}_n^p = \frac{v_n^p - \mu_p}{\sigma_p}$ is the normalized position of the p^{th} vertex at the n^{th} frame (with μ_p and σ_p respectively the mean vector and standard deviation vector of the p^{th} vertex over time):

$$V = \begin{bmatrix} \bar{v}_1^1 & \cdots & \bar{v}_n^1 & \cdots & \bar{v}_N^1 \\ \cdots & \cdots & \cdots & \cdots & \\ \bar{v}_1^p & \cdots & \bar{v}_n^p & \cdots & \bar{v}_N^p \\ \cdots & \cdots & \cdots & \cdots & \\ \bar{v}_1^P & \cdots & \bar{v}_n^P & \cdots & \bar{v}_N^P \end{bmatrix} \quad (6)$$

In our approach, the p^{th} line V_p of this matrix (i.e. the whole trajectory of the p^{th} vertex along the training sequence) is considered as an observation. The K-means algorithm aims at partitioning these observations into K clusters C^k by iteratively assigning each observation to the nearest cluster C^k represented by its centroid m^k such as:

$$C^k = \{V^p / \|V^p - m^k\| \leq \|V^p - m^{k^*}\|, \forall k^* \in \{1, ..., K\}, k^* \neq k\} \quad (7)$$

We then recompute each cluster's centroid from its assigned vertices and repeat iteratively both operations until convergence. The centroids have been randomly initialized and only vertices that are affected by at least one blendshape are considered (4859 on 12021).

6.2 Results

We first analyze the influence of the training matrix on the quality of the markers found by the clustering K-means technique. As illustrated in figure 7, the evolution of RMSE over the number of markers doesn't change, whether the K-means is applied from a training achieved on one actor - actor A (left) or B (right) - or on several actors - actors B, C and D - for which the training sequences are concatenated. Moreover, we can see that the error rapidly decreases with the increase of the number of markers and reaches a minimum at about 30 markers. This preliminary analysis therefore highlights that taking a large amount of clusters will not lead to better results.

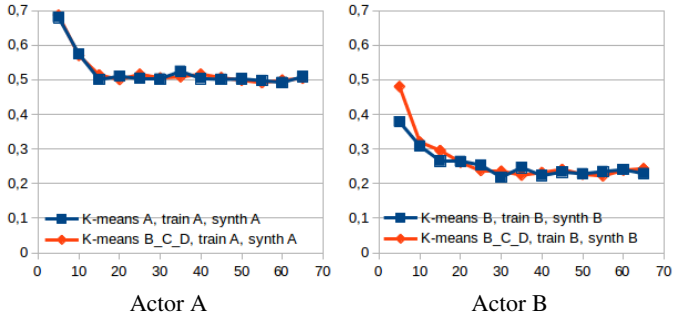


Figure 7: *Influence of the training dataset on the quality of markers found by clustering. Red: results with K-means trained on* 3 *actors; blue: results for K-means trained on only one actor.*

Nevertheless, as shown in figure 1, the regions that are determined by the K-means algorithm are less fragmented and the marker placement appears more symmetric when the clustering is performed on training data from multiple actors. For these reasons, it seems best to keep the marker sets automatically determined from the combined training data of the actors B, C and D.

We also compared the performances of the marker sets obtained by the K-means clustering technique with the empirical marker sets STAR and MAN applied to the four actors. figure 8 shows that the marker sets automatically determined via K-means clustering always leads to better results than the empirical marker sets.

7 Conclusion

We presented two approaches to help in the pose of markers for facial motion capture. In a first approach we tested empirical marker sets that have previously proven to give satisfactory results in blendshape animation from MoCap data. In a second approach, we used the K-means clustering method to partition facial meshes into geometrical regions that are significant for given actors. The results, quantified though the error between ground truth and synthesis animations, seem coherent. However, the automatic clustering method gives better results in terms of RMSE error. After a training on each actor, we showed in particular that it was possible, for a given number of clusters, to determine a good partitioning of vertices that show similar dynamics. Moreover, with a training performed over several actors, we showed that it was possible to identify more stable regions, and therefore to deduce reliable candidates for the pose of markers.

Nevertheless, some further investigations are required to complete this statement. In real conditions, the installation of markers is subject to inaccuracies, especially when there are several actors with different facial morphologies. Thus, the robustness to noise of the marker sets determined by this method is still to be verified. Future work that includes the use of these marker sets in real conditions will answer this question.

Furthermore, the facial representation used for facial animation, i.e.

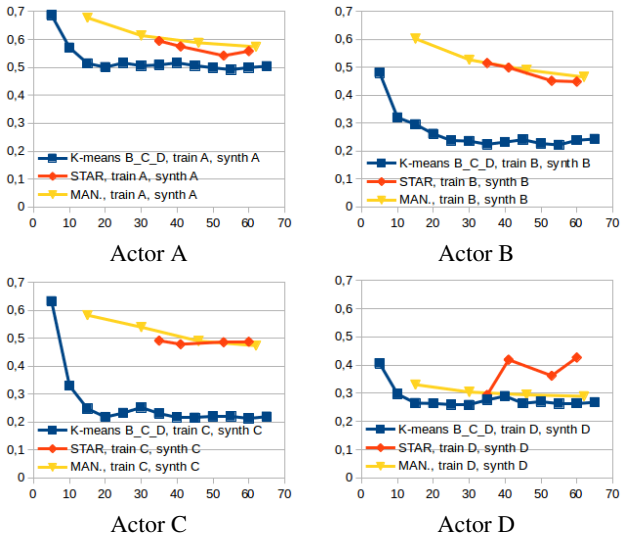


Figure 8: *Comparison of RMSE between STAR, MAN and K-means marker sets. The K-means is performed on combined* range-of-motion *sequences of the actors B, C and D.*

