PhD Dissertation

International Doctorate School in Information and Communication Technologies

DISI - University of Trento

Discrimination of Computer Generated versus Natural Human Faces

Duc-Tien Dang-Nguyen

Advisor:

Prof. Giulia Boato

Università degli Studi di Trento

Co-Advisor:

Prof. Francesco G. B. De Natale

Università degli Studi di Trento

February 2014

"In the temple of his spirit, each man is alone."

- Ayn Rand -

Abstract

The development of computer graphics technologies has been bringing realism to computer generated multimedia data, e.g., scenes, human characters and other objects, making them achieve a very high quality level. However, these synthetic objects may be used to create situations which may not be present in real world, hence raising the demand of having advance tools for differentiating between real and artificial data. Indeed, since 2005 the research community on multimedia forensics has started to develop methods to identify computer generated multimedia data, focusing mainly on images. However, most of them do not achieved very good performances on the problem of identifying CG characters.

The objective of this doctoral study is to develop efficient techniques to distinguish between computer generated and natural human faces. We focused our study on geometric-based forensic techniques, which exploit the structure of the face and its shape, proposing methods both for image and video forensics. Firstly, we proposed a method to differentiate between computer generated and photographic human faces in photos. Based on the estimation of the face asymmetry, a given photo is classified as computer generated or not. Secondly, we introduced a method to distinguish between computer generated and natural faces based on facial expressions analysis. In particular, small variations of the facial shape models corresponding to the same expression are used as evidence of synthetic characters. Finally, by exploiting the differences between face models over time, we can identify synthetic animations since their models are usually recreated or performed in patterns, comparing to the models of natural animations.

Keywords:

[Computer Generated versus Natural Data Discrimination, Digital Image Forensics, Digital Video Forensics, Facial Animations Analysis]

Contents

List of Tables

List of Figures

Chapter 1

Introduction

This chapter overviews the research field investigated in this doctoral study. In particular, we describe computer generated versus natural multimedia data discrimination techniques, focusing on human faces. The main objectives and the novel contributions of this thesis are also presented. Finally, we describe the organization of this document.

"A journey of a thousand miles must begin with a single step"
Lao Tzu

People have been attempting to represent the real world since ancient times. A version of an oft-told Greek story in around 450BC concerns two painters Parrhasius and Zeuxis. Parrhasius asked Zeuxis to judge one of his paintings that was behind a pair of curtains. Zeuxis was asked to pulled back the curtains, but when he tried, he could not, as the curtains were Parrhasius's painting. That was one of the first stories of *Trompe loeil*, literally means 'deceiving the eye' or often called 'trick of the eye', an art technique that uses realistic imagery to create the optical illusion that depicted objects. For example, a painting by Jacopo de' Barbari in 1504

of a partridge, gauntlets, and crossbow bolt (see Figure 1.1(a)), which is considered as the first small scale *Trompe l'oeil* painting since antiquity. Another example is shown in Figure 1.1(b) from a painting of Henry Fuseli (1750).

(a) Jacopo de' Barbari, 1504. (b) *Trompe loeil* by Henry Fuseli, 1750.

Figure 1.1: Examples of *Trompe loeil* paintings.

In modern day, *Trompe loeil* artists create their art by combining traditional techniques with the modern technologies to create more types of illusions. For example, while the house on 39 George V street, Paris was being renovated, they printed and hung an interesting artwork on the scaffolding to shelter the rehabilitation, which is shown in Figure 1.2(a). Another modern *Trompe loeil* can be seen in Figure 1.2(b), created by Pierre Delavie on facade of the Palais de la Bourse, Marseille, which shows the Canebière - the historic high street in the old quarter of Marseille.

(a) The 39GeorgeV building in Paris.

(b) Modern *Trompe loeil* on facade of the Palais de la Bourse, Marseille.

Figure 1.2: Examples of modern *Trompe loeil* paintings.

Trompe loeil not only makes a painting more realistic, but also exploits the techniques that can attack the weaknesses of human visual system, which can be applied to digital image forensics. Using modern computer graphics technologies, synthetic scenes, human characters or objects can be easily created with a very high quality level, which could take years to artists in classic *Trompe loeil*. Some examples of computer graphics images are shown in Figure 1.3, in which most of the images are very realistic.

However, these synthetic objects may be used to create situations which may not be present in real world, and hence raising security risks. For example in the US, possession of child pornography is illegal since it implies abuse of minors. However, establishing the presence of minors from the child pornography is challenging on legal ground, as owners of child pornography can declare the images to be computer generated [36]. This raise a need of tools able to automatically and reliably discriminate between CG and natural images in this particular case, and in multimedia data in general. Hence, many techniques have been proposed to deal with this problem. A big picture is shown in the next section while the literature will be detailed in Chapter 2.

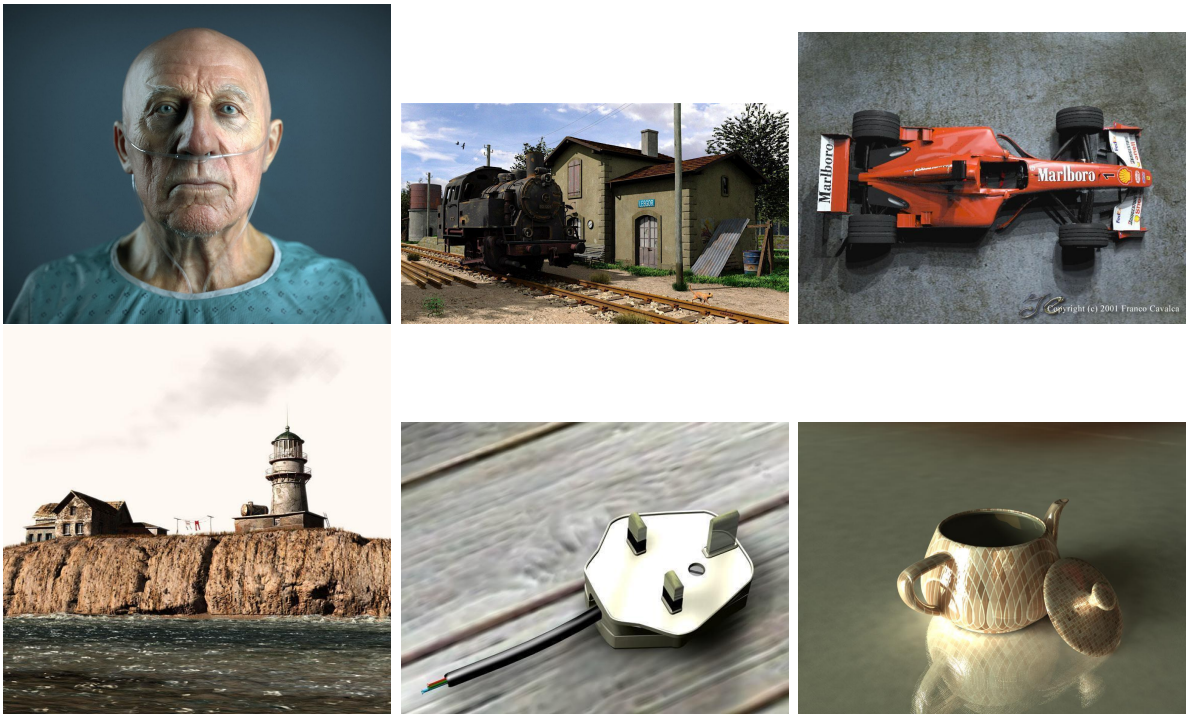


Figure 1.3: Examples of realistic CG photos.

1.1 CG versus Natural Multimedia Data Discrimination

Detecting computer graphics images has been studied in decades, starting with classification methods on the type of images [2], mostly for differentiation between graphics and photographs. However, these methods only targeted to simple graphics images, e.g., cartoons, clip arts or drawing, which are very different from the photographs. An example of the two kind of images are shown in Figure 1.4 (a) and (c). Shown in Figure (b) is an example of a photorealistic CG image, which is almost indistinguishable by human perception. Only since 2005, with the raising of digital image forensics, identifying photorealistic computer graphics became attractive to the multimedia forensic community with many studies on this problem. We can group these studies into 4 categories:

(a) A natural photograph. (b) A photorealistic CG image. (c) A graphic image.

Figure 1.4: Examples of a photorealistic CG image, a photograph and a graphic image.

- **Methods using Recording Devices properties:** Photographic images are created in general by a camera, or a scanner. These devices have various of characteristics that computer could not reproduce in CG images. Some studies have proposed solutions by analyzing physical variances in the image (e.g., local patch statistics, fractal and quadratic geometry, surface gradient) as introduced by Ng. et al. [18] in 2005. Dehni et al. in 2006 and Khanna et al. in 2008 proposed methods for solving this problem by evaluating the noise introduced by the recording device, presented in [9] and [15], respectively. Dirik et al. [10] and Gallager and Chen [21] introduced methods to discriminate images created by the computer from the ones captured by the camera by detecting traces of demosaicing and chromatic abberation.

- **Methods from Natural Image Statistics:** Natural images have some special properties different from the other types of images. One of it is the sparse distribution of the wavelet coefficients which are suitably modeled by a generalized Laplacian density [37]. Hence, in 2005, Lyu and Farid proposed a method in [36] to differentiate between CG and photographic images by estimating statistical differences in wavelet-based decomposition, which can be considered as one of the

first approaches to this problem. Another method working on wavelet domain was proposed by Wang and Moulin [56] in 2006, where they discovered that the characteristic function of the coefficient histogram of wavelet sub-bands is different for CG and natural images. In 2007, Chen et al. [5] applied an idea from steganalysis to deal with this problem on wavelet domain. Another method based on the Benford's law on Discrete Cosine Transform is proposed by Xu et al. [58] in 2011.

- **Methods using Visual Features:** Visual descriptors refer to features motivated by visual appearance such as color, texture, edge properties, and surface smoothness [40]. These kind of methods were used mostly to compared between simple CG with photograph, but some of them are able to used in detecting highly realistic CG images. In 2006, Wu et al. [57] proposed a method using several visual clues, e.g., the number of unique colors, local spatial variation of color and obtained highly performance on classification. In 2007 Ladonde and Efros [28] proposed an method based on an assumption that color composition of natural images is not random, and some compositions appear more likely than the others. Hence, color compatibility can be used as discriminate features to distinguish computer graphics from photographic images. The other method in this group is proposed by Pan et al. [41] in 2009, in which they used fractal dimension to detect CG images on the Internet.

- **Hybrid and other methods:** Sutthiwan et al. proposed two different methods in [49, 50] using high dimension feature vectors to differentiate CG and natural images. Sankar et al. [47] in 2009 introduced a method by simply combine the features from various previous state-of-the-art methods. In 2011, a method solving this problem by

combining various data in a hybrid approach was proposed by Conotter et al. [8]. Recently, Wand and Double [55] and Kee and Farid [26] proposed methods to measure visual photorealism, which can be applied to measure the degree of photorealism in an image. However, such measure is still weak and thus a better measure is required to be able to differentiate between CG and natural data.

Although many interesting methods have been proposed, most of these methodologies do not achieve satisfactory performance in the detection of CG characters. Some examples of human characters are shown in Figure 1.5 where CG and natural faces are almost perceptually indistinguishable. As a matter of fact, generic methods able to recognize synthetic images cannot cope with the complexity of this specific problem, which requires the use of specialized models.

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

Figure 1.5: Examples of highly realistic CG characters.

Only the right-most picture is photographic while the first 3 pictures are computer generated.

People are, in many cases, a crucial target for computer graphics community, hence designers often try their best to create realistic virtual characters. Indeed, computer generated (CG) characters are increasingly used in many applications such as talking-faces, e-learning, virtual meeting and

especially video games. Since the first virtual newsreader Ananova[1] introduced in 2000, significant improvements have been achieved in both quality and realism of CG characters, which are nowadays often very difficult to be distinguished from real ones. Therefore, we consider critical to be able to distinguish between computer generated and photographic faces in multimedia data. This is the objective of this doctoral study. In the next section, our proposed solutions and innovation are briefly reported.

1.2 Proposed Solutions and Innovation

The objective of this doctoral study is to develop efficient techniques to distinguish between computer generated and natural human faces, which can be used in various contexts, i.e., in both still images and videos, with different face poses or in complex situations, e.g., occlusions, different lightning conditions or varying facial animations.

Given such requirements, during this doctoral research we contributed in each application scenario proposing the following approaches :

- **Discrimination based on Asymmetry Information (AsymMethod)**

  Usually, when creating a human face, designers only create a half of the face, then replicate it to form the other half. Based on that idea, we proposed a geometric approach supporting the distinction of CG and real human faces, which exploits face asymmetry as a discriminative feature. This method can be used without requiring classification tools and training or combined with existing approaches to improve their performances.

[1] http://news.bbc.co.uk/2/hi/entertainment/718327.stm

- **Discrimination through Facial Expressions Analysis (Express-Method)**

  As mentioned, we aim at developing methods not only for still images, but also for discriminating between CG versus natural subjects in video sequences. The first method can work also on a single shot, but when a video source is available, much more information can be extracted from the data. For instance, CG and real characters can be discriminated by analyzing the variation of facial expressions in a video. The underlying idea here is that humans can produce a large variety of facial expressions with a high range of intensities. For example, the way a person smiles changes depending on his/her mood, and hence the same expression is usually produced in similar but not equal ways. Computer generated faces, instead, typically follow repetitive patterns, coded into pre-defined models. Therefore, their variations are not as wide as in real faces. Consequently, a CG character can be theoretically identified by analyzing the diversity of facial expressions, through appropriated models. In this method, face and expression models are created through sets of feature points identified in critical areas of the face.

  To the best of our knowledge, this is the first multimedia forensics approach that aims at discriminating between CG versus natural multimedia data in video sequences.

- **Identifying synthetic facial animations through 3D face models (ModelMethod)**

  The last method is aim at even more complicated situations, where characters are moving and turning their faces. The analysis of the 3D model allows to deal more easily with human faces, which are various, deformable and can occur in multi ways depending on expression,

lightning condition, poses, etc. Therefore, we propose to study the evolution in chronological order of the 3D model of the analysed character, assuming that its variations allow to reveal synthetic animation. Indeed, facial animation following fixed patterns can be distinguished from natural ones which follow much more complicated and various geometric distortion, i.e., bigger variations in the 3D model deformation.

To summarize, we primarily studied geometric-based techniques, which make use of measurements on human faces. We investigated both image and video CG versus natural discrimination methods, exploiting knowledge of objects in the world and of the process of image formation. All of the proposed methods can be used as standalone methods or combined with existing approaches.

