Eurographics / IEEE Symposium on Visualization 2011 (EuroVis 2011)
H. Hauser, H. Pfister, and J. J. van Wijk
(Guest Editors)

Volume 30 (2011), Number 3

Visualization of Time-Series Data in Parameter Space for Understanding Facial Dynamics

G. K. L. Tam[1], H. Fang[2], A. J. Aubrey[1], P. W. Grant[2], P. L. Rosin[1], D. Marshall[1] and M. Chen[2]

[1]School of Computer Science, Cardiff University, UK [2]Department of Computer Science, Swansea University, UK

Abstract

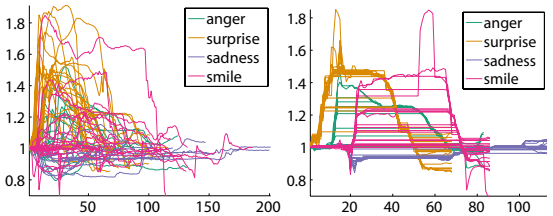
Over the past decade, computer scientists and psychologists have made great efforts to collect and analyze facial dynamics data that exhibit different expressions and emotions. Such data is commonly captured as videos and are transformed into feature-based time-series prior to any analysis. However, the analytical tasks, such as expression classification, have been hindered by the lack of understanding of the complex data space and the associated algorithm space. Conventional graph-based time-series visualization is also found inadequate to support such tasks. In this work, we adopt a visual analytics approach by visualizing the correlation between the algorithm space and our goal – classifying facial dynamics. We transform multiple feature-based time-series for each expression in measurement space to a multi-dimensional representation in parameter space. This enables us to utilize parallel coordinates visualization to gain an understanding of the algorithm space, providing a fast and cost-effective means to support the design of analytical algorithms.

1. Introduction

Facial dynamics analysis has many applications, including facial expression recognition [PP06], evaluation of trustworthiness [KMC*07], synthesis and reconstruction of facial expression [ZJZY08]. Facial dynamics data is commonly captured as video sequences. The traditional analysis pipeline is composed of (i) techniques for extracting and tracking some features across video frames, (ii) features for measuring facial dynamic changes, and (iii) algorithms for classifying measured data typically in the form of time series (commonly referred to as trajectories of features in facial dynamics analysis).

Human observers are capable of differentiating facial expressions or emotions by using a variety of visual cues [ER05]. In automatic facial dynamics analysis, one usually selects a small set of facial features and uses an analytical algorithm to discriminate one expression from another. While it has been observed that some elements of facial dynamics data may contribute to certain facial expressions, other may not [PP06]. There has not been a systematic study on the data space and algorithm space for facial dynamics analysis in the literature. Hence, researchers have limited understanding as to what is the most discriminant feature for a type of expression, or what is the most effective descriptor for measuring the dynamics of such a feature. Researchers often take an ad hoc decision to select one or a few features and try to devise an algorithm for classifying various expressions. The progress in automated facial dynamics analysis has been slow.

Analyzing time series representations of measured dynamics is a difficult challenge. One primary goal is to determine whether time series of similar expressions form a cluster, and whether such a cluster has a reasonable separation from other clusters. Conventional graph-based time series visualization has been used to present measured trajectories of features. Each trajectory typically represents the dynamics of one feature measurement M_j for one expression e_i by



(a) original time series (b) normalized time series

Figure 1: Conventional time series visualization of one measurement (mouth height) of a particular feature.

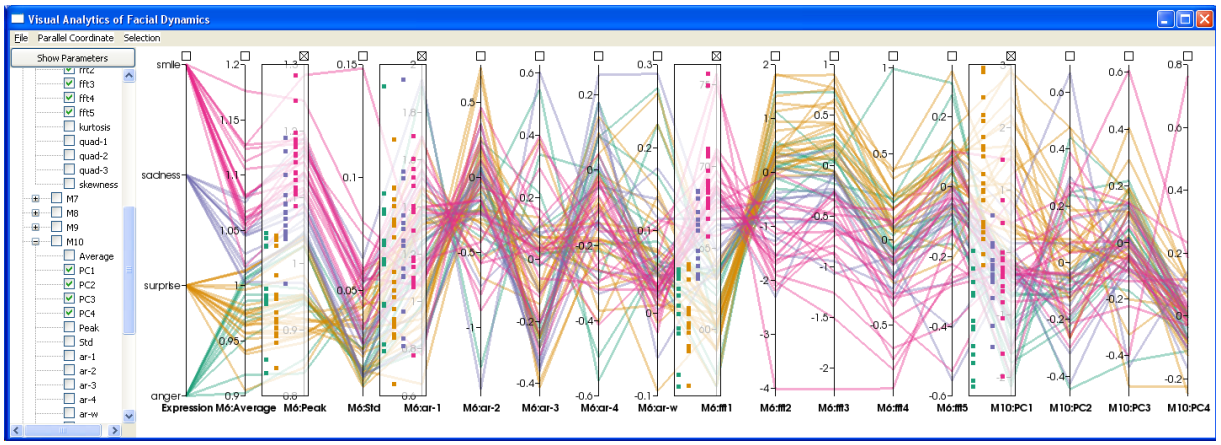


Figure 2: A parallel coordinates visualization depicting the parameter space of facial measurements in relation to 4 different types of expression. These parameters can be chosen from the left panel. Four parameters are augmented with scatter plots.

one subject s_k. One may attempt to visualize multiple trajectories in order to determine *clustering* and *separation*. For example, Figure 1a shows a set of trajectories of one feature measurement of four different facial expressions. The visualization is cluttered and incomprehensible. An alignment method, such as *Dynamic Time Warping* (DTW) [RS78], can be used to reshape such trajectories (Figure 1b). However, DTW requires a set of predefined templates, and a priori justification about clustering different trajectories around these templates. In our case, the latter should be the a posteriori knowledge that is to be obtained from the visual analysis. Applying such a template-based alignment introduces undesirable, and sometimes even flawed, bias in the analytical process. This shows the difficulties in determining clusters and separation using graph-based time series visualization.

In this work, we propose to transform the time series data to a multi-dimensional parameter space, where temporal information (e.g., shape of curve) is encoded using a variety of parameters and each parameter corresponds to a particular measurement M_j of a particular feature. As shown in Figure 2, this enables us to use parallel coordinates visualization to gain insight into the capability of each parameter in determining clusters. We augment parallel coordinates with scatter plots to support interactive design of analytical algorithms in the form of decision trees. We report our study of 22 subjects, 68 videos, 14 different feature measurements and some 23 parameters using this approach.