the blendshapes, is extremely compact. Hence, on the one hand, the mapping of marker positions to the space of blendshape coefficients is fairly trivial if we consider the abundance of training data available to us. On the other hand, the simplicity of this representation does not allow the expression of all the subtleties of human facial expression. For these reasons, it would be interesting to test this approach with other animation methods which would allow us to check the genericity of the proposed method.

The clustering approach has several limitations. Although it is independent of the synthesis method in itself, this approach remains dependent on the training data. Furthermore, being independent of the synthesis method will not give a marker set optimized for that synthesis method in particular. Finally, the K-means algorithm seeks to minimize the intra-class variance and to maximize the inter-class variance which leads to the creation of vertex groups that have a similar behavior. This means that the algorithm captures in priority the major sources of deformation leaving out the more subtle sources of deformation such as wrinkles.

Acknowledgements

This work is partly supported by the French National Research Agency (Ingredible project), and partly by a French ministry grant.

References

BEELER, T., HAHN, F., BRADLEY, D., BICKEL, B., BEARDSLEY, P., GOTSMAN, C., SUMNER, R. W., AND GROSS, M. 2011. High-quality passive facial performance capture using anchor frames. *ACM Trans. Graph.*.

BOTSCH, M., AND SORKINE, O. 2008. On linear variational surface deformation methods. *Visualization and Computer Graphics, IEEE Transactions on*.

BRADLEY, D., HEIDRICH, W., POPA, T., AND SHEFFER, A. 2010. High resolution passive facial performance capture. *ACM Trans. Graph.*.

CAO, Y., TIEN, W. C., FALOUTSOS, P., AND PIGHIN, F. 2005. Expressive speech-driven facial animation. *ACM Transactions on Graphics*.

CSATÓ, L., AND OPPER, M. 2002. Sparse on-line gaussian processes. *Neural Computation*.

DENG, Z., CHIANG, P.-Y., FOX, P., AND NEWMANN, U. 2006. Animating blendshape faces by cross-mapping motion capture data. In *Proceedings of the 2006 symposium on Interactive 3D graphics and games*.

EKMAN, P., AND FRIESEN, W. 1978. *Facial Action Coding System: A Technique for the Measurement of Facial Movement.* Consulting Psychologists Press.

2012. faceshift. http://www.faceshift.com/product/.

GIBET, S., COURTY, N., DUARTE, K., AND LE NAOUR, T. 2011. The SignCom System for Data-Driven Animation of Interactive Virtual Signers : Methodology and Evaluation. *ACM Transaction on Interactive Intelligent Systems*.

LE, B., ZHU, M., AND DENG, Z. 2013. Marker optimization for facial motion acquisition and deformation. *Visualization and Computer Graphics, IEEE Transactions on*.

LEFEBVRE-ALBARET, F., GIBET, S., TURKI, A., HAMON, L., AND BRUN, R. 2013. Overview of the Sign3D Project High-fidelity 3D recording, indexing and editing of French Sign Language content. In *Third International Symposium on Sign Language Translation and Avatar Technology (SLTAT) 2013*.

LI, H., ADAMS, B., GUIBAS, L. J., AND PAULY, M. 2009. Robust single-view geometry and motion reconstruction. *ACM Trans. Graph.*.

LI, H., WEISE, T., AND PAULY, M. 2010. Example-based facial rigging. *ACM Trans. Graph.*.

LIU, X., MAO, T., XIA, S., YU, Y., AND WANG, Z. 2008. Facial animation by optimized blendshapes from motion capture data. *Computer Animation and Virtual Worlds*.

MEYER, M., AND ANDERSON, J. 2007. Key point subspace acceleration and soft caching. *ACM Trans. Graph.*.

RASMUSSEN, C. E., AND WILLIAMS, C. K. I. 2005. *Gaussian Processes for Machine Learning*. The MIT Press.

SATTLER, M., SARLETTE, R., AND KLEIN, R. 2005. Simple and efficient compression of animation sequences. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*.

SORKINE, O., AND COHEN-OR, D. 2004. Least-squares meshes. In *Shape Modeling Applications, 2004. Proceedings*.

SOUTHERN, R., AND ZHANG, J. 2011. Motion-sensitive anchor identification of least-squares meshes from examples. *Visualization and Computer Graphics, IEEE Transactions on*.

WEISE, T., LEIBE, B., AND VAN GOOL, L. 2008. Accurate and robust registration for in-hand modeling. In *Computer Vision and Pattern Recognition.*

WEISE, T., BOUAZIZ, S., LI, H., AND PAULY, M. 2011. Realtime performance-based facial animation. *ACM Trans. Graph.*.

ZHANG, L., SNAVELY, N., CURLESS, B., AND SEITZ, S. 2008. Spacetime faces: High-resolution capture for modeling and animation. In *Data-Driven 3D Facial Animation*.