Following, we briefly present our main contribution to this field:

- **CG versus natural human faces:** To the best of our knowledge, in the context of Multimedia Forensics, we proposed first approaches to deal with the problem of differentiate between CG and natural human faces. This is also the first time the problem of discrimination of CG versus natural data in videos are considered.

- **Geometric-based techniques:** the modeling and estimation of geometry is less sensitive to resolution and compression that can easily confound statistical properties of images and video, i.e., our proposed methods are robust with different situations.

- **Model-based techniques:** analyzing through 3D models better fits the analysis of human faces, taking into account their variety, deformability, diversity of expressions, different poses, as well as the external factors such as illumination conditions and framing, etc.

1.3 Thesis Structure

The thesis is organized in 5 chapters describing the research field together with the main objectives of this doctoral study.

Chapter 2 presents an overview on visual realism of computer graphics and the way synthetic facial animations are created. CG versus Natural Data Discrimination methods are also deeply reviewed in this chapter.

In Chapter 3, the details of our proposed approaches are presented and discussed.

Chapter 4 discusses about datasets used in our experiments together with the experimental results.

Finally, Chapter 5 collects some concluding remarks and discusses the open issues related to CG versus natural multimedia data discrimination.

Chapter 2

State of the Art

This chapter presents a concise overview about discrimination between computer generated and natural multimedia data. We also focus our attention on visual realism of computer graphics and the way that synthetic facial animations are created.

"Study the past, if you would divine the future."
Confucius

2.1 Visual Realism of Computer Graphics

Since the level of photorealism of a CG product is considered as a value of success, computer graphics community aware of the important of photorealism and its perception. Such studies on perception of photorealism offer some hints about the perceptual differences between natural and artificial multimedia data.

In 1986, Meyer et al. [39] showed to 20 people pairs of CG/natural images and asked them to label the images. 9 over 20 people who joined

Figure 2.1: Examples of the pictures tested from [20].

For each pair of image, the left picture is computer generated while the right one is natural. Figure source: [20].

that test selected the wrong answer.

McNamara [38] in 2005 carried a similar experiment with more complex CG images. They invited 20 people and show them randomly 10 images, and asked them to label which images are CG, which are natural. The results showed that some high quality CG images are undistinguishable under some conditions of lighting.

More recently, in 2012, Farid and Bravo [20] conducted some experiments that used human face images in different resolution, JPEG compression qualities, and color to explore the ability of human to distinguish computer generated faces from the natural ones. The CG images are downloaded from the Internet. The experiments provided a probability that an image that is judged to be a photograph is indeed a true photograph, which has 85% reliability for color images with medium resolution (between 218×218 and 436×436 pixels in size) and high JPEG quality. The reliability drops for lower resolution and grayscale images. This work shows that the CG faces from the Internet are quite distinguishable for human. However, not all of the selected CG images from this study are highly realistic, for example the CG faces shown in Figure 2.1 are less realistic than the ones from Figure 1.5.

2.2 CG versus Natural Data Discrimination methods

As mentioned in Chapter 1, since about 10 years, the research on multimedia forensics have started developing methods to identify photorealistic CG data, mainly focusing on still pictures. These methods can be grouped into 4 categories, illustrated in Figure 2.2. Details of these groups are presented in the following sections.

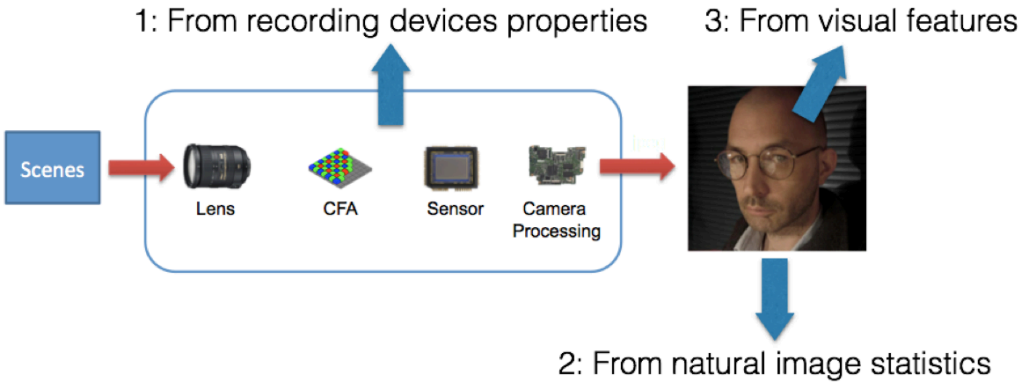


Figure 2.2: State-of-the-art approaches on still images.

The first group uses the recording device properties, mostly by analyzing the noises from the camera sensor, to identify natural images. Second group differentiates the two types of images based on natural image statistics like wavelet coefficients while third group investigates information from the visual features of the image. The last group contains other and hybrid methods from the first three groups.

2.2.1 Methods from Recording Devices properties

Characteristics of the recording devices and the processing from the manufacturer software are often presented in the images, e.g., chromatic aberration or distortions (see Figure 2.2). Further more, most of digital camera now are using charge-coupled device (CCD) or metal-oxide-semiconductor (CMOS) sensors, which contain imperfect patterns such as pattern noise, dark current noise, shot noise, and thermal noise [22]. Such noises are typical for natural images and do not exist in most of the CG images. Hence,

natural images can be detected based on the analysis on these characteristics.

In 2005, Ng et al. [18] identified three differences between photographic images and CG images:

1. Natural images are subject to the typical concave response function of cameras.

2. Colors of natural images are normally represented as continuous spectrum while in CG the color channels are often rendered independently.

3. Natural objects are more complicated while CG objects are normally modelled by simple and coarse polygon meshes.

Hence, the authors proposed a method using image gradient, Beltrami flow vectors, and principal curvatures to analyze the three mentioned differences, which is summarized in Figure 2.3. Using SVM classification on the Columbia open data set [17], they achieved an average classification accuracy of 83.5%.

Dehnie et al. [9] in 2006 indicated that noise patterns, extracted by a wavelet denoising filter, of natural images is different from the ones in CG images. Hence, an input image can be classified as CG or natural based on its correlation to the reference noise patterns. On their own data set, the method achieved an average accuracy of about 72%.

In 2007, Dirik et al. [10] proposed a method to detect natural images by analyzing the traces of demosaicking and chromatic aberration. According to them, in natural images, the changes of the color filter array (CFA) is smaller, comparing to CG images, when an input image is re-interpolated. The authors also combined their proposed features with wavelet-based features from the method of [36] to obtain a better performance. Their method achieved an average classification accuracy of about 90% on their own data set.

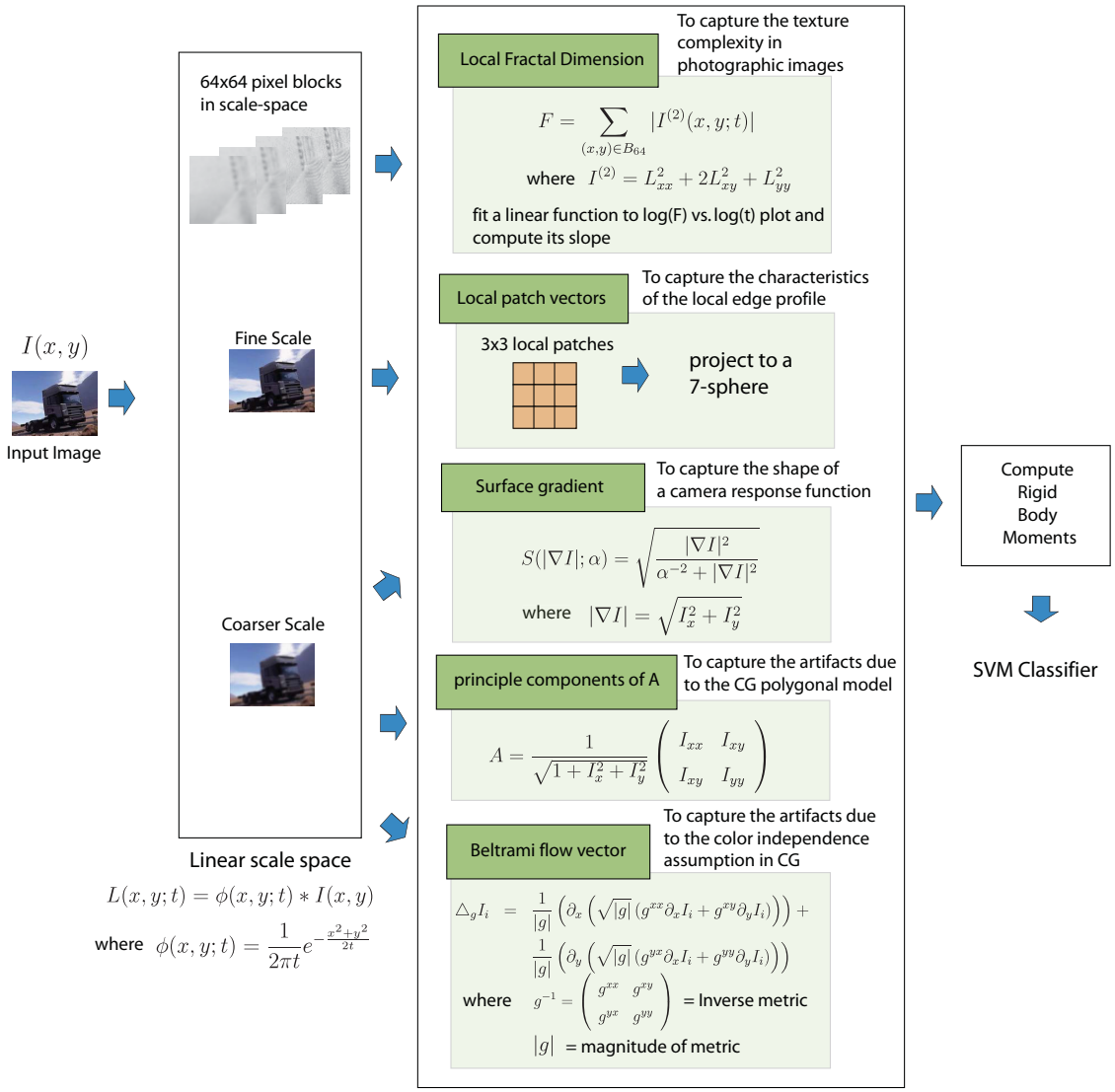


Figure 2.3: Schema of the method in [18].

Figure source: [40].

Based on the estimation of the noise pattern of the devices, in 2008, Khanna et al. presented in [15] a method for discriminating between scanned, non-scanned, and computer generated images. In this study, the basic idea is analyzing noises of the scanner from row to row and column to column, and then combining them with the noise of the camera, calculated as difference between the de-noised image and the input one. Their idea is summarized in Figure 2.4. On their own data set, the method achieved an average accuracy of 85.9%.

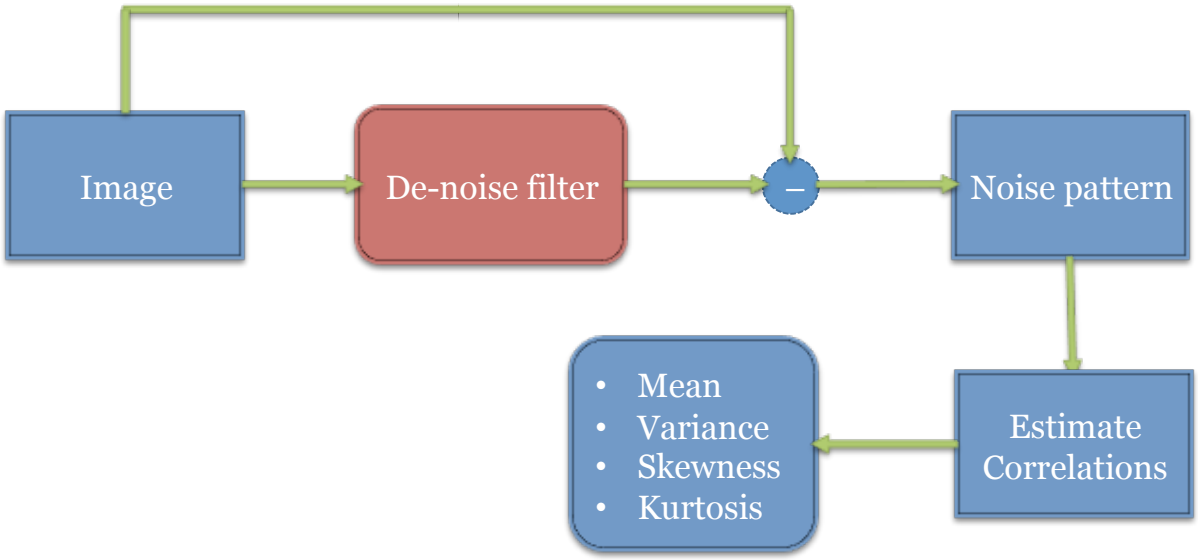


Figure 2.4: Schema of the method in [15].

Gallagher and Chen [21] in 2008 demonstrated that the CFA in original-size natural images can be detected. Firstly, they use a high-pass filter to highlight the observation that interpolated pixels have a smaller variance than the original ones. Then, they analyze the variance on green color channel from the diagonal scan lines since interpolated and original pixels respectively occupy the alternate diagonal lines. Their method achieved an average classification accuracy of 98.4% on the Columbia open dataset [17].

2.2.2 Methods from Natural Image Statistics

Natural images have some particular statistical properties that do not appears frequently in other types of images (computer generated, microscopic, aerial, or X-ray images). One of the important natural image statistics is the sparse distribution of the wavelet coefficients: natural images that are suitably modeled by a generalized Laplacian density [37]. Shown in Figure 2.5 is the wavelet coefficient distributions of the second-level horizontal subbands, respectively, for a photograph and a computer graphics. Based on these statistical differences, some methods have been developed to distinguish CG images from the natural ones.

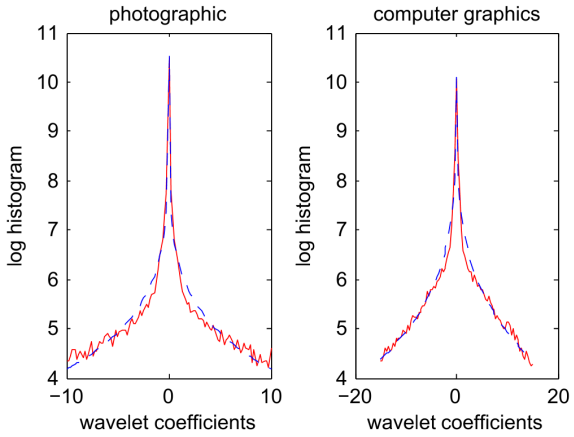


Figure 2.5: The log-histogram of the first level detail wavelet coefficients.