There are two main contributions of this work. From the perspective of facial dynamics analysis in computer vision and psychology, this work enables researchers to gain insight into the complex data and algorithm space, and incorporate domain knowledge into the algorithm design process. It complements and enhances the traditional approaches (e.g., bag of features), with which researchers typically rely on automated learning mechanisms to establish classifiers and have little knowledge of how the algorithm works on the data [CLKP10]. From the visualization perspective, we have demonstrated how to utilize a large collection of low-level analytical measurements mostly developed in computer vision to support interactive visualization and analysis of time series data in its parameter space.

2. Related Work

This section provides a brief overview of three relevant areas, namely, analysis of facial features and dynamics (Section 2.1), time-series visualization (Section 2.2) and parallel coordinates visualization (Section 2.3).

2.1. Facial Dynamics Analysis Problem

There are many methods to analyze facial dynamics. Most of them capture dynamics (time-series data) from raw videos. These dynamics are related to some appearance or geometric-based features. Geometric-based features include locations of facial points (e.g., eye corners, mouth corners) [PP06], or regions of face components [BY97]. They are useful for describing the motion of faces. Appearance-based features include the use of Gabor filter [TLJ07], Local Binary Pattern (LBP) [SGM09] and Histogram Of Gradient (HOG) [DT05]. They are good for describing texture changes (e.g., appearance of wrinkles) of a particular facial region. To align faces across video sequences and track these features over time, specific techniques like groupwise registration [CTP*10], optical flow [BY97,FCCD08] and particle filtering [PP06] have been used. To analyze the relations of these dynamic data, some methods use sets of rules to recognize facial expressions (e.g., [PP06]), whilst others train facial dynamic models like Bayesian and boosting methods (e.g., [ZJZY08]). It is interesting that many of these studies use various dynamic features, but seldom discuss the underlying rationale.

2.2. Time Series Visualization

Our input videos and facial dynamics data are multivariate time-series data. The most intuitive way to analyze time-series is to display them as trajectory plots (e.g., [TAS04, KL06,MMKN08]). Such a visual design is usually only suitable for visualizing a very small set of time series or a collection of trajectories with similar temporal patterns as illustrated in Figure 1.

By using appropriate data encoding and interaction, several tools allow users to gain more insight into time series data. "Time Searcher" [BAP*05] enables users to select interesting patterns and search for them in all multivariate time-series (query-by-example). The method is useful if a particular interesting pattern is known. "VizTree" [LKL05] transforms time series into symbolic representations [LKLC03], represents encoded data as trees and maps frequency to thickness of tree branch. The tool allows motifs and anomalies to be picked up easily within a long time-series. Kumar et al. [KLK*05] proposed to encode time-series as colour-coded 2D matrices (called "bitmaps") allowing direct visual inspection. The method also provides a similarity measure of time-series through Multi-Dimensional Scaling (MDS) that intuitively shows the clustering and comparison on a 2D screen. Recently, Meyer et al. [MWS*10] developed a tool for visualizing short time series (gene expressions) using a set of shapes in a "curvemap". Hao et al. [HMJ*11] transformed a large multivariate time-series to a one-dimensional sequence of events.

In our application, we need to study time series collected from different facial features using different measurements. The data thus exhibits a wide range of numerical and geometric variance. In addition, the data is very noisy because of errors in video capture and video processing. Hence, simple visual mapping (e.g., [KLK*05]) or direct geometry query (e.g., [BAP*05, MWS*10]) would provide limited support for our analytical tasks. An encoding strategy (e.g., [LKLC03,HMJ*11] is more promising as the measurements for feature analysis are essentially a form of encoding. However, the algorithmic space is huge, and determining an appropriate encoding scheme is in itself a challenge. Therefore, the visualization has to support the analysis of the algorithmic space for encoding time series.

2.3. Parallel Coordinates

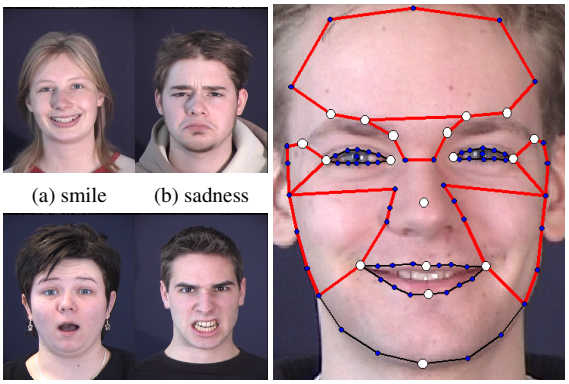
Parallel coordinates is a visualization technique that analyzes high-dimensional data [Ins09]. It arranges individual variables as parallel axes and represents individual data points (tuples) as polylines passing through all the axes. The plot is useful for identifying clusters, separability and correlation between different variables. Over the years, many improvements have been proposed. These include using cluster representations to avoid over-plotting [NH06], using densities to depict importances [HW09, FKLT10], using a stacking technique to handle overlapping cases [DWA10], adding scattering points to aid visual search [YGX*09], and optimizing the ordering of axes [DK10]. A comparison of different variants of parallel coordinate plots for cluster identification was given in [HVW10].

Parallel coordinates have been used extensively to aid scientific investigation. Notable examples include diagnosis of shoulder impairments from multi-joint kinematic data [KVDG*10]; analysis of magnetic resonance spectroscopy [FKLT10], hurricane data [HW09], fluid mixture [NH06], biomechanical motion data [KERC09]; transfer function design in volume rendering [YXG*10]; construction and exploration of decision trees [TM03]; image classification [CLKP10] and temporal trend analysis [LS09]. All these works are noteworthy, especially [TM03,CLKP10, LS09]. However, their approaches cannot be directly applied here. For example, the approach of [LS09] assumes similar trend sequences as inputs, which are not possible in our case.

3. Input Data and Initial Processing

In this section, we describe the video data and the feature measurements resulting from initial processing. We will describe the transformation of feature measurements to multi-dimensional parameter space in Section 4. There are a few expression image / video databases in the public domain (e.g., [GD]). We chose the MMI Face Database by Imperial College [PP06] because it contains more video data than others. The original database contains 1395 videos, with 197 of them labelled from the six basic facial expressions, namely smile, anger, sadness, surprise, disgust and fear. As one of our focuses is to compare the effectiveness of different features, we exclude videos where expressions are occluded by some facial attributes, such as facial hair and glasses, or profile-view videos where only half of a face is shown. We concentrate on the first 4 expressions because some studies (e.g., [JBS*09]) suggest that even humans may find it difficult to distinguish disgust and fear in some situations. This resulted in 68 videos, as exemplified in Figure 3, including 13 for anger, 22 for surprise, 14 for sadness and 19 for smile, with a total of 22 subjects. We also trimmed all videos manually so that the first (last) frame is a neutral face just before (after) the onset (offset) of any facial expressions.

