# Fex-Metrica: A Toolbox for the Analysis of Facial Expression Time Series

F. Rossi[1,*], L.J. Chang[2], I.R. Fasel[3], C.T. Morrison[4], A.G. Sanfey[5], and M.S. Bartlett[1,3]

[1]Institute for Neural Computation, University of California, San Diego, La Jolla, CA
[2]Computational Social Affective Neuroscience Lab, Dartmouth College, Hanover, NH
[3]Emotient Inc., San Diego, CA
[4]School of Information, Science, Technology and Art, University of Arizona, Tucson, AZ
[5]Donders Center for Brain, Cognition and Behavior, Radboud University, Nijmegen, NL
[*]Contact Information: `frossi@ucsd.edu`

March 31, 2015

**Abstract**

One of the fast-growing fields of cognitive neuroscience is the study of emotions and their effects on cognitive processes []. A very promising approach, which allows unique insights into human emotions is the observation of humans' facial expressions []. Studies have shown that facial expressions are systematically associated with emotions such as anger, joy or fear [], therefore tracking facial movements allows to study underlying emotional experiences. Facial expressions tracking used to be labor-intense. However, advances in computer vision have rendered automatic facial expression detection fast and reliable []. In this work we introduce *FexMetrica*, a toolbox developed to conduct statistical analyses of automatically detected facial expression time series []. As with most time series data, there are nuisances due to the nature of the signal, which need to be addressed before moving to statistical inference. Our toolbox is meant to provide a basic set of algorithms for these analyses, and the scaffold for the collaborative development of procedures for the study of facial expressions.

# 1 Introduction

## 1.1 Automatic Detection of Facial Expressions

There are two main frameworks for the study of emotions. One of them operationalizes emotions in terms of a three-dimensional space with axes for valence, salience and arousal []. Alternatively, the Facial Action Coding System (FACS) [] is more quantize and it assumes the existence of seven primary emotions: anger, contempt, disgust, fear, joy, sadness and surprise []. Our research uses this second framework, which relies on Paul Ekman's seminal work on the relationship between facial muscles and emotions []. In particular, muscular activity in the face is described in terms of Action Units (AU), such as frowning of the eyebrows (AU 4) or raising the cheeks (AU 6) []. Emotions are then described as combinations of these AUs [] (see **Fig. 1**).

Human-based assessment of emotions is extremely time-demanding []. Facial expressions would not be a particularly serviceable signal if it took hours to identify emotions over one single minute of video []. Fortunately, machine learning and computer provide automatic and reliable readings of facial expressions from video recordings or even real-time video capturing [].

We have been contributing to this endeavor by developing one very popular system: the Computer Expression Recognition Toolbox (CERT) []. This system finds a face in a square patch of pixels from a video frame and it detects both AUs and emotions from the detected face []. Notably, CERT was successfully used to study emotions in a broad set of scenario such as pain and clinical studies [], driving simulation [], and even neuroeconomic studies [].

In the past few years, CERT has evolved in even more reliable set of machine learning tools provided by Emotient Analytics. This new system is spreading across academics (approximately 150 as of late 2014), and it guarantees reliable readings of emotions from facial expressions. For comparison, **Fig. 2** shows CERT and Emotient Analytics on a large database of posed facial expressions (N $>$ 100,000). The new system significantly outperforms CERT, which in itself is considered state-of-the-art in computer vision []. Based on these results, we decided to use Emotient Analytics as starting point for *FexMetrica*.

The development of accurate facial expression detectors is an ongoing process. Parallel to this process, we also have to design adequate algorithms that account for the nuisances of facial expressions time series []. The purpose of our toolbox is to provide an initial set of "canonical" tools for the analysis of facial expressions time series. Additionally, we hope that *FexMetrica* will constitute a shared platform for the development of appropriate techniques for mining facial expression data.

### 1.2 Dataset Description

We generated a database of videos with posed expressions of emotions occurring at different temporal frequencies, in order to exemplify the key aspects of the analysis of facial expression time series. The dataset include 2.5 minute video from 10 participants. During the "experiment," we asked participants to imitate specific expressions during the "Production" window for 20 trials (see **Fig. 3**). We prompted only four facial expressions: joy, surprise, disgust, and contempt. Each trial lasts for 10 seconds, and expressions are produced for either one or two seconds.

There are three main parameters manipulated in the video:

1. Emotion Produced;
2. Duration;
3. Onset;

## 2 Preprocessing Facial Expression Time Series

### 2.1 Facial Expressions Time Series Objects

Generate a **fexc** object ...

```
fex_init;
fexobj = fexc('ui');
```

### 2.2 "Spatial" Processing

Look for "false positive," namely patches of pixels that the system evaluate as containing a face, when they are not.

```
fexobj.falsepositive('method','position','threshold',2);
fexobj.coregister('fp',true,'threshold',2);
fexobj.motioncorrect('thrs',0.50,'normalize','-whiten');
```

### 2.3 Temporal Processing

### 2.4 Normalization

```
fexobj.setbaseline('mean','-global');
fexobj.rectification(-1);
```

# 3 Tools for Statistica Analyses and Visualization

## 3.1 “Design” Objects

## 3.2 Features Extraction and Generation

## 3.3 Regression and Classification

## 3.4 Simple Models of Multi-Modal Integration

## 3.5 Visualization Tools

# 4 Conclusions

```
for i = 1:3
    if i ≥ 5 && a ≠ b       % literate programming replacement
        disp('cool');           % comment with some LaTeXin it:  πx²
    end
    [:,ind] = max(vec);
    x_last = x(1,end) - 1;
    v(end);
    really really long really really long really really long ...
        really really long really really long line % blaaaaaaaa
    ylabel('Voltage (μV)');
end
```