The wavelet coefficients are computed using Daubechies filters. Dash-line is the least squared fitted generalized Laplacian density. Figure source: [40].

The first approach in this group was introduced in 2005 by Lyu and Farid [36], which is normally considered as one of the first forensics approaches in this problem. In this study, the authors use a statistical model on 216-dimensional feature vectors calculated from the first four order statistics of the wavelet decomposition. The idea of this method is summarized in Figure 2.6, where input image is first decomposed into three levels, then four moments (mean, variance, skewness, and kurtosis) of the wavelet coefficient

distribution and the linear prediction error distribution are computed for each subband as features, finally a classic Support Vector Machine classifier is applied.

They obtained classification rate of 66.8% on the photographic images, with a false-negative rate of 1.2%. Datasets were collected from the Internet.

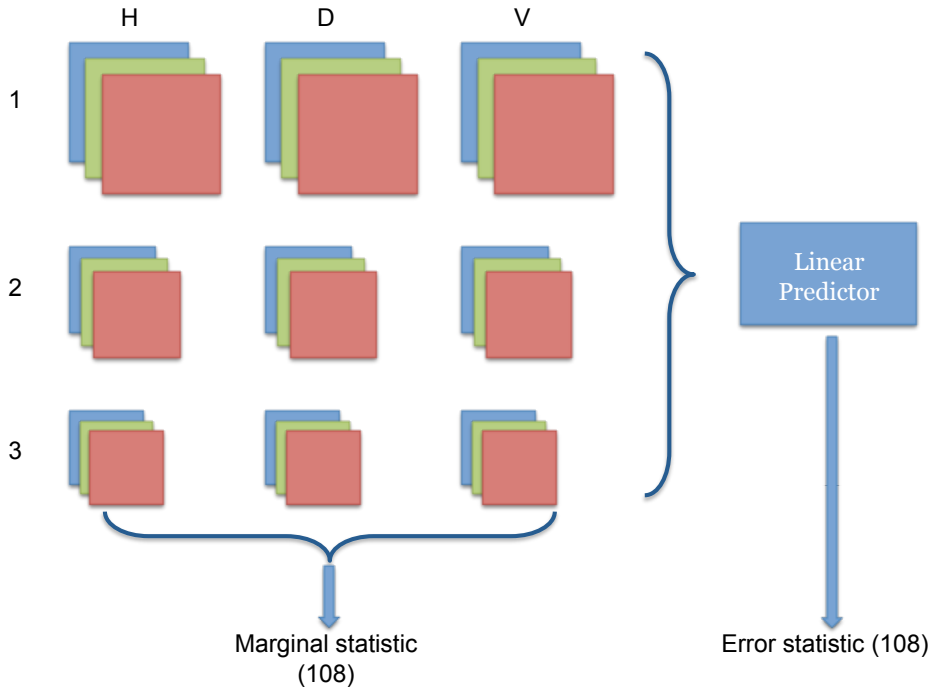


Figure 2.6: Schema of the method in [36].

In a similar way, in 2006, Wang and Moulin [56] used a statistical model with only 144-dimensional feature vectors achieving slightly better results with respect to [36]. With less number of features, the computation speed is about four times faster than that of Lyu and Farid [36]. They also compared indirectly with the method in [18] and the obtained computational is 70 times faster.

In 2007, Chen et al. [5] applied an idea from steganalysis, in which they compute three levels of wavelet decomposition on the original and the prediction images for each of the HSV color channels. The first three sta-

tistical moments were computed for each subband which gave 234 features in total. On a data set expanded from the Columbia open data set [17], they achieved a classification accuracy of 82.1%.

Xu et al. [58] proposed a method in 2011 based on Benford's law to identify natural images. According to the authors, statistics of the most significant digits extracted from Discrete Cosine Transform (DCT) coefficients and magnitudes of the gradient image of natural images are different from CG images. They achieved an accuracy of 91.6% on their own dataset.

2.2.3 Methods from Visual Features

In 2006, Wu et al. [57] used the number of unique colors, local spatial variation of color, ratio of saturated pixels, and ratio of intensity edges as discriminative features and used k-NN to classify CG and natural images. On their own dataset, they achieved an average accuracy of 95%.

In 2007, Lalonde and Efros [28] proposed a method that can identify composite images by compare the color distribution between the background and the foreground objects. Their idea is based on an assumption that color composition of natural images is not random, and some compositions appear more likely than the others. This idea can be apply to differentiate natural images from CG images where color compatibility can be used as discriminate features. Figure 2.7 illustrates the idea of this method.

Pan et al. [41] in 2009 proposed a method that use fractal dimension to analyze the differences between CG and natural images. In particular, they computed the simple fractal dimensions on the Hue and Saturation components of an image in the HSV color space as discriminative features. On their own data set, they achieved an average accuracy of 91.2%, while the method by Lyu and Farid [36] achieved 92.7% on the same data set.

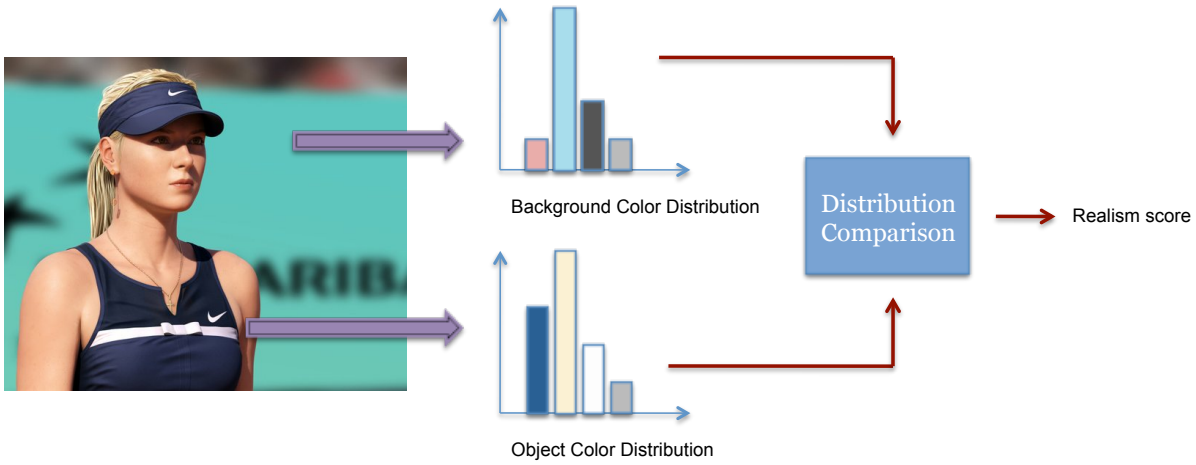


Figure 2.7: Idea of the method in [28].

2.2.4 Hybrid and Other Methods

In 2009, Sutthiwan et al. [49] considered the JPEG horizontal and vertical difference images as first-order 2D Markov processes and used transition probability matrices to model their statistical properties [40]. On their data set, they achieved the average accuracy of 94.0% by using SVM with the 324 dimensional feature vectors. An improvement was proposed by using Adaboost: the number of features is reduced to 150 and the accuracy increased to 94.2%. In [50], they extended the work by Chen et al. [5]. In this study, they computed the features on the original image, its JPEG coefficient magnitude image and the residual error. On the same dataset, they achieved an accuracy of 92.7% by using Adaboost on 450 dimensional feature vectors.

Sankar et al. [47] in 2009 proposed a hybrid method, in which they combined all the features from Ianeva et al. [23], Chen et al. [5], Ng et al. [18], and Popescu and Farid [42]. On the Columbia open data set [17], they achieved an average accuracy of 90%.

In 2011, Wang and Doube [55] proposed a method to measure visual realism based on the following characteristics of natural images: surface

roughness, shadow softness, and color variance. However, they evaluated the realism by comparing the new video games to the old ones, hence such measure is considered week and better measures are needed.

In 2011, Conotter and Cordin in [8] developed an hybrid method, which not only exploits the higher-order statistics of [36] but also uses the information from the image noise pattern (36-dimensional feature vectors calculated from the PRNU [33] and used also for source identification [7]). Figure 2.8 illustrate the idea of this method.

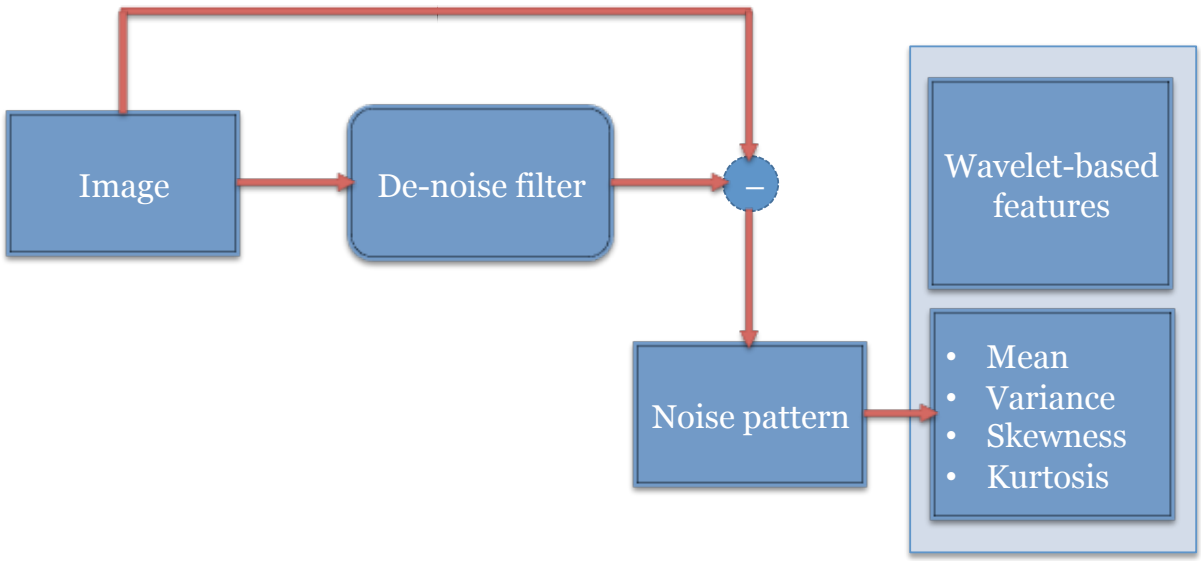


Figure 2.8: Schema of the method in [8].

We introduced a series of State-of-the-Art methods so far together with their performances, however, it is not easy to have a direct comparison since most of the methods were tested on different datasets. Thus, to summarize all of them, we reported their performance together with the datasets in Table 2.1.

Table 2.1: Summary of State-of-the-Art methods on CG versus Natural Multimedia Data Discrimination.

Approach	Study	Feature Dimension	Dataset	Highest Accuracy
Recording Devices properties	Ng et al. [18]	192	Columbia	83.5%
	Khanna et al. [15]	1	Internet Images	85.9%
	Dehnie et al. [9]	15	Internet Images	72%
	Dirik et al. [10]	77	Internet Images	90%
	Gallagher and Chen [21]	1	Columbia	98.4%
Natural Image Statistics	Lyu and Farid [36]	216	Internet Images	66.8%
	Wang and Moulin [56]	144	Internet Images	Comparable to [36]
	Xu et al. [58]	54	Internet Images	91.6%
Visual Features	Wu et al. [57]	38	Internet Images	95%
	Lalonde and Efros [28]	-	Internet Images	Not mentioned
	Pan et al. [41]	30	Internet Images	91.2%
Hybrid and Others	Chen et al. [5]	234	Columbia	82.1%
	Sutthiwan et al. [49]	150	Internet Images	92.7%
	Sutthiwan et al. [50]	450	Internet Images	94.2%
	Sankar et al. [47]	1	Columbia	90%
	Conotter and Cordin [8]	228	Internet Images	Comparable to [36]

2.3 Generating Synthetic Facial Animations

Understanding how synthetic faces are generated and animated is the basis for defining suitable algorithms to model them and to discriminate them for natural images.

There are studies dating back to the 70s that analyse facial animations (see for instance [12]). The Facial Action Coding System (FACS) by Ekman [13] (updated in 2002 [11]) and the MPEG-4 standard [25] are the basis for most algorithms generating synthetic facial animations. According to FACS, face muscles are coded as Action Units (AUs) while expressions are represented as AUs combination. In MPEG-4, explicit movements of each face point are defined by Facial Animation Parameters (FAPs). These parameters (FACS of FAPs) make the existed physically-constructed model more realistic. Thus, synthesis of facial animations is performed by modeling the facial animations and controlling parameters (Lee and Elgammal [29]). Linear models, e.g., PCA by Blanz et al. [3] in 1999 and Chen et al. [4] in 2012, and bilinear models [6][59] have been used for facial expression analysis and synthesis.

In 2000, Seung et al. [48] discovered that a facial expression sequence lies on a low-dimensional manifold. Thus, based on that inference, nonlinear algorithms (e.g., local linear embedding (LLE) proposed by Roweis et al. [46]) have been applied to find manifold from face datasets. However, these data-driven approaches fail to find manifold representations where there are large variations in the expression data by different type of expression and different style of people [30].

Chapter 3

CG versus Natural Human Faces

In this chapter, we focus on the specific class of images and videos containing faces, since we consider critical to be able to discriminate between photographic faces and the photorealistic ones. To this aim, we present new geometric-based approaches relying on face asymmetry information and the repetitive pattern from CG animations. These methods are able to detect CG characters in both still images and videos with high performances.

"Who sees the human face correctly:
the photographer, the mirror, or the painter?"
Pablo Picasso

As mentioned in Chapters 1 and 2, we primarily studied geometric-based techniques, which make use of measurements on human faces, to discriminate between CG and natural faces. We investigated both image and video CG versus natural discrimination methods, exploiting knowledge of how synthetic animations are created and performed. One of the advantages of geometric-based techniques is that the modeling and estimation of geometry is less sensitive to resolution and compression that can

easily confound statistical properties of images and video. Furthermore, all of these methods can be used as standalone methods or combined with existing approaches. In the next sections, details of these methods are introduced.