In this paper, the term "feature" has the same meaning as that in everyday conversation, referring to a specific shape on or region of a face. There are two basic categories of features, namely geometric and texture features. A geometric feature is typically defined by a collection of points and/or line segments (e.g., the ellipse of mouth and eyes, curves of lips and brows). A texture feature is represented by a collection of pixels in a segmented region (e.g., the regions of forehead and cheeks). Both geometric and texture features are defined using salient points as shown in Figure 3e. These salient points and regions are defined as a template on the first frame of each video, and are tracked across time using a registration method (e.g., [FCCD08, CTP*10]).



(a) smile (b) sadness (c) surprise (d) anger (e) feature points and regions

Figure 3: Example facial expressions and features

Each feature can be analyzed by multiple measurements. For example, one can measure the height, width, and vertical displacement of a mouth feature. Similarly texture feature can also be analyzed by different filters (e.g., Gabor filter or LBP (see Section 2.1). Here, we concentrate on using a bank of 40 Gabor filters, which are capable of covering a set of 8 orientations and a range of 5 frequencies. They are good for describing wrinkles. After obtaining a set of response images from Gabor filters, we compute the maximum value for each pixel across 40 images and average these maximum values across the whole region. This reduces the dimensionality of the measurement and its sensitivity to noise.

Table 1 details all the *feature measurements* considered in this paper, where M1-M10 and M11-M14 are respectively geometric- and texture-related. For every feature, each measurement forms a time-series $m_j(t)$ corresponding to frames of the video. In summary, the data space after initial processing is a set of 14 time-series for each video $v_i \in V$:

$$< v_i,\ e_i,\ m_{i,1}(t),\ \ldots,\ m_{i,j}(t),\ \ldots,\ m_{i,14}(t) > \qquad (1)$$

where $e_i \in \{\text{smile}, \text{surprise}, \text{sadness}, \text{anger}\}$ is the known expression type, $i \in \{1...68\}$, $j \in \{1...14\}$ and $m_{i,j}(t)$ is a time-series ($1 \le t \le n_i$). Each video has a different length n_i, ranging from 24 to 137 frames.

4. Transformation to Parameter Space

Analysis of such a large data space, with time-series of various lengths, poses a difficult challenge to both facial dynamic analysis and visualization. In order to address the shortcoming of conventional time-series visualization as discussed in Section 1, we convert each $m_{i,j}(t)$ into a space of 23 parameters. Each parameter (p^k) encodes different aspects of time-series focusing on temporal characteristics. We first compute **length** (p^1) and **peak** (p^2) of all time-series. After preserving these two pieces of critical information, we linearly interpolate and normalize all $m_{i,j}(t)$ to equal length (137 frames, the overall maximum). The normalization is necessary for computing remaining parameters. Each video is summarized with the following parameter space:

$$< v_i,\ e_i,\ p^1_{i,1},\ldots,p^{23}_{i,1},\ \ \ldots,p^k_{i,j},\ldots,\ \ p^1_{i,14},\ldots,p^{23}_{i,14} > \qquad (2)$$

where $i \in \{1...68\}$, $j \in \{1...14\}$ and $k \in \{1...23\}$. In the following subsections, we detail the 23 parameters used. To simplify our notation, we represent all time-series as $\{x_1, x_2, \cdots, x_n\}$ where n is the duration and x_t is the value at time t.

4.1. Simple descriptors (p^1 – p^2)

In a diagrammatic way, each time series can be considered to be a curve, and the peak value (**peak**) of each curve is one of the best parameters to represent the overall scale of expression change. In fact, most static facial expression recognition systems extract features only from the image where a peak expression is assumed. The distance between the onset point and the offset point (**length**) encodes another overall temporal characteristic.

4.2. Low-order Moments (p^3 – p^6)

Statistical low-order moments describe important characteristics of time-series data distributions. The first four moments include central tendency (**average** μ), data variability (variance σ^2 / standard deviation (**std**) σ), curve asymmetry (**skewness** γ) and peakedness (**kurtosis** κ) [TF06]. γ and κ are defined as follows:

$$\gamma = \frac{1}{n\sigma^3}\sum_{t=1}^{n}(x_t-\mu)^3 \quad \kappa = \frac{1}{n\sigma^4}\sum_{t=1}^{n}(x_t-\mu)^4 \qquad (3)$$

4.3. PCA coefficients (p^7 – p^{10})

Principle Component Analysis (PCA) is a linear transformation which builds a new coordinate system such that the greatest variance of data lies on the first few axes. After removing the common characteristics from all time-series by

Table 1: 14 Facial feature measurements

Index	Feature measurements
M1	Chin: Vertical displacement
M2	Inner brows: Vertical displacement
M3	Outer brows: Vertical displacement
M4	Brows: Relative horizontal displacement
M5	Mouth: Height
M6	Mouth: Width
M7	Mouth corner: Vertical displacement
M8	Lower lip: Curvature
M9	Upper lip: Curvature
M10	Eye region: Size
M11	Forehead: Gabor response intensity
M12	Eye corner: Gabor response intensity
M13	Inter-brows: Gabor response intensity
M14	Cheeks: Gabor response intensity

subtracting the mean curve, the time-series are encoded using singular value decomposition (SVD). All time-series can be approximated by a few coefficients c_j [FP02]:

$$\vec{x} = \vec{x_0} + \sum_{j=1}^{n} \vec{b_j} c_j \tag{4}$$

where $\vec{x}$ is a time-series, $\vec{x_0}$ is the mean curve, $\vec{b_j}$ and c_j represents the j^{th} eigenvector and eigenvalue. The first few eigenvectors of PCA capture the main temporal characteristics, whilst the non-significant signals (possibly noise) lie on those eigenvectors having small eigenvalues. The first 4 c_j are represented as **PC1** ... **PC4** in our visualizations.