3.1 Discrimination based on Asymmetry Information

To the best of our knowledge, when creating synthetic human faces, designers, in most cases, just make a haft of a face and then duplicate it to create the other one. Then, they often apply post processing to achieve photorealistic results but usually not modifying the geometry of the model. Hence, if a given face present a high symmetric structure, this could be considered as a hint that it is generated via computer. On the other hand, although human faces are symmetric, there does not exist a perfectly symmetrical face, as confirmed by Penton-Voak et al. in [14]. The combination of such two hints allow us to make the following assumption: the more asymmetric a human face, the lower its probability to be computer generated. Based on this assumption, we have developed a method (named AsymMethod) to compute asymmetry information and thus discriminate between computer generated and photographic human faces.

Our method contains three main steps as detailed in Figure 3.1: shape normalization, illumination normalization and asymmetry estimation. First, in shape normalization step, the input image is transformed into the 'standard' shape, i.e. is normalized into the same coordinate system for every face, in order to make the measurements comparable. Then, in illumination normalization step, unexpected shadows, which could affect the accuracy of the measurements, are removed from the normalized face. Asymmetry measurements which are stable under different face expressions are then calculated in asymmetry estimation step. Finally, based on these measure-

ments, we assign to the given face a probability whether it is computer generated or not.

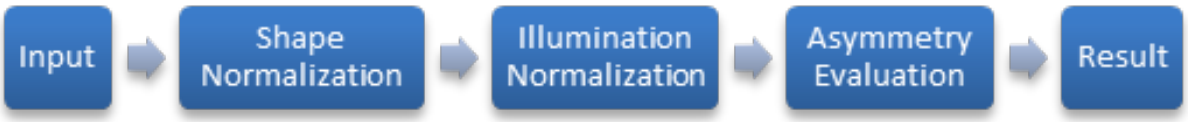


Figure 3.1: Schema of AsymMethod.

An example of the process is shown in Figure 3.2, where a) represents the input image, b) the normalized face, and e) the result after illumination normalization.

3.1.1 Shape Normalization

We apply the traditional approach from [32] to normalize a shape of a face in order to have a common coordinate system. This normalization is not only making the measurements easier, but allows to combine them with other facial features (e.g., EigenFace or Fisher Face). In particular, two inner eye-corners, denoted as C_1 and C_2 and the filtrum, denoted as C_3 of a face are chosen. The given face is then normalized by moving [C_1, C_2, C_3] into the normalized positions, by the following three steps as follows:

- Step 1. Rotate $\overline{(C_1, C_2)}$ into a horizontal line segment.
- Step 2. Apply the shearing transformation that make the philtrum be on the perpendicular line through the middle point of $\overline{(C_1, C_2)}$.
- Step 3. Scale the image that $\overline{(C_1, C_2)}$ has the length a and the distance from C_3 to $\overline{(C_1, C_2)}$ is b. See Figure 3.3 for example.

We used the same normalized values as mentioned in [32], which used the fix values of the special points: $C_1 = (40, 48)$, $C_2 = (88, 48)$, and $C_3 = (64, 84)$. The (0, 0) is the top-left corner. Figure 3.2 b) shows an example of this step applied to Figure 3.2 a).

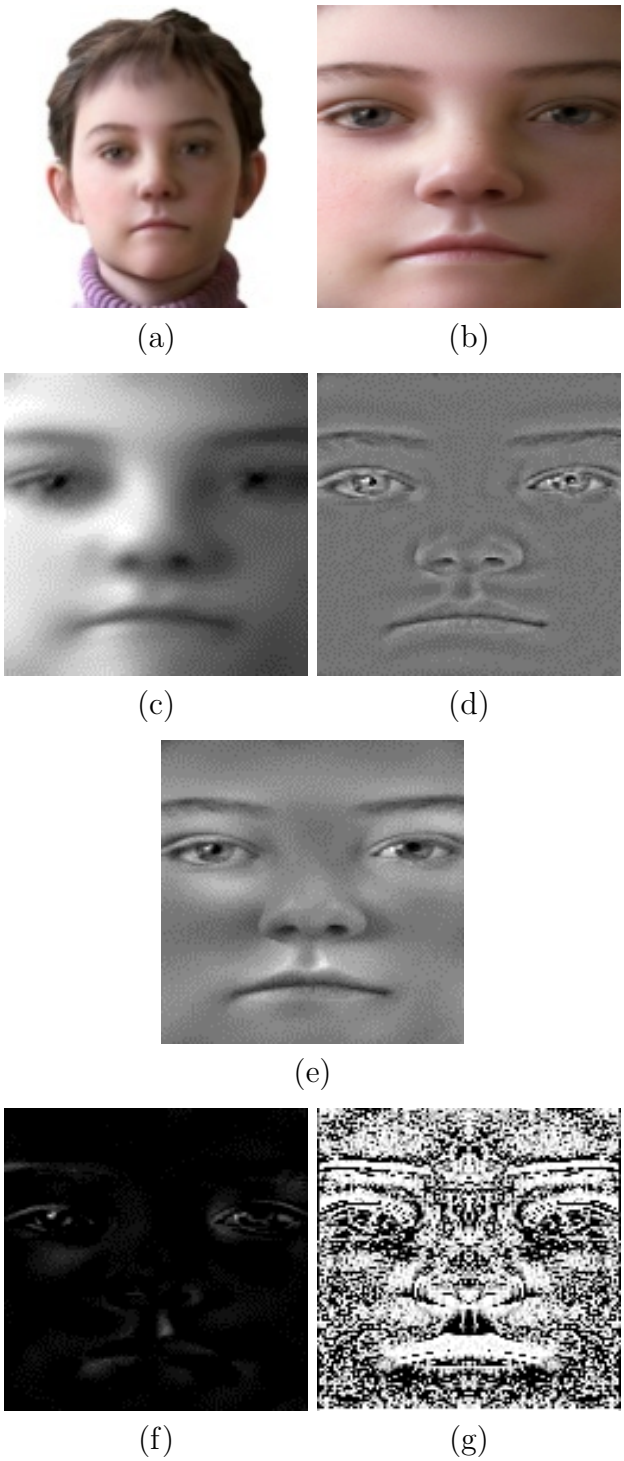


Figure 3.2: Face asymmetry estimation.

(a) input photo; (b) normalized photo; (c), (d) components of illumination normalization step; (e) result after illumination normalization; (f), (g) sub-results of asymmetry evaluation step.

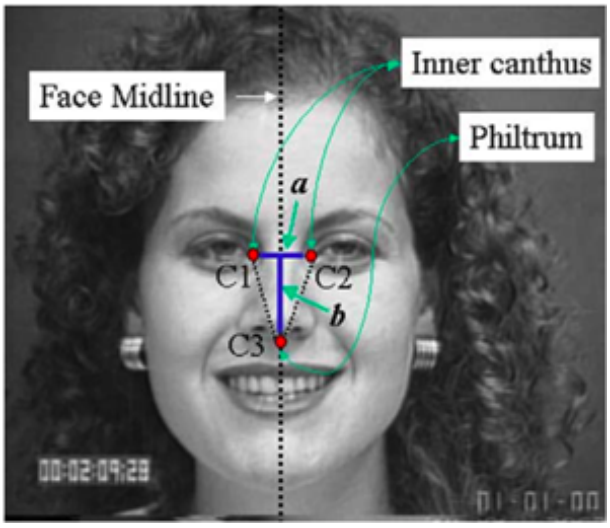


Figure 3.3: Face normalization via inner eye-corners and a philtrum.

Figure source: [32]

3.1.2 Illumination Normalization

Illumination causes most challenging problems in facial analysis. Asymmetry measure is calculated based on the intensity of the face image, thus, the shadows, which are usually quantized as low value regions, play an important role. However, what we need is the information of the face structure, without any effects from shadows or unexpected lighting illumination. Hence, illumination normalization is required in order to enhance the accuracy of the asymmetry measurements.

We apply the approach presented by Xie et al. in [19]. The basic idea is to use the albedo of large scale skin and background, denoted as $R_l(x, y)$ to split the face image I into large-scale and small-scale components.

Based on Lambertian theory, we have:

$$I(x, y) = R(x, y)L(x, y) \tag{3.1}$$

where R is the albedo of the face and L is the illumination. Estimating this information consists of as an ill-posed problem, hence Xie et al. in [19]

apply a transformation to overcome this issue as follows:

$$\begin{aligned} I(x,y) &= R(x,y)L(x,y) \\ &= \left(\frac{R(x,y)}{R_l(x,y))}\right)(R_l(x,y)L(x,y)) \\ &= \rho(x,y)S(x,y) \end{aligned} \tag{3.2}$$

where ρ contains the intrinsic structure of a face image, and S contains the extrinsic illumination and the shadows, as well as the facial structure. ρ and S are called small-scale features and large-scale features, respectively.

In order to split the image into large-scale and small-scale, the Logarithm Total Variance (LTV) estimation is used. This estimation is introduced in [16] and is the best method to extract illumination-invariant features so far. After splitting the image I into ρ and S, smoothing filter, which is also introduced in [19], are required to be applied on ρ in order to remove unexpected effects from the decomposition in (3.2).

An example of this step is shown in Figure 3.2 where c) and d) represent the large-scale and small scale components, respectively, after applying LTV on the image b). The illumination normalized result is shown in e).

3.1.3 Asymmetry Evaluation

In order to estimate asymmetry, we use the measure introduced by Liu et al. in [32], which is less depend on face expressions. Let us denote the density of the image with I, and the vertically reflected of I with I'. The edges of the densities I and I' are extracted and stored in I_e and I'_e, respectively. Two measurements for the asymmetry are introduced as follows:

Density Difference (D-Face):

$$d(x,y) = \|I(x,y) - I'(x,y)\| \tag{3.3}$$

Edge orientation Similarity (S-Face):

$$s(x,y) = \cos\left(\theta_{I_e(x,y),I'_e(x,y)}\right) \tag{3.4}$$

where $\theta_{I_e(x,y),I'_e(x,y)}$ is the angle between the two edge orientations of images I_e and I'_e, at position (x,y). Figure 3.2 (e) shows the estimated frontal face resulting from the illumination normalization step. In Figure 3.2 (f) and (g) the D-Face and S-Face are shown, respectively.

Based on these measurements, we can estimate the asymmetry of a given face photo since the higher the value of D-Face, the more asymmetric is the face, and the higher the value of S-face, the more symmetric the face. The total difference of D-Face and total dissimilarity of S-Face are calculated as follows:

$$D = \frac{\sum_{x,y\in\Omega} d(x,y)}{\eta_1}; \tag{3.5}$$

$$S = 1 - \frac{\sum_{x,y\in\Omega} s(x,y)}{\eta_2} \tag{3.6}$$

where η_1, and η_2 are the normalized thresholds, which scale D and S into (0; 1), and Ω is the estimated region. Since our images are normalized to the fixed size 128×128, and Ω is fixed as in [32], both thresholds η_1, and η_2 are fixed.

Finally, we assign to image I an exponential probability to be computer generated, as follows:

$$P = \lambda e^{-\lambda\sqrt{D^2+S^2}} \tag{3.7}$$

where λ is a constant (we use $\lambda = 1.0$). If P is over a threshold τ, I is classified as a computer generated human face (we use $\tau = 0.5$).

The results, which are introduced in details in Chapter 4, show that AsymMethod can be used as a stand alone method or in combination with other information to improve state-of-the-art techniques for still images.

In the next section, ExpressMethod, which discriminate between CG and natural faces in videos or sequences of faces, is presented.

3.2 Discrimination through Facial Expressions Analysis

In order to deal with more complicated situations in videos or sequences of images, we proposed a second method, namely ExpressMethod, to distinguish between CG and real characters by analysing facial expressions. The underlying idea is that facial expressions in CG human characters follow a repetitive pattern, while in natural faces the same expression is usually produced in similar but not equal ways (e.g., human beings do not always smile in the same way). Our forensic technique take as input various instances of the same character expression (extracting corresponding frames of the video sequences) and determine whether the character is CG or natural based on the analysis of the corresponding variations. We show that CG faces often replicate the same expression exactly in the same way, i.e., the variations is smaller than the natural ones, and can therefore be automatically detected.

Our method contains five steps as detailed in Figure 3.4:

- From a given video sequence, frames that contain human faces are extracted in the first step A.

- Then, in step B, facial expression recognition is applied in order to recognize the expressions of the faces. Six types of facial expressions are used in this step, following the six universal expressions of Ekman (happiness, sadness, disgust, surprise, anger, and fear) [13] plus a 'neutral' one. Based on the recognition results, faces corresponding to a particular expression (e.g., happiness) are selected for the next

steps. Notice that the 'neutral' expressions are not considered, i.e., faces showing no expression are not taken into account for further processing.

- In the next step C the Active Shape Model (ASM), which represents the shape of a face, is extracted from each face. In order to measure their variations, all shapes have to be comparable.

- In step D, each extracted ASM is then normalized to a standard shape. After this step, all ASM shapes are normalized and are comparable.

- Finally, in step E, differences between normalized shapes are analysed, and based on the variation analysis results, the given sequence is confirmed to be CG or natural.

The right part of Figure 3.4 shows an illustration of the analysis procedure on happiness expression. Seven frames that contain faces are extracted in step A. Then, facial expression recognition is applied in step B and three happy faces are kept. For each face, the corresponding ASM model, which is represented by a set of reference points, is extracted in step C. Then, each model is normalized to a standard shape, step D. All normalized shapes are then compared together in step E, and based on the analysis results the given character is confirm as computer generated since the differences between the normalized shapes are small (details about the variation analysis are given in the following Subsection 3.2.5).