4.4. Fourier Coefficients (p^{11} – p^{15})

The Discrete Fourier Transform (DFT) [FP02]:

$$F(u) = \sum_{t=1}^{n} x_t e^{-i2\pi u(t-1)/n} \tag{5}$$

transforms time-series signals into the frequency domain, where u represents index in the frequency space. We use labels **fft1** ... **fft5** to represent $F(1) \ldots F(5)$.

4.5. Polynomial Fitting (p^{16} – p^{18})

Polynomial fitting finds a representation for a curve which describes the complete time-series data. It is assumed that the feature values are related to the time based on the following polynomial equation [FP02]:

$$x_t = a_0 + a_1 t + \ldots + a_{q-1} t^{q-1} \tag{6}$$

where $a_0, \cdots a_{q-1}$ are the parameters. A quadratic function $(q = 3)$ is used here, and a_0, a_1, a_2 are labelled as **quad-1**, **quad-2** and **quad-3**.

4.6. Auto-regressive (AR) Model (p^{19} – p^{23})

The autoregressive model is used to predict current state values based on the values at k previous states. The AR model is defined by the primitive recursion:

$$x_t = \omega + \sum_{q=1}^{L} A_q x_{t-q} + \varepsilon_q \tag{7}$$

The stepwise least squares estimation [NS01] is adopted to get the parameters due to its robustness and efficiency. We use $L = 4$ and ω, A_p are labelled as **ar-w** and **ar-1** ... **ar-4** in all figures.

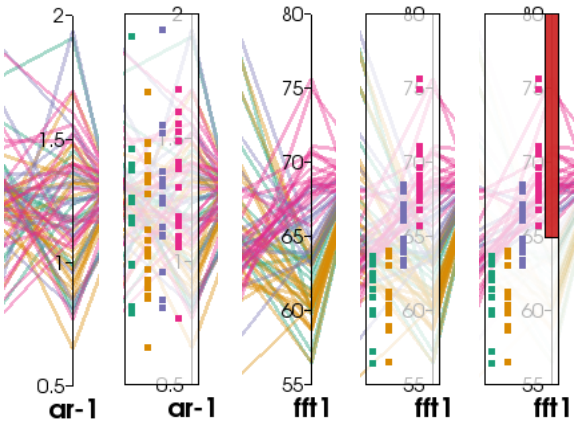
5. Interactive Visualization

The parameter space described in Section 4 consists of 14 feature measurements encoded in 23 parameters. We did not have a priori knowledge as to which feature or which parameter is effective in separating one expression from another, though we were reasonably certain that none of them was good enough on its own. Meanwhile, we had a set of labelled videos, which could serve as the classification goals, with which we would like to gain an understanding as to how these 14 feature measurements and 23 parameters are related to the goals. Furthermore, we also hoped that gaining an understanding might lead us to design a better classification algorithm than a black-box algorithm created through machine-learning.

Parallel coordinates visualization [Ins09] is known to be useful for identifying clusters, separations and outliers in high dimensional data space. The use of scatter plots in conjunction with parallel coordinates has been shown to give quick and good cluster identification [HVW10]. In this work, the focus was to study the parameter space in relation to the classification goals, rather than for identifying correlation between parameters on different axes. While scatter plots are effective in showing clusters and separations, parallel coordinates enable us to connect different parameters of each time series, which is essential for outlier identification in a complex parameter space. The translucent connection lines also add an extra visual cue about cluster formation and data distribution, in addition to that obtained from scatter plots. The most important use of parallel coordinates in this work is the interactive brushing, which dynamically informs the users about the consequence of a brushing action. Without the "flow" of redrawing the changing connection lines, it would not be easy to experiment with different configuration of a decision tree as a classification algorithm.

Consider the example in Figure 2, which shows a selection of parameters for feature measurement M6 (mouth width) and M10 (eye size). We can easily verify that none of the selected parameters alone is able to provide a clean separation of any of the expressions, and there are many outliers. Further inspection shows that some parameter spaces (e.g., **fft1** and **fft2** of M6, Figure 4c) seem to provide good clusters of expressions but others do not (e.g., **ar-1**, Figure 4a). To verify the separability, users can switch on the scatter plot on a particular axis (Figure 4b and 4d) for a detailed inspection, with the groups of expressions plotted on the x-axis.

Data are colored according to different expression groups to highlight clusters and separations, with color scheme chosen from the online tool ColorBrewer [HB03]. Interactively, users can specify the opacity of lines or adjust the scales of axis to zoom in a particular region. For this work, the groups of expressions are always plotted on the leftmost axis of parallel coordinates, all lines are drawn with opacity in the range 100-120, and all axes are in linear scales and zoomed to show the maximum and minimum values. To browse through different feature measurements, even with limited screen space, users can select different parameters (individually or as a group) on the left panel for visualization (Figure 2). A brushing tool is provided to select different data points (Figure 4e, the red bar). We also implemented a



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

Figure 4: Identifying (a) (b) dispersions and (c) (d) clusters in parallel coordinate and scatter plot visualization, and (e) the use of brushing tool.

load and save mechanism to store a user's selection, which is useful for building decision trees. All these are implemented using the Visualization Toolkit [SML03]. In this work, reordering of parallel coordinates axes is possible, but is not necessary because users are interested mainly in the correlation between each parameter axis with the known classification (i.e., the leftmost axis), and the capability of each parameter in differentiating classes (i.e., scatter plot).

To build a decision tree, we browse through all 14 feature measurements and scan through the axes of parallel coordinate plots to look for clusters, non-clusters, outliers and separations. If a strong cluster appears, these transparent lines will come close and overlap, leading to a more opaque colour. If the parameters vary a lot, lines will scatter along the axis with less opaque colour. If a combination of feature measurements and parameters gives a clear separation or forms a strong cluster that overlap relatively little with other expressions, they are good candidates to discriminate a particular expression. To test whether the parameters are suitable, we apply brushing to select data points on an axis with some thresholds. This is equivalent to making a potential decision (a node) in the decision tree, and the separability can be immediately observed. If a decision is confirmed, the selected and non-selected data are saved and carried forward to the right and left branch of a decision tree. The save reload mechanism allows the user to focus on a particular branch, without caring about other branches, allowing a multi-level analysis of the decision tree.

6. Results

In this section, we first detail the observations and insights from analyzing feature measurements, namely 1) clusters / non-clusters, 2) outliers (anomalies) and 3) separations. Then we discuss building a decision tree for classification.

6.1. Clusters and Non-Clusters

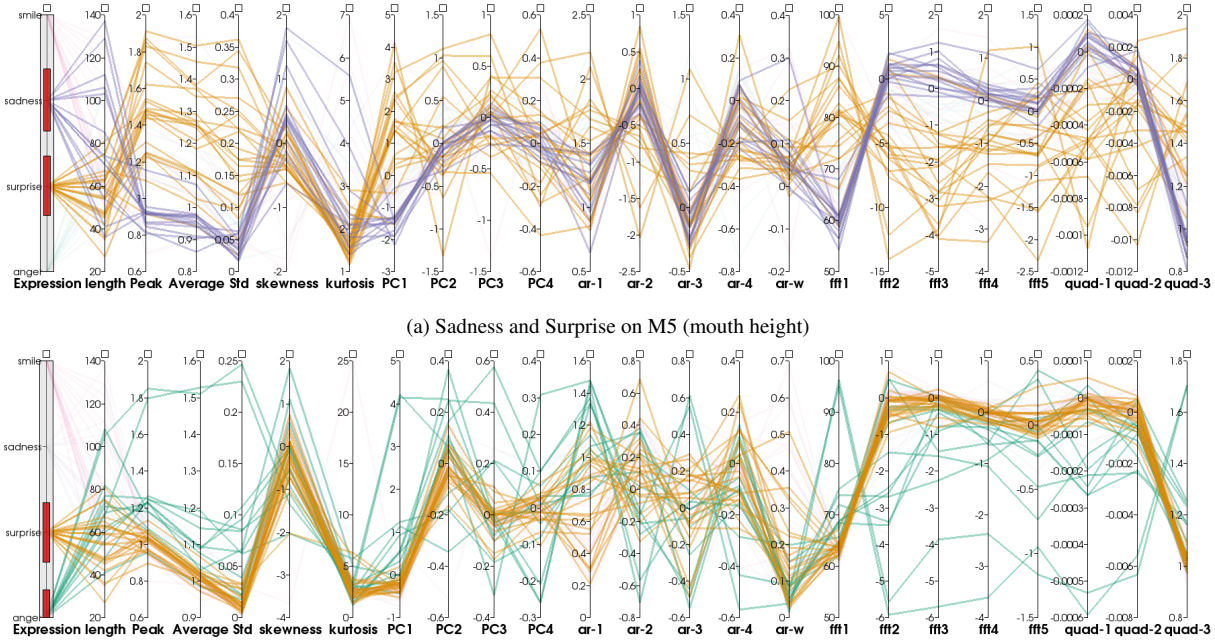
Clusters and non-clusters are equally important to understand facial dynamics. A cluster in some parameter spaces suggest that these data have similar temporal characteristics. One typical example is that some expressions do not activate certain parts of the face and these parameter spaces of the associated feature measurement show stable values (e.g., 0). Non-clustering suggests that there are many measured values in some parameter spaces. As these time-series are not aligned, those values will fluctuate across different subjects' videos leading to a dispersion (non-clustering) in the plot. Some examples are shown in Figure 5. In Figure 5a, the expressions of surprise show large variations while sadness shows little variation in M5 (mouth height). In Figure 5b, anger shows larger variations in M13 (inner brow texture intensity) whilst most surprise expressions cluster at some values 0 (or 1 in simple descriptors) in corresponding axes indicating that there is little movement. Due to limited space, we summarize our observations in Table 2, using two levels of dispersion: large and small, which are manually defined. They indicate whether there are large or small variations on more than 75% of all parameters (some parameters (e.g., **length**, **skewness**, **kurtosis**) are observed to be less reliable.). Small* further indicates the dispersion is smallest among all measurements. **peak**, **average**, **std**, coefficients of PCA, Fourier and quadratic fitting are observed to give better understanding of facial dynamics.

6.2. Outliers and anomalies

Parallel coordinates facilitate the identification of outliers or anomalies. In particular, we observe two videos that are outliers of their expression groups. Figure 7a shows that several parameters of a smile expression (code number: 1905) are consistently different from other smile expressions in terms of M2, M3, M11 and M14 dynamics. Watching the video sequence we find that the subject is simultaneously perform-

Table 2: Visual observation of the levels of variations in the measurement of the 14 facial features.

variations	surprise	smile	sadness	anger
M1	large	small	small	small
M2	large	small	small	small
M3	large	small	small	small
M4	small	small	large	large
M5	large	small	small*	small
M6	large	large	small	large
M7	large	large	small	small
M8	large	large	large	small
M9	large	large	small	small
M10	large	small	small	small
M11	large	small	small	small
M12	small	small	small	small
M13	small*	small*	small*	large
M14	small	large	small	small



(a) Sadness and Surprise on M5 (mouth height)

(b) Anger and Surprise on M13 (inner brow texture intensity)

Figure 5: Identifying clusters and non-clusters with parallel coordinate and scatter plot visualization

ing a mix of smile and surprise (a cartoon-ish smile) with lots of head movement (Figure 6a). M2, M3 and M11 are descriptive features for surprise, and M14 for smile (see Section 6.1). Figure 6b shows a surprise expression having the most similar M2 feature to this smile outlier. In Figure 7b, several parameters (M2, M3, M4) of an anger expression (code number: 1931) are observed to be outliers as well. From our perspective the video shows a subject performing a sad expression (Figure 6c) (this is labelled in the database as anger), which is very similar to another sadness video of the same subject (code name: 1943, Figure 6d). The close relation of the two videos is observed in our visualization.

There are other expression outliers but we discuss these two because they are outliers in terms of the largest number of feature measurements. Since the videos capture facial expressions from real people, and there was no report of mislabelling in the original paper [PP06], we include these two expressions in building our decision tree in Section 6.4.

(a) 1905 (b) 1807 (c) 1931 (d) 1943

Figure 6: From Figure 7, smile 1905 in (a) and anger 1931 in (c) are identified as outliers of their groups. The corresponding video frames reveal their similarity to surprise 1807 in (b) and sadness 1943 in (d) respectively.