3.2.1 Human Faces Extraction

Face detection problem has been solved with the Viola-Jones method [54], which can be applied in real-time applications with a high accuracy. In this step, we reuse this approach to detect faces from video frames, and frames that contain faces are extracted. More details about this well-known

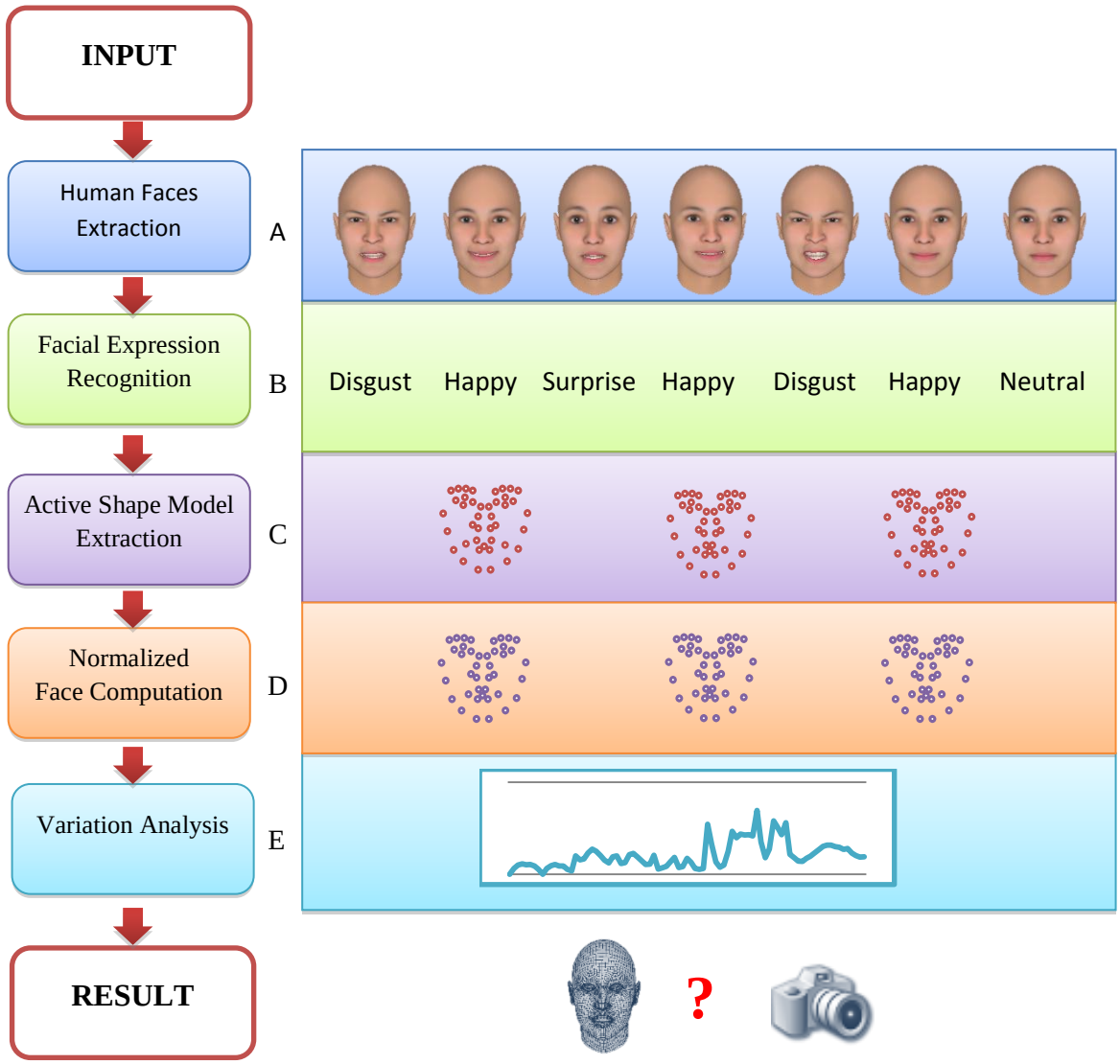


Figure 3.4: Schema of ExpressMethod.

A. Human faces are extracted from the video sequence(s). B. Facial expressions are recognized (in the example 3 happy, 2 disgust, 1 surprise and 1 neutral). C. Faces with the same expression are selected (in this example only happy faces) and their active shape models are extracted. D. The extracted models are normalized. E. Differences on the normalized models are analysed to determine whether the character is CG.

method can be found in [53] and [54]. It is worth mentioning that in this first work we do not face the problem of face recognition, thus assuming to have just a single person per video sequence (the analysed character).

3.2.2 Facial Expression Recognition

Facial expression recognition is a nontrivial problem in facial analysis. In this study, we applied an EigenFaces-based application [45] developed by Rosa for facial expression recognition. The goal of this step is to filter out the outlier expressions and keep the recognized ones for further steps. Notice that this application associates an expression to a given face without requiring any detection of reference points. In Figure 3.4 an example of results of this application is shown with 7 faces (3 happy, 2 disgust, 1 surprise and 1 neutral).

3.2.3 Active Shape Model Extraction

Input images for this step are confirmed to have the same facial expression of the same person, thanks to the preprocessing in the first two steps. In order to extract face shapes, which are used in our analysis, an alignment method is applied. In this step, we follow the Component-based Discriminative Search approach [31], proposed by Liang et al. The general idea of this approach is to find the best matching from the mode candidates, where modes are important predefined points on face images (e.g., eyes, nose, mouth) and are detected from multiple component positions [31]. Given a face image, the result of this step is a set of reference points, representing the detected face. In Figure 3.6 (a) an example of this step is shown, where the right image shows reference points representing the face in the left image. In this method, the authors exploit the so called ASM, which contains 87 reference points as shown in Figure 3.5. Another

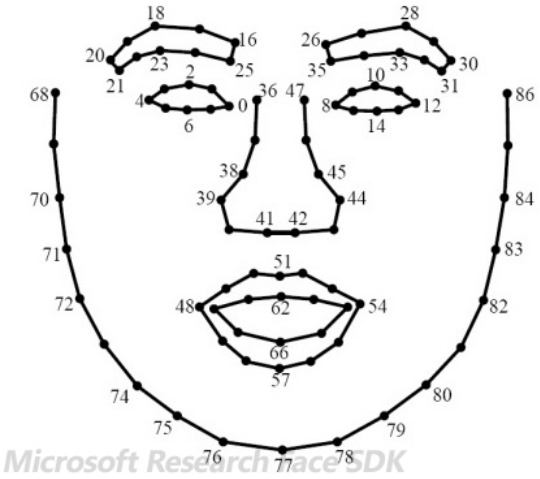


Figure 3.5: The 87 points of Active Shape Model (ASM).

Figure source: Microsoft Research Face SDK.

example of this step on a CG face is also reported in Figure 3.6 (c), where the left image shows the synthetic facial image and the right one shows the corresponding ASM.

3.2.4 Normalized Face Computation

ASM models precisely and suitably represent faces, but they are incomparable since faces could be different in sizes or orientations. They need to be normalized in order to be comparable. In this step, we apply the traditional approach from [32] to normalize a shape of a face in order to have a common coordinate system. This normalization is an affine transformation used to transform the reference points into fixed positions. Since eye inner corners and the philtrum are stable under different expressions, these points have been chosen as reference points. Shown in Figure 3.5, the reference points number 0 and 8 are two inner eye corners. The last reference point, the philtrum, can be computed via the top point of outer lip and two nostrils (point 51 and 41, 42 on the ASM model, respectively),

as follows:

$$p_{philtrum} = \frac{\frac{p_{41}+p_{42}}{2} + p_{51}}{2} \tag{3.8}$$

where p_{41}, p_{42}, and p_{51} are the reference points on the extracted ASM.

After computing the three reference points, each ASM model is normalized by moving $\{p_{41}, p_{42}, p_{phitrum}\}$ into their normalized positions, as follows: (i) rotate the segment $[p_{41}, p_{42}]$ into an horizontal line segment; (ii) shear the philtrum to be on the perpendicular line through the middle point of $[p_{41}, p_{42}]$; and finally (iii) scale the image so that the length of segment $[p_{41}, p_{42}]$ and the distance from $p_{philtrum}$ to $[p_{41}, p_{42}]$ have predefined fixed values (see [32] for more details). An illustration of this normalization is shown in Figure 3.3 in Section 3.1.

Shown in Figure 3.6 (b) and (d) are examples of the normalized faces after Face Normalization step. The left images show the normalized faces and the right ones show the normalized reference points.

3.2.5 Variation Analysis

In this step, differences among normalized ASM models are analysed in order to determine if a given character (and therefore the corresponding set of faces) is CG or real. We analyse the differences as described in the following paragraphs.

First, the distance $d_{i,p}$ of each reference point p on a model i to the average of all points p of all models is calculated as:

$$d_{i,p} = \|(x, y)_{i,p} - (\overline{x}, \overline{y})_p\| \tag{3.9}$$

where $(x, y)_{i,p}$ is the position of the reference point p on the model i; $(\overline{x}, \overline{y})_p = \frac{1}{N}\sum_{i=1}^{N}(x, y)_{i,p}$, where N is the number of normalized ASM models; and $\|\cdot\|$ is Euclidean distance.

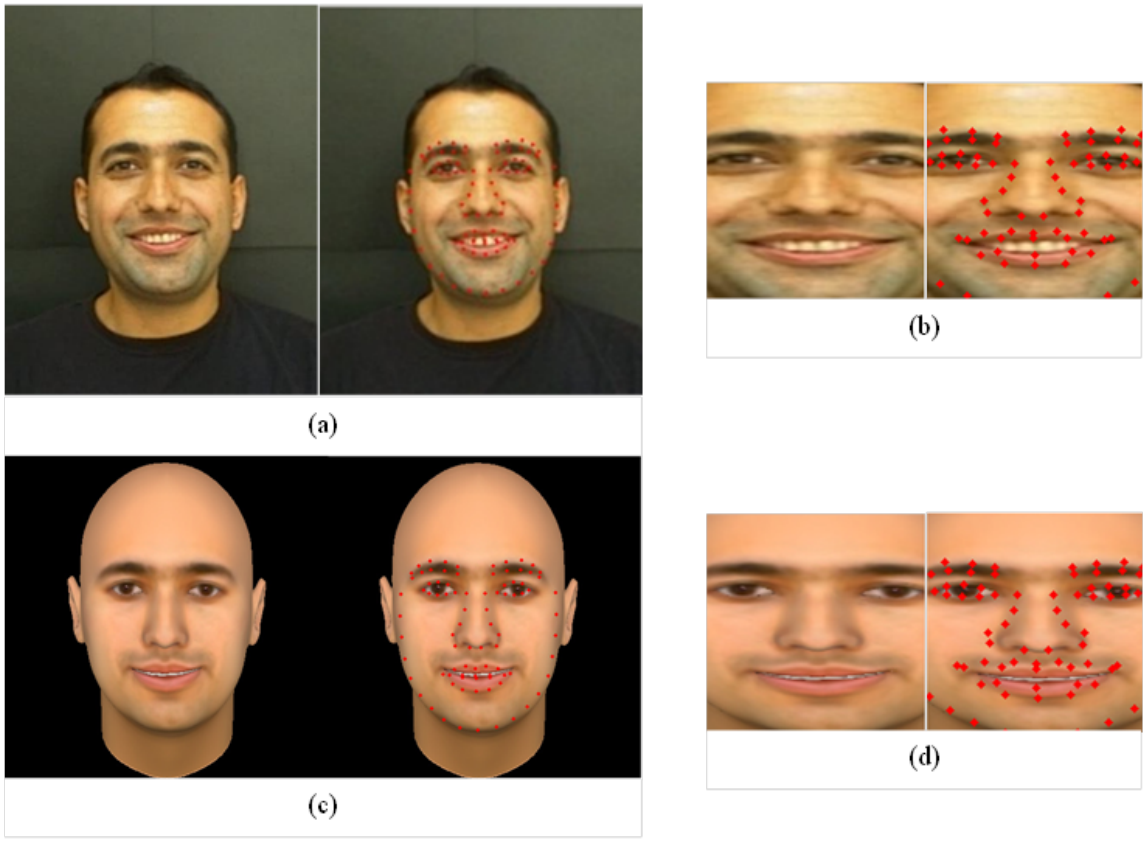


Figure 3.6: Examples of computed ASM and normalized ASM.

(a) and (c) show a photographic and a computer generated happy face, respectively, and their corresponding ASM points; (b) and (d) show the normalized images of (a) and (c), respectively, and their corresponding normalized points.

Depending on the facial expression ξ (among six universal expressions), a subset S_ξ of reference points (not all 87 points) are selected for the analysis. For example, with the happy facial expression ($\xi = 1$) only reference points from 0 to 15 and from 48 to 67, which represent the eyes and the mouth, are considered, i.e., $S_1 = \{0, 1, 2..15, 48, 49, .., 67\}$. The subsets are selected based on our experiments and suggestions from EMFACS [13], in which a facial expression is represented by a combination of AUs codes. Shown in Table 3.1 are the reference points selected in our method and the correspondent AUs codes from EMFACS. Some explanations of the AUs codes are also listed in Table 3.2. Full codes in EMFACS could be seen in [13].

Table 3.1: Expressions with Action Units and correspondent ASM points

ξ	Expression	Action Units (AUs)	Reference Points (S_ξ)
1	Happiness	6+12	$S_1 = \{0-15, 48-67\}$
2	Sadness	1+4+15	$S_2 = \{0-35, 48-57\}$
3	Surprise	1+2+5B+26	$S_3 = \{16-35, 48-67\}$
4	Fear	1+2+4+5+20+26	$S_4 = \{16-35, 48-57\}$
5	Anger	4+5+7+23	$S_5 = \{0-64\}$
6	Disgust	9+15+16	$S_6 = \{0-15, 48-67\}$

Two main properties are taken into account in this analysis: mean and variance, calculated as their traditional definitions:

$$\mu_p = \frac{1}{N}\sum_{i=1}^{N} d_{i,p}, \text{ and } \sigma_p = \frac{1}{N}\sum_{i=1}^{N} ||d_{i,p} - \mu_p||^2 \tag{3.10}$$

where μ_p and σ_p are the mean and variance of all distances $d_{i,p}$ at reference point p over all models.

The given set of models on expression ξ is confirmed to be CG or natural by comparing the *Expression Variation Value* EVV_ξ to the threshold τ_ξ.

[1] 5B mean slight intensity of AU 5, i.e., the upper lid slightly raises.

Table 3.2: Meaning of the AUs.

AU Number	FACS name
1	Inner Brow Raiser
2	Outer Brow Raiser
4	Brow Lowerer
5	Upper Lid Raiser[1]
6	Cheek Raiser
7	Lid Tightener
9	Nose Wrinkler
12	Lip Corner Puller
15	Lip Corner Depressor
16	Lower Lip Depressor
20	Lip Stretcher
23	Lip Tightener
26	Jaw Drop

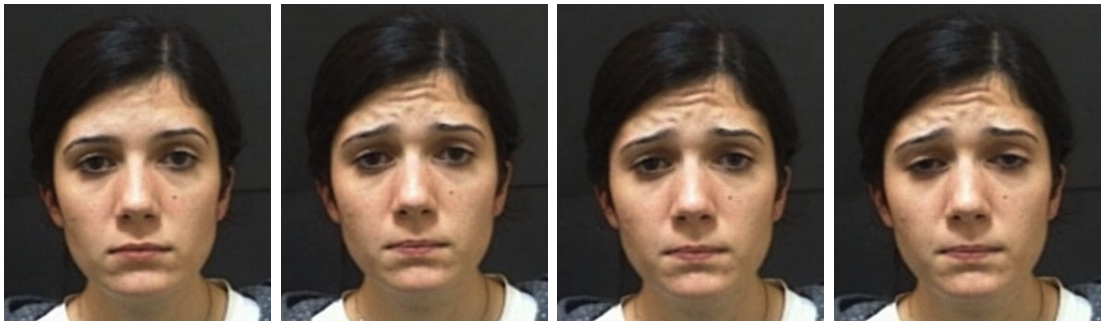
The value of EVV_ξ is computed as follows:

$$EVV_\xi = \alpha_\xi \frac{\frac{1}{|S_\xi|}\sum_p \mu_p}{\lambda_{1,\xi}} + (1-\alpha_\xi)\frac{max_p\{\sigma_p\}}{\lambda_{2,\xi}} \tag{3.11}$$

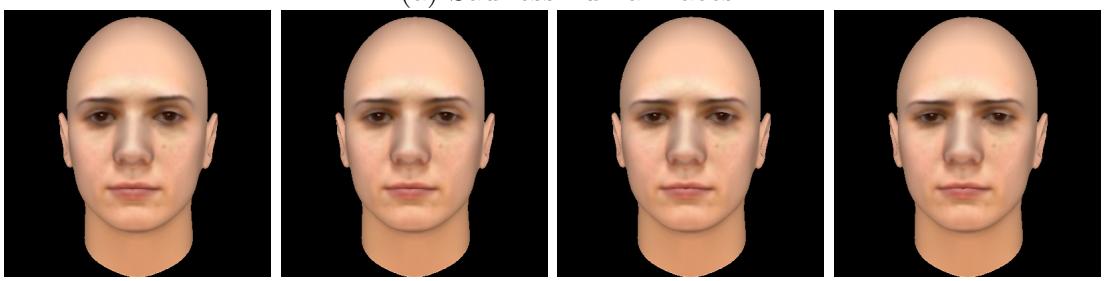
where α_ξ is a weighted constant, $\alpha_\xi \in [0;1]$; $\lambda_{1,\xi}$ and $\lambda_{2,\xi}$ are the normalization values used to normalize the numerators into $[0;1]$. In our experiments α_ξ are set to 0.7 for $\xi = 1, ..., 6$.

EVV_ξ is then compared with τ_ξ, recognizing the character corresponding to the set of faces as CG if $EVV_\xi < \tau_\xi$, natural otherwise.

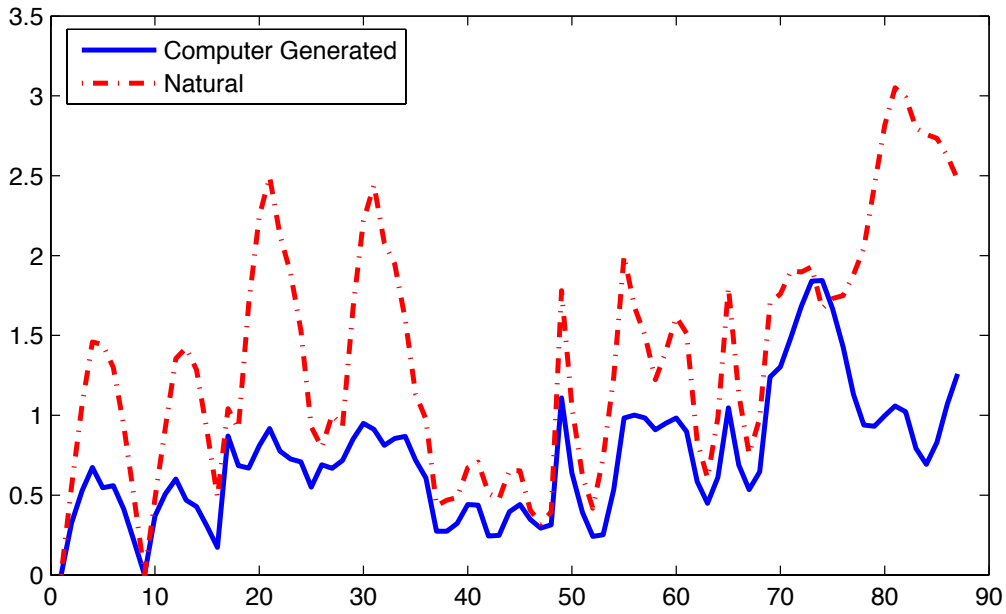
Shown in Figure 3.7 are the mean values, corresponding to all 87 ASM points, for the sadness expression ($\xi = 2$) analysed on the two set of images shown in Figures 3.7(a) and 3.7(b). The horizontal axis represents p, from 1 to 87, while the vertical axis shows the value of μ_p. Since the facial expression is sadness ($\xi = 2$), only the values from μ_0 to μ_{35} and from μ_{48} to μ_{57} are considered (see the selected reference points in Table 3.1). In this example, the *Expression Variation Value* EVV_2 of the CG face is 0.35 comparing to 0.74 of the natural one ($\tau_2 = 0.6$). Another example on



(a) Sadness human faces.



(b) Sadness CG faces.

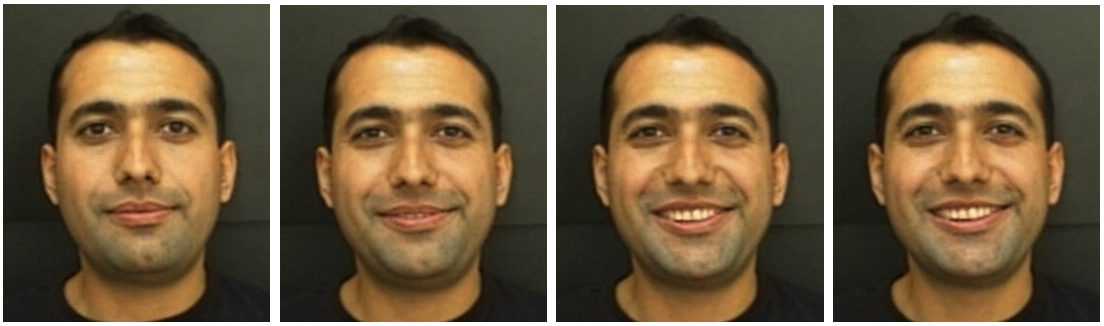


Differences on the mean of ASM points between (a) and (b).

Figure 3.7: Example of differences on the mean of ASM points on sadness expression.

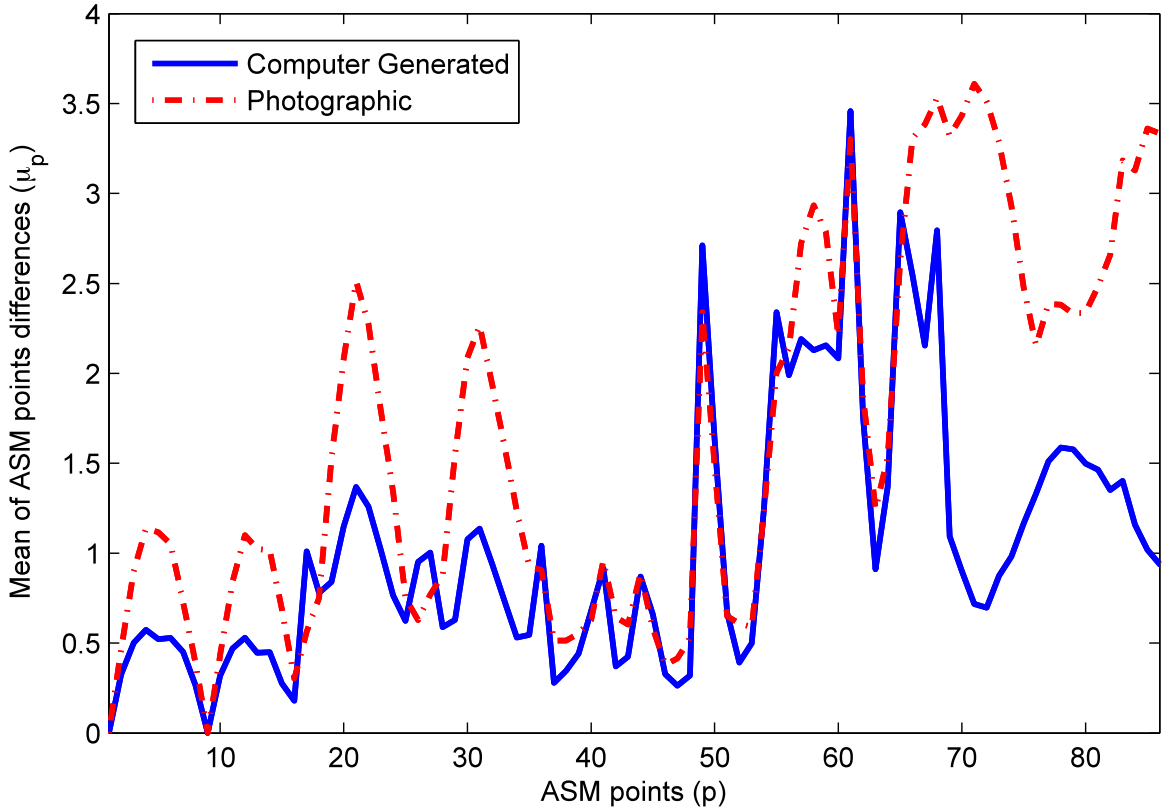
happiness faces is shown in Figure 3.8.

Values of the thresholds $\tau_{\xi(\xi=1..6)}$ are manually set based on experiments,



(a) Happiness human faces.

(b) Happiness CG faces.



Differences on the mean of ASM points between (a) and (b).

Figure 3.8: Example of differences on the mean of ASM points on happiness expression.

with the goal of keeping the miss classification as small as possible.

In the next section, our last proposed method is presented, which can be applied to more complex situations and animations without using different configurations while ExpressMethod requires the analysis of different sets of points for each single expression.

3.3 Identifying Synthetic Facial Animations through 3D Face Models

ModelMethod is a model-based method which allow to deal with natural behaviours of represented facial animation, where characters are moving and turning their faces. Computer generated facial animations are usually created by deforming a face model according to given rules or patterns. Typically, the deformation patterns are pre-defined and contain some parameters to rule the intensity of the expression. Also natural expressions are similar to one each other to some extent, due to physical limitations and personal attitudes, although in this case the variety of the expressions is much higher, taking into account asymmetries, different grades, bland of various sentiments, context, and so on.

What we propose here is to define a metric which we can be used to measure the diversity in animation patterns, in order to assess if the relevant video shows a synthetic characters or a human being. The idea is that a high regularity of the animations suggests that the face is computer generated, while a high variety is typically associated to natural images. The proposed method associates a 3D model to the face to be analyzed and maps the various instances of the face in the video to the model, applying appropriate deformations. Then, it computes a set of parameters associated to the relevant deformation patterns. Finally, it estimates the variation of the parameters along time to achieve a measure of the diversity,

thus leading to the classification of the face in synthetic or natural.

The choice of using a face model instead of working directly on the image is motivated as follows:

- The face model is less dependent on changes of the pose;
- Meaningful information about the whole face is taken into account instead of separate feature points;
- The face model reconstruction does not require all facial feature points, which are not always available due to occlusion or lighting condition, and can be computed for many more instances of the face within the video.

The proposed method, illustrated in Figure 3.9, consists of 3 main steps as below:

- **(A) Video normalization:** the video sequence is brought to standard parameters in terms of resolution and framerate, so as to minimize possible alterations of the model caused by different video formats;
- **(B) Face model reconstruction:** facial feature points, which represent the face shape, are extracted via Active Shape Model (ASM) and a face form in 3D is reconstructed by modelling a neutral shape to best approximate the extracted ASM;
- **(C) CG characters identification:** the sequence of actualized 3D face models is represented by applying a Principle Component Analysis (PCA), and the variations of the obtained feature vectors are used to classify the face.

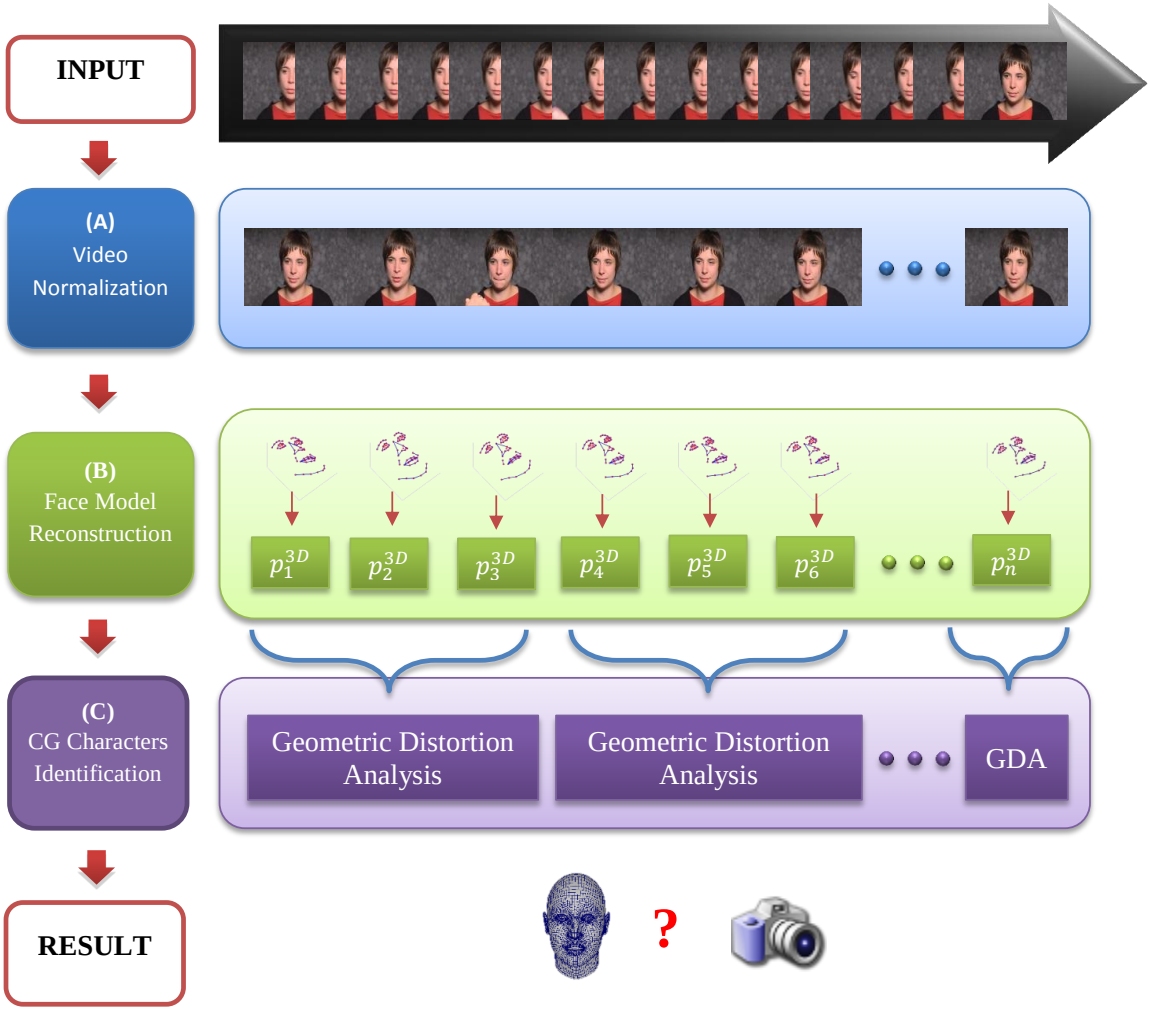


Figure 3.9: Schema of ModelMethod.