6.3. Separation

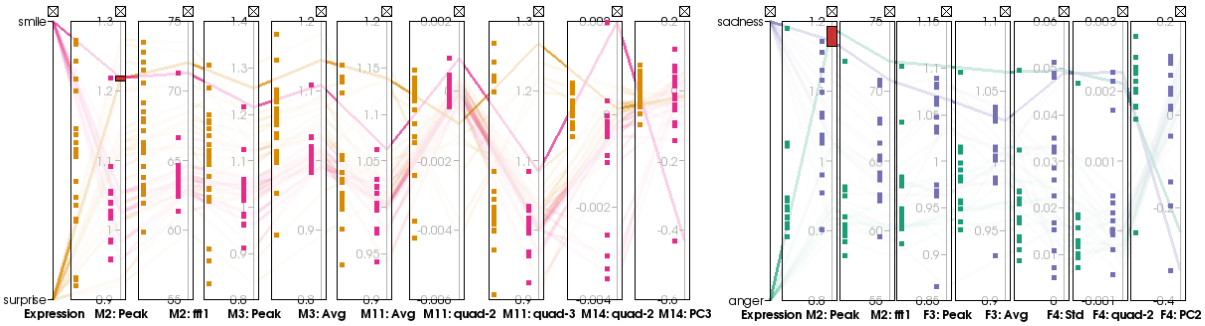
Parallel coordinates, together with scatter plots, can help users to identify separation of data easily. We summarize a few observations below:

M6 (mouth width) Smile is consistently separable from surprise and anger expressions (Figure 2 **peak**, **fft1**, **fft2** and Figure 8a). The reason is that smile expressions usually lengthen mouth width, whilst surprise and anger shorten mouth width.

M10 (eye size) Larger eye size is a consistent feature of surprise expressions (Figure 8b and 8c). Though it cannot give a clear separation of surprise from other expressions, it is an important feature.

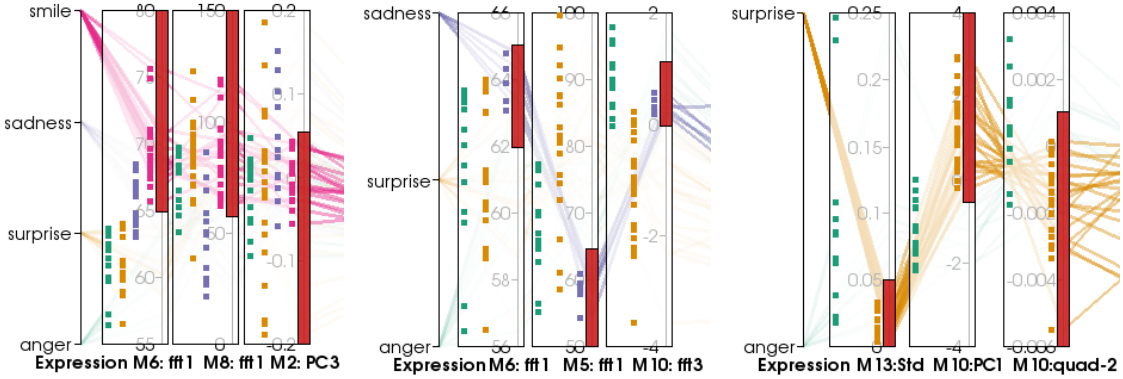
6.4. Decision Tree

By using the above observations, a decision tree can be built as follows. We first see that M6 (mouth width) is a good measurement to start with because it gives a clear separation of smile from surprise or anger expressions (Figure 8a). In particular, we pick **fft1** because the parameter provides a clearer separation. Setting **fft1** > 65 splits the decision tree into two main branches. One consists of smile and sadness expressions (top branch), whilst another consists of sadness, surprise and anger expressions (bottom branch) (Figure 9,



(a) Smile 1905 (outlier), Surprise 1807 (b) Anger 1931 (outlier), Sadness 1943

Figure 7: Parallel coordinates help identify outliers in time series classification. In (a) Smile 1905 is an outlier. A time-series in a different class, Surprise 1807, exhibits a similar set of parameters. In (b), Anger 1931 is an outlier, which is very similar to Sadness 1943 in parameter space.



(a) Separate smile from all expressions (b) Identifying sadness at Decision 3 (c) Identifying surprise at Decision 4

Figure 8: Decision making with parallel coordinates and brushes

Decision 1). Next, we observe that M8 (Lower Lip Curvature) is a good indicator for separating smiles and sadness. Though we cannot find a clean separation between the two, we pick M8 : **fft1** to separate all smiles from some sadness expressions. Smile is also shown to be stable (not much movement) on inner brow measurement (M2 : **PC3**). After picking these selections, all smiles are able to be separated from all other expressions. Since smile expressions are isolated, the remaining sadness on the top main branch can be found (Figure 8a, Figure 9 Decision 2).

On the bottom main branch of the decision tree, we observe that some sadness expressions form strong clusters in M6 (mouth width) **fft1**, M5 (mouth height) **fft1**, and M10 (eye size) **fft3**, because sadness expressions show little variations or movements on these features. By picking them all, we can quickly separate the remaining sadness from surprise and anger expressions (Figure 8b, Figure 9 Decision 3). To separate surprise from anger expressions, we pick M13 (wrinkle intensity of inter-brows) and **Std** (standard deviation). This is based on the observation that surprise expressions form a strong cluster (surprise have little movement

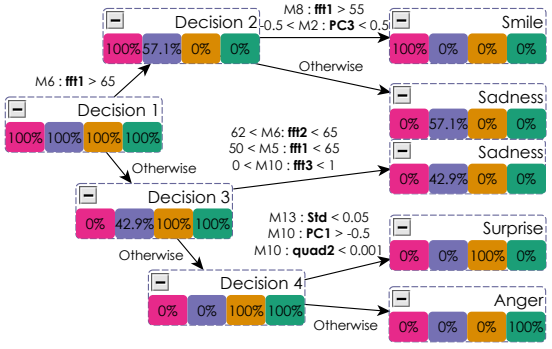


Figure 9: Decision tree generated from visualization

between inter-brows; whilst anger expressions show lots of movements) in this region. To further refine the decision tree, we observe that M10 (eye size) **PC1**, **quad-2** again provides some separation between surprise and anger expressions. By defining the selection ranges that allow some errors, all surprise can be separated from anger expressions (Figure 8c, Figure 9 Decision 4).

The way we build the decision tree is not unique. One can always pick some other combinations of feature measurements and parameter spaces to build a new one. The above steps try to show that it is possible to build such a decision tree easily by 1) browsing through all these features and parameters quickly with our tool to identify important ones, 2) making sure they are reliable and sensible before use (semantics), and 3) ensuring some form of clusters and separations.