Step (A), the same amount of frames of are extracted every second. Step (B), based on the extracted ASM points, the face model is reconstructed for the face in each frame. Step (C), analyze the variations of the models in time order to identify the synthetic animation.

3.3.1 Video Normalization

Since the proposed method applies to any kind of video source, first of all we normalize the source to a standard format to avoid variations in the model due to the characteristics of the video. The frame rate is therefore reported in the range 10-12 frames/second, which are largely sufficient to capture all the significant expression variations in a human face (noticed that the human visual system can process 10 to 12 separate images per second [43]).

To perform this operation, a distance measure $D_f(F_i, F_j)$ between face models in frame F_i and frame F_j is defined and computed by exploiting three special feature points:

$$D_f(F_i, F_j) = \frac{1}{3} \sum_{k \in K} ||\rho_i^k - \rho_j^k|| \tag{3.12}$$

where $|| \cdot ||$ is Euclidean distance, ρ_i^k and ρ_j^k are the spatial coordinates of point k on the face in frame F_i and frame F_j and $K = \{$left-eye inner corner, right-eye inner corner, philtrum$\}$ (see these points in Figure 3.3).

For every second, i.e., for every i and j such that $||i - j|| = 1$ second, if $D_f(F_i, F_j)$ is smaller or equal to a threshold T, M frames between F_i and F_j are grabbed. In our experiments, with T equal 8, 10, and 12, the number of frames grabbed M are 10, 11, and 12, respectively. Notice that there are no videos with $D_f(F_i, F_j) > 12$ in our experimental datasets. As to the spatial resolution, each face is analyzed in a resolution of 400×400.

We choose two inner corners and a philtrum due to their stability under different expressions and lighting conditions [32]. Therefore, distances of these points can be considered as a measure of speed of the head movement.

This step helps to convert an input video into a homogeneous series of face instances of the same person in chronological order (see Figure 3.9).

3.3.2 Face Model Reconstruction

In order to reconstruct the face model from a 2D input image, we apply the method from [4]. After building a reference 3D model, this method adapt this reference model to the 2D image through an optimization procedure.

To build the reference 3D model, Algorithm 1 is applied on a training set of 3D images, to construct a normalized mean shape $\overline{S}^{3D}$ which can be considered as a general 3D face model, and the corresponding eigenvectors matrix φ^{3D}, which can be used to transform a given 3D shape into the 3D face model. This normalized mean shape $\overline{S}^{3D}$ and the eigenvectors matrix φ^{3D} are called 3D Point Distribution Model (PDM). Notice that the PDM is built only once and can be applied to different faces in different videos.

By using the PCA, new shapes can be expressed by linear combinations of the training shapes [27], hence the normalized mean shape $\overline{S}^{3D}$ can be deformed (larger eyes, wider chin, longer nose, etc.) to best fits the input faces.

Given a PDM, we now have to approximate it to all instances of faces output of step (A). In order to reconstruct the face model from a 2D input image, we have to project the 3D PDM into 2D space. This could be done through an optimization procedure, which is summarized as Algorithm 2. The main idea is to perform the optimization process on a single instance each time: face pose is estimated based on the generated shape, then based on the new computed pose, the new face shape is re-estimated, and so on, i.e., either shape or pose is estimated each time based on the other information. Thus, step (B) will produce all the information needed to map the set of 2D faces into the corresponding set of 3D face models.

Notice that differently from [4], the ASM points for each 2D face s^{2D} are extracted by using Luxand FaceSDK [34], which able to extract 66 ASM points for each face in still images. Camera intrinsic parameters (f,

Algorithm 1 Compute 3D Point Distribution Model (PDM)

Input: n different shapes of faces $\{s_1^{3D}, s_2^{3D}, \ldots, s_n^{3D}\}$, where $s_i^{3D} = \{x_i^1, y_i^1, z_i^1, x_i^2, y_i^2, z_i^2, \ldots, x_i^d, y_i^d, z_i^d\}$, where d is the number of ASM points and (x_i^k, y_i^k, z_i^k) is the spatial position of point k^{th} on face i^{th}.

Output: A normalized mean shape $\overline{S}^{3D}$ and the corresponding eigenvectors matrix φ^{3D} of the training faces. **Method:** (inspired from [4])

1: Normalize all face shapes: all the points are scaled into [-1, 1]: $S_i^{3D} \leftarrow Normalize3D(s_i^{3D})$, $i = 1, 2, ..., n$.
2: Compute the mean shape: $\overline{S}^{3D} \leftarrow \frac{1}{n}\sum_{i=1}^{n} S_i^{3D}$
3: **repeat**
4: **for each** normalized shape S_i^{3D} **do**
5: Find rotation matrix R_i and translation vector t_i to transform S_i^{3D} into $\overline{S}^{3D}$.
6: $S_i^{3D} \leftarrow R_i(S_i^{3D}) + t_i$.
7: **end for**
8: Re-compute the mean shape: $\overline{S}^{3D} \leftarrow \frac{1}{n}\sum_{i=1}^{n} S_i^{3D}$
9: **until** convergence.
10: Apply Principal Component Analysis (PCA) to all normalized shapes S_i^{3D}, $i = 1, 2, ..., n$ to have the eigenvectors matrix φ^{3D}.

(o_x, o_y)), the rotation matrix R and the translation vector t in equation (3.13) and (3.14), are represented as a single camera projection matrix, and hence can be jointly approximated. They can be decomposed from the camera projection matrix by using the method in chapter 6, section 6.3.2 in [52].

Given the 3D PDM defined exploiting Algorithm 1, we report in Figure 3.10 an example 3D face reconstruction (step B). The left picture shows an example of 2D facial features extraction using Luxand FaceSDK [34]. Algorithm 2 is then applied to the extracted 2D points, and the 3D shape and model are reconstructed, as shown in the right picture.

The accuracy of step (B) is critical for the following step (C) and will be demonstrated in the experiments (Section 4.5).

Algorithm 2 Extract pose and face parameters (from [4])

Input:

- A shape in 2D: $s^{2D} = \{x^1, y^1, x^2, y^2, \ldots, x^d, y^d\}$, where d is the number of ASM points and (x^k, y^k) is the spatial position of point k^{th} on the input face.
- A PDM ($\overline{S}^{3D}$ and φ^{3D}).

Output: The face model p^{3D}, rotation matrix R and translation vector t.
Method:

1: Normalize the face shape: all the points are scaled into [-1, 1]: $S^{2D} \leftarrow Normalize2D(s^{2D})$
2: $p^{3D} \leftarrow 0$.
3: **while** p^{3D}, R, and t do not converge **do**
4: Compute R and t by solving

$$Err = \left|\left|S^{2D} - P(R(\overline{S}^{3D} + \varphi^{3D}p^{3D}) + t)\right|\right|_2 \tag{3.13}$$

using Zhang's method [60], P is the projection transformation. A 3D point $(x_i, y_i, z_i)^T$ is projected by P into 2D space as follows:

$$P\begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix} = \frac{f}{z_i}\begin{pmatrix} x_i \\ y_i \end{pmatrix} + \begin{pmatrix} o_x \\ o_y \end{pmatrix} \tag{3.14}$$

where f is the focal length, and (o_x, o_y) is the principal point location on 2D image.
5: Compute the new face: $S^{*3D} \leftarrow R(\overline{S}^{3D} + \varphi^{3D}p^{3D}) + t$
6: Generate the ideal 3D shape S'^{3D}: $S'^{3D} \leftarrow x$ and y values from S^{2D} and z values from S^{*3D}.
7: Recompute p^{3D} from S'^{3D}:
$p^{3D} = (\varphi^{3D})^T(R^{-1}(S'^{3D} - t) - \overline{S}^{3D})$.
8: **end while**

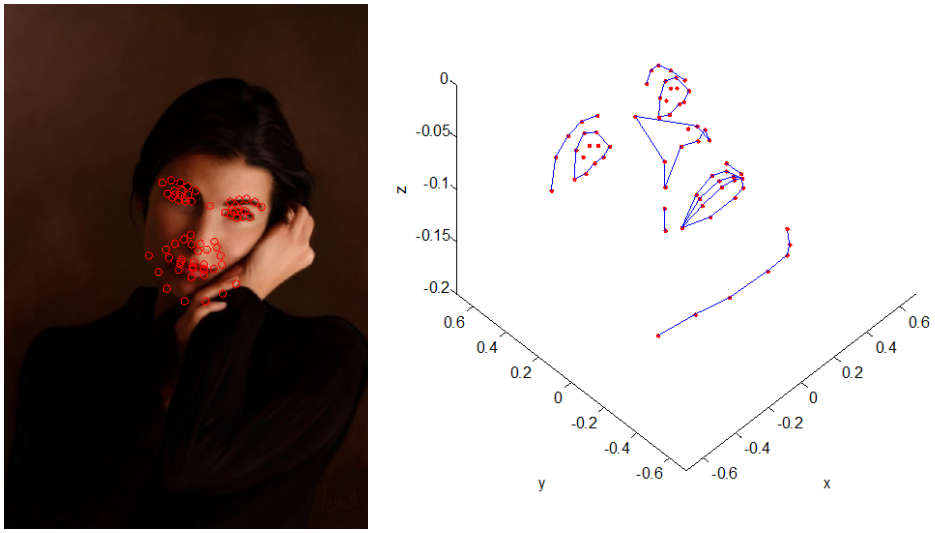


Figure 3.10: An example of step (B): face model reconstruction.

Left picture: the extracted ASM feature points. Right picture: A reconstructed 3D face shape. The computed pose is $(roll = -15.55^o, yaw = -0.45^o, pitch = -2.97^o)$, the translation vector is $t = [-0.0217, -0.0042, -0.0478]^T$, and the focal length $f = 0.0478$.

3.3.3 Computer Generated Character Identification

Step (B) outputs p^{3D}, R and t for each analyzed 2D face representation. Based on this information, step (C) analyzes the evolution of such 3D face model p^{3D} along the video. Thanks to PCA, exploiting the face model p^{3D} allows us to work on a space where the information about the whole face is encoded but also compressed in a limited number of coefficients, which contain the most discriminative components of the signal.

Figure 3.11 shows various 3D shapes generated from different values of the first component of the face model of the mean shape $\overline{S}^{3D}$ (note that $\overline{S}^{3D}$ has $p^{3D} = \overrightarrow{0}$). Hence, differences of a face in an animation can be analyzed based on the first components instead of all feature points.

Furthermore, we study the evolution of the model not only during the whole video, but also on non-overlapping windows (see Figure **??**) that can highlight particular animations and expressions of the represented character (e.g., two or more different animations in a same video can be better

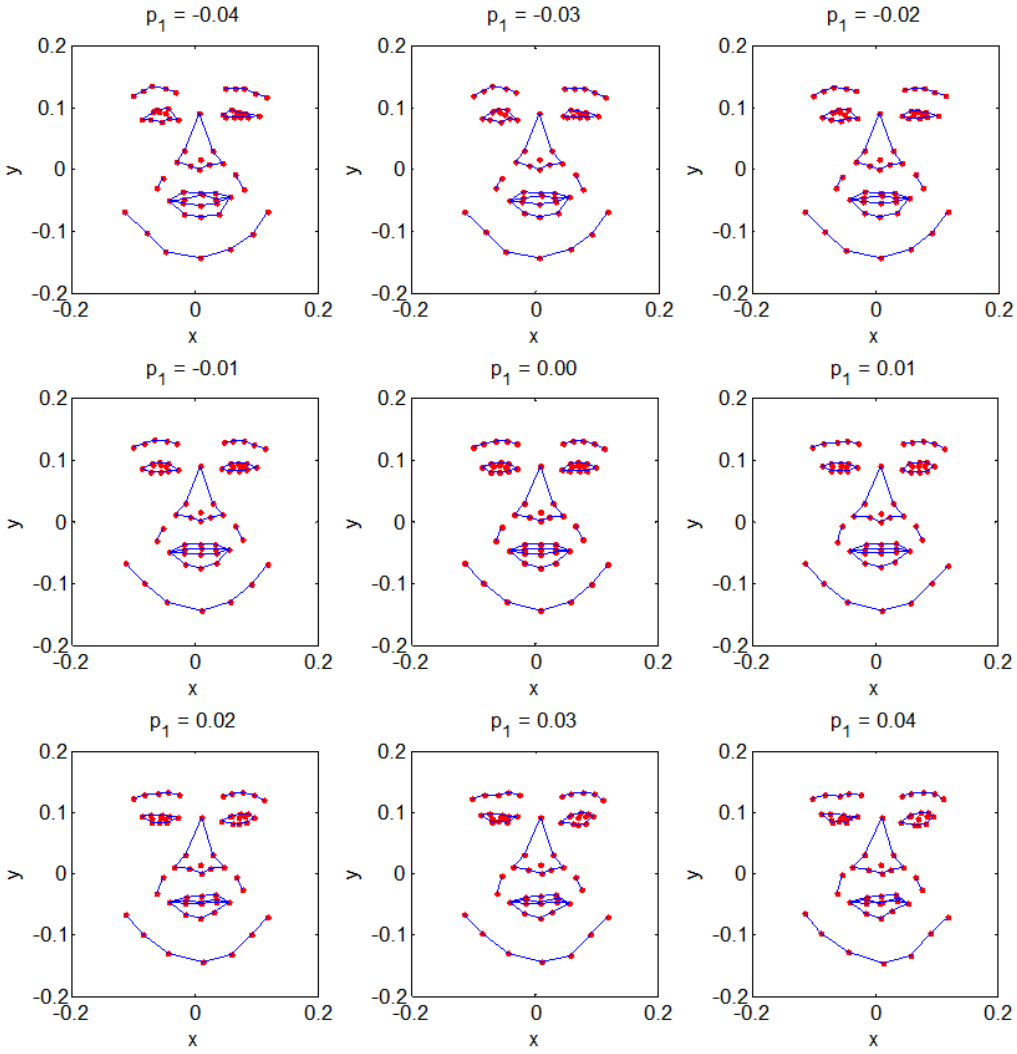


Figure 3.11: The role of face models.

Various 3D shapes generated from different values of the first component p_1 of the face model p^{3D} of the mean shape $\overline{S}^{3D}$. Notice that small differences in p_1 lead to visible differences in the 3D face shapes.

analyzed using small windows).

Let us assume each window W_j of length of l, and a face model p_i^{3D} extracted and encoded using PCA as $p_i^{3D} = (p_i^1, p_i^2, ..., p_i^m)$ for each frame i in W_j. Hence each window W_j is encoded as a matrix $l \times m$:

$$\begin{pmatrix} p_1^1 & p_1^2 & \cdots & p_1^m \\ p_2^1 & p_2^2 & \cdots & p_2^m \\ \vdots & \vdots & \ddots & \vdots \\ p_l^1 & p_l^2 & \cdots & p_l^m \end{pmatrix} \tag{3.15}$$

Notice that $m = 3d$, where d is the number of feature points in Algorithm 2. In order to study how the 3D face model evolves in W_j and thus to measure the complexity of the geometric distortion during the animation, we extract the following properties:

1. The mean values μ_j^c, $c \leq m$, in W_j.

$$\mu_j^c = mean(||p_2^c - p_1^c||, ||p_3^c - p_1^c||, ..., ||p_l^c - p_1^c||) \tag{3.16}$$

2. The standard deviations of differences σ_j^c, $c \leq m$, in W_j.

$$\sigma_j^c = sdv(||p_2^c - p_1^c||, ||p_3^c - p_1^c||, ..., ||p_l^c - p_1^c||) \tag{3.17}$$

3. The average lengths of the trajectories τ_j^c, $c \leq m$, in the first components.

$$\tau_j^c = \frac{1}{l-1} \sum_{i=2}^{l} ||p_i^c - p_{i-1}^c|| \tag{3.18}$$

4. The average length of the combination of trajectories T_j^c, $c \leq m$, of the first components (note that $T_j^1 = \tau_j^1$).

$$T_j^c = \frac{1}{l-1} \sum_{i=2}^{l} \sqrt{\sum_{k=1}^{c} ||p_i^k - p_{i-1}^k||^2} \tag{3.19}$$

Here, μ_j^c contains the mean of the differences between the models and σ_j^c measures the spread of the models (from μ_j^c), while τ_j^c and T_j^c evaluate the amount of changes of the models over time. Considering this problem as the analysis of a point moving in the space, Equation (3.16) and Equation (3.17) represent the mean position and variance, while Equation (3.18) and Equation (3.19) describe the length of the path of the moving point (see Figure 3.12 for a graphical explanation).

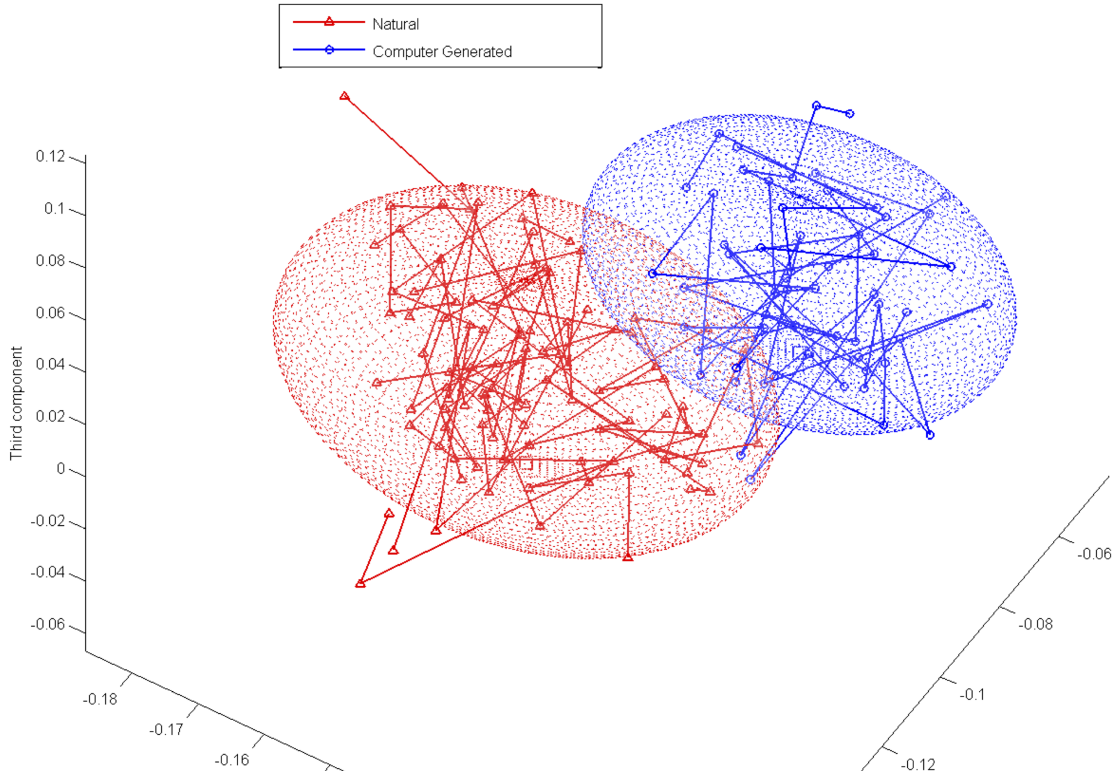


Figure 3.12: Graphical explanation of the chosen properties.

Each face is transformed as a point in PCA space, hence the analysis of the evolution of a face over time can be done by study the trajectory of the corresponding moving point in PCA space. Giving the correspoding 'sphere' where the points are moving, Equation (3.16) and Equation (3.17) correspond to its center and its size, while Equation (3.18) and Equation (3.19) are related with the length of the trajectory.

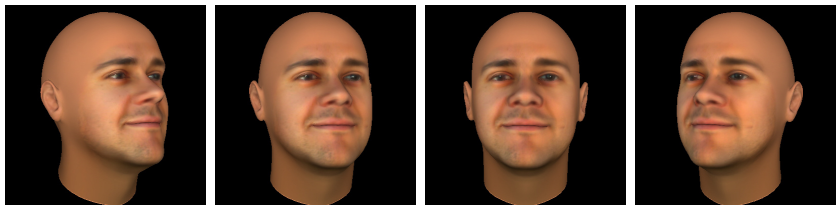
We have then a set of features extracted from each window W_j, and

another set of features extracted in the same way over the whole video, i.e., $l = N$, where N is the number of frames. The features computed on the whole video are fundamental for videos containing a main single expression, while the average computed on sets corresponding to W_j, $\forall j$ is critical when we deal with more complicated videos containing complex animations. Both properties are then important to take final decision. Shown in Figure 3.13 is an example on videos of a CG characters (panel (a)) and a real human (panel (b)) changing the animations from talking to smiling. Panel (c) shows the mean values over the whole video, in this case, the average difference on the first component between the CG and natural video is 0.0019, and panel (d) shows the same mean values, but on non-overlapping windows, and in this case, the difference is 0.0038.

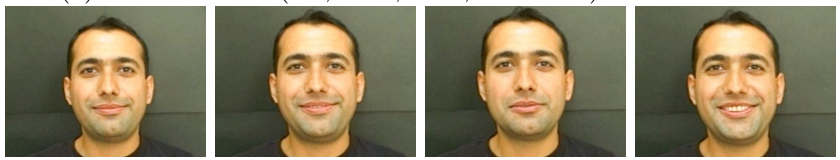
The last step is a binary classifier that differentiates between CG and natural face animations. To this purpose we use a support vector machine (SVM) fed with the above features. A polynomial kernel is used with Sequential Minimal Optimization (SMO) method.

3.4 Discussions

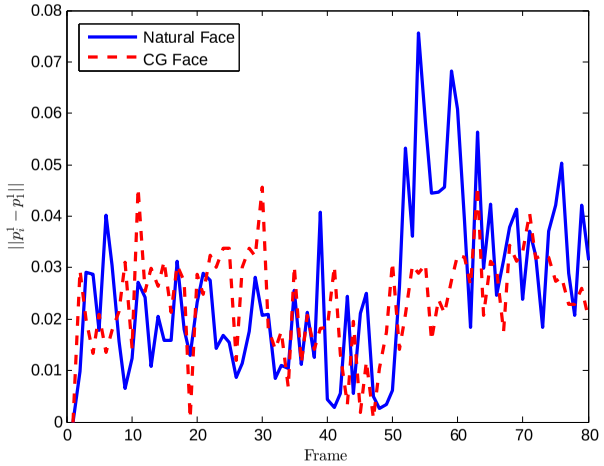
In the previous sections, we presented novel ways to tackle the problem of differentiating between computer generated and photographic human faces. Based on the characteristics of these methods, ModelMethod, which analyses the evolution of face models over time, seems to be the best solution for most of the cases. However, there are situations that using AsymMethod and ExpressMethod is more suitable. Hence, in this section, we give some discussions on advantages and disadvantages of each method in order to complete our solutions to this problem. This could be considered as a strategy for differentiate between CG and natural human faces in multimedia data.



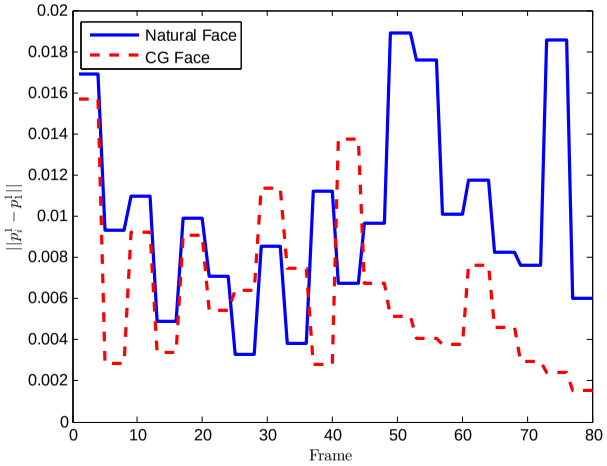
(a) A CG video. (1^{st}, 30^{th}, 50^{th}, and 80^{th}) frames are shown.



(b) A natural video. (1^{st}, 30^{th}, 50^{th}, and 80^{th}) frames are shown.



(c) Analylis of video sequences (a) and (b) with $l = N$, $c = 1$



(d) Analylis of video sequences (a) and (b) with $l = 4$, $c = 1$

Figure 3.13: Example of step (C) of the Modelmethod.

The first 80 frames of a CG video (a) and a natural video (b) were extracted and analyzed. Panels (c) and (d) show the values of $||p_i^1 - p_1^1||$, analyzed with $l = N$ and $l = 4$, respectively. Using $l = N$, $|\mu^1_{l,natural} - \mu^1_{l,CG}| = 1.9 \times 10^{-3}$, while $|\mu^1_{4,natural} - \mu^1_{4,CG}| = 3.8 \times 10^{-3}$.

- **AsymMethod:**
  - **Advantages:** this method is easy to implement and can be combined with other state-of-the-art methods on still images, or applied to frontal faces in ExpressMethod and ModelMethod.
  - **Disadvantages:** this approach only works only with frontal faces so far, since the normalization step can only normalize rotated faces, but not the turned ones; moreover it is very sensitive to the normalization step. Although this is a geometric-based method, the $D-face$ is computed based on the intensity of the face, hence it is dependent on the resolution and quality of the input image.
  - **Best situations:** this method can work on a single photo, especially when combined with other natural images statistic method, e.g., the method in [36].
- **ExpressMethod:**
  - **Advantages:** this method can work not only on a video but also on a set of images (of a same person on a same expression). Furthermore, there is no machine learning required to used in order to detect the computer generated character.
  - **Disadvantages:** The major drawback of this method is in the fact that differences in facial expressions are difficult to be modeled and may be sensitive to the limited set of expressions available on the video. For instance, the displacement of the points located on the lips in a sad expression is not comparable to the displacement of the same points in a happy expression, thus requiring the analysis of different sets of points for each single expression. Moreover, this requires the availability of multiple instances of the same (or similar) expression, which is not often the case in

real videos. Finally, since the method works in 2D, it is able to manage only on frames where the face appears in nearly frontal view, which again reduces the information that can be extracted from a video clip.

- **Best situations:** on the set of images of a same person on a same expression, especially on happiness expression since most of the expression recognition methods work well on happiness.

- **ModelMethod:**

  - **Advantages:** ModelMethod is developed from limitations of the previous approaches, hence it covers a larger variety of situations and can model in a more dependable way the characters' behaviours, even when the subject is moving and turning the face. The proposed model better fits the analysis of human faces, taking into account their variety, deformability, diversity of expressions, different poses, as well as the external factors such as illumination conditions and framing.
  - **Disadvantages:** ModelMethod requires machine learning systems and has a complex installation. It also requires a very good PDM, which must be computed from a large dataset of 3D facial images.

To summarize, ModelMethod could be applied to differentiate between CG and natural human faces in a general way. However, when working with a set of faces without any knowledge about the chronological order of the faces, ExpressMethod could be used instead of ModelMethod. Furthermore, when lacking of training datasets or working with video of happiness faces, ExpressMethod can be a good solution. Finally, only AsymMethod can deal with the problem in still images, hence fusing it with other method

can significantly increase the performance of the discrimination method. Table 3.3 summarizes all of the characteristics of the proposed methods.

Table 3.3: Summary of the proposed methods.

	AsymMethod	ExpressMethod	ModelMethod
Implementation level	1 ms/image	1 s/frame	30 s/frame
Works on a single photo	Yes	No