6.5. Comparison and Discussion

Using the same data in parameter space, we compared our approach with a well-known and publicly available decision tree building tool C4.5 [Qui93]. Quantitatively, the training accuracy of the C4.5-decision tree is 98.5% with one anger expression misclassified as surprise (Figure 10, Decision 2), while our interactively constructed decision tree achieved 100% training accuracy. Because the data set is small, we treat this comparison with caution. Qualitatively, we also observe that in the automatically generated tree, a) very tight decision boundaries are specified, and b) no range constraints are used. This is likely due to the fact that the automated algorithm is coded to seek the best parameter with highest information gain. With a visual analytics approach, one can adjust selection criteria dynamically. For example, in constructing the decision tree in Figure 9, we maintained larger margins for decision boundaries and used range constraints as long as we felt safe to do so. We also used visualization to observe the robustness of features (e.g., Figure 9 Decision 3: M5, M6, M10 are largely stationary for sadness expression). Such dynamic reasoning with additional criteria is not available in C4.5. In addition, our findings in Sections 6.1 and 6.3, and thus the parameters used in our decision tree, correspond well with some psychology studies [NCWB08].

Our experience shows that it is important to have a balance between automated analysis and interactive visualization in dealing with large data sets and complex analytical tasks. On the one hand, using solely interactive visualization does not scale in relation to very large parameter space. On the other hand, using solely automated analysis prohibits the users from incorporating new semantic understanding and improvisational reasoning strategies dynamically. The development of automated solutions is often costly due to its problem-specific nature and requirements for large training datasets. We consider that our facial dynamics analysis has struck a good balance in scalability. We used automation to select a set of good parameter candidates, for example, the first three PCA coefficients (instead of many). We relied on interactive visualization to make higher-level analysis and selection decisions. Should the data or algorithm space become much larger, one may have to seek more help from the automated aspect of the analytic process.

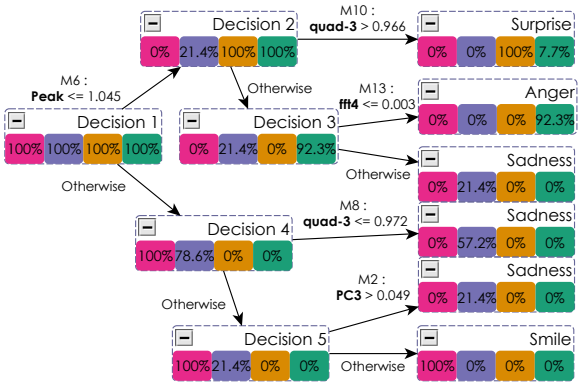


Figure 10: Decision tree obtained from C4.5 algorithm

7. Conclusion and Future Work

We have presented our experience of using interactive visualization to gain an understanding of facial dynamics data and to derive a decision tree as a classification algorithm. The transformation of a large collection of time series to parameter space using a set of low-level computer vision measurements facilitates the visualization of the temporal characteristics of time series. The interactive visualization, using parallel coordinates and scatter plots, in return provides a meaningful view of the algorithm space for data classification, and guides users in selecting components and determining their ordering in a decision tree. Throughout the process, we 1) observe patterns of data in parameter space rather than speculating ad hoc hypotheses; 2) identify anomalies (outliers); 3) reason with the measured data and algorithmic space as well as the videos; and 4) construct an algorithm that we understand rather than be given as a black-box.

We are in the process of collecting a new data set, which may benefit more from the newly found capability of visual analytics. In addition, we are working with a machine-learning team to investigate the feasibility of integrating an automated machine-learning mechanism into our approach.

References

[BAP*05] BUONO P., ARIS A., PLAISANT C., KHELLA A., SHNEIDERMAN B.: Interactive pattern search in time series. In *Proc. Conf. on Vis and Data Ana.* (2005). 3

[BY97] BLACK M. J., YACOOB Y.: Recognizing facial expressions in image sequences using local parameterized models of image motion. *Int. J. Comp. Vis. 25*, 1 (1997), 23–48. 2

[CLKP10] CHOO J., LEE H., KIHM J., PARK H.: ivisclassifier: An interactive visual analytics system for classification based on supervised dimension reduction. In *Proc. IEEE Symp. on Vis. Anal. Sci. and Tech.* (2010), pp. 27–34. 2, 3

[CTP*10] COOTES T., TWINING C., PETROVIC V., BABALOLA K., TAYLOR C.: Computing accurate correspondences across groups of images. *IEEE Trans. Pat. Anal. & Mach. Int. 32* (2010). 2, 3

[DK10] DASGUPTA A., KOSARA R.: Pargnostics: Screen-space metrics for parallel coordinates. *IEEE Trans. Vis. & Comp. Graphics 16* (2010), 1017–1026. 3

[DT05] DALAL N., TRIGGS B.: Histograms of oriented gradients for human detection. In *Proc. IEEE Conf. on Comp. Vis. and Pat. Rec.* (2005), vol. 1, pp. 886–893. 2

[DWA10] DANG T. N., WILKINSON L., ANAND A.: Stacking graphic elements to avoid over-plotting. *IEEE Trans. Vis. & Comp. Graphics 16* (2010), 1044–1052. 3

[ER05] EKMAN P., ROSENBERG E. L.: *What the face reveals: Basic and applied studies of spontaneous expression using the Facial Action Coding System*. Oxford University Press, 2005. 1

[FCCD08] FANG H., COSTEN N., CRISTINACCE D., DARBY J.: 3d facial geometry recovery via group-wise optical flow. In *Proc. IEEE Conf. Auto. Face and Gest. Rec.* (2008), pp. 1–6. 2, 3

[FKLT10] FENG D., KWOCK L., LEE Y., TAYLOR R.: Matching visual saliency to confidence in plots of uncertain data. *IEEE Trans. Vis. & Comp. Graphics 16* (2010), 980–989. 3

[FP02] FORSYTH D. A., PONCE J.: *Computer Vision: A Modern Approach*. Prentice Hall, 2002. 5

[GD] GRGIC M., DELAC K.: Face recognition homepage. http://www.face-rec.org/databases/. 3

[HB03] HARROWER M. A., BREWER C. A.: Colorbrewer.org: An online tool for selecting color schemes for maps. *The Cartographic Journal 40*, 1 (2003), 27–37. 5

[HMJ*11] HAO M., MARWAH M., JANETZKO H., SHARMA R., KEIM D. A., DAYAL U., PATNAIK D., RAMAKRISHNA N.: Visualizing frequent patterns in large multivariate time series. In *Proc. Conf. on Vis and Data Ana.* (2011). 3

[HVW10] HOLTEN D., VAN WIJK J. J.: Evaluation of cluster identification performance for different PCP variants. *Computer Graphics Forum 29*, 3 (2010), 793–802. 3, 5

[HW09] HEINRICH J., WEISKOPF D.: Continuous parallel coordinates. *IEEE Trans. Vis. & Comp. Graphics 15* (2009), 1531–1538. 3

[Ins09] INSELBERG A.: *Parallel Coordinates: Visual Multidimensional Geometry and Its Applications*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2009. 3, 5

[JBS*09] JACK R. E., BLAIS C., SCHEEPERS C., SCHYNS P. G., CALDARA R.: Cultural confusions show that facial expressions are not universal. *Journal Current biology 19*, 18 (2009), 1543–1548. 3

[KERC09] KEEFE D., EWERT M., RIBARSKY W., CHANG R.: Interactive coordinated multiple-view visualization of biomechanical motion data. *IEEE Trans. Vis. & Comp. Graphics 15*, 6 (2009), 1383–1390. 3

[KL06] KINCAID R., LAM H.: Line graph explorer: scalable display of line graphs using focus+context. In *Proc. ACM Conf. on Advanced visual interfaces* (2006), pp. 404–411. 3

[KLK*05] KUMAR N., LOLLA N., KEOGH E., LONARDI S., RATANAMAHATANA C. A.: Time-series bitmaps: a practical visualization tool for working with large time series databases. In *Proc. SIAM Data Mining Conf.* (2005), pp. 531–535. 3

[KMC*07] KRUMHUBER E., MANSTEAD A. S. R., COSKER D., MARSHALL D., ROSIN P. L., KAPPAS A.: Facial dynamics as indicators of trustworthiness and cooperative behavior. *Emotion 7*, 4 (2007), 730–735. 1

[KVDG*10] KREKEL P. R., VALSTAR E. R., DE GROOT J., POST F. H., NELISSEN R. G. H. H., BOTHA C. P.: Visual analysis of multi-joint kinematic data. *Computer Graphics Forum 29*, 3 (2010), 1123–1132. 3

[LKL05] LIN J., KEOGH E., LONARDI S.: Visualizing and discovering non-trivial patterns in large time series databases. *Information Visualization 4*, 2 (2005), 61–82. 3

[LKLC03] LIN J., KEOGH E., LONARDI S., CHIU B.: A symbolic representation of time series, with implications for streaming algorithms. In *Proc. ACM SIGMOD workshop on Research issues in data mining and knowledge discovery* (2003), pp. 2–11. 3

[LS09] LEE T.-Y., SHEN H.-W.: Visualization and exploration of temporal trend relationships in multivariate time-varying data. *IEEE Trans. Vis. & Comp. Graphics 15* (2009), 1359–1366. 3

[MMKN08] MCLACHLAN P., MUNZNER T., KOUTSOFIOS E., NORTH S.: Liverac - interactive visual exploration of system management time-series data. In *Proc. SIGCHI Conf. on Human Factors in Computing Systems* (2008), pp. 1483–1492. 3

[MWS*10] MEYER M., WONG B., STYCZYNSKI M., MUNZNER T., PFISTER H.: Pathline: A tool for comparative functional genomics. *Computer Graphics Forum 29*, 3 (2010), 1043–1052. 3

[NCWB08] NUSSECK M., CUNNINGHAM D. W., WALLRAVEN C., BÜLTHOFF H. H.: The contribution of different facial regions to the recognition of conversational expressions. *Journal of Vision 8*, 8:1 (2008), 1–23. 9

[NH06] NOVOTNY M., HAUSER H.: Outlier-preserving focus+context visualization in parallel coordinates. *IEEE Trans. Vis. & Comp. Graphics 12* (2006), 893–900. 3

[NS01] NEUMAIER A., SCHNEIDER T.: Estimation of parameters and eigenmodes of multivariate autoregressive models. *ACM Trans. Math. Softw. 27*, 1 (2001), 27–57. 5

[PP06] PANTIC M., PATRAS I.: Dynamics of facial expression: recognition of facial actions and their temporal segments from face profile image sequences. *IEEE Trans. Syst. Man Cybern. B-Cybern. 36*, 2 (2006), 433–449. 1, 2, 3, 7

[Qui93] QUINLAN J. R.: *C4.5: programs for machine learning*. Morgan Kaufmann Publishers Inc., San Francisco, USA, 1993. 9

[RS78] RABINER L. R., SCHAFER R. W.: *Digital Processing of Speech Signals*. Pearson Education, 1978. 2

[SGM09] SHAN C., GONG S., MCOWAN P.: Facial expression recognition based on local binary patterns: A comprehensive study. *Image and Vision Computing 27* (2009), 803–816. 2

[SML03] SCHROEDER W., MARTIN K., LORENSEN B.: *The Visualization Toolkit*, 3rd edition ed. Kitware, Inc., 2003. 6

[TAS04] TOMINSKI C., ABELLO J., SCHUMANN H.: Axes-based visualizations with radial layouts. In *Proc. ACM Symp. on Applied Computing* (2004), pp. 1242–1247. 3

[TF06] TABACHNICK B. G., FIDELL L. S.: *Using Multivariate Statistics (5th Edition)*. Allyn & Bacon, Inc., Needham Heights, MA, USA, 2006. 4

[TLJ07] TONG Y., LIAO W., JI Q.: Facial action unit recognition by exploiting their dynamic and semantic relationships. *IEEE Trans. Pat. Anal. & Mach. Int. 29*, 10 (2007), 1683–1699. 2

[TM03] TEOH S. T., MA K.-L.: Paintingclass: interactive construction, visualization and exploration of decision trees. In *Proc. ACM SIGKDD* (2003), KDD '03, pp. 667–672. 3

[YGX*09] YUAN X., GUO P., XIAO H., ZHOU H., QU H.: Scattering points in parallel coordinates. *IEEE Trans. Vis. & Comp. Graphics 15* (2009), 1001–1008. 3

[YXG*10] YUAN X., XIAO H., GUO H., GUO P., KENDALL W., HUANG J., ZHANG Y.: Scalable multi-variate analytics of seismic and satellite-based observational data. *IEEE Trans. Vis. & Comp. Graphics 16* (2010), 1413–1420. 3

[ZJZY08] ZHANG Y., JI Q., ZHU Z., YI B.: Dynamic facial expression analysis and synthesis with mpeg-4 facial animation parameters. *IEEE Trans. Circuits Syst. Video Tech. 18*, 10 (2008), 1383–1396. 1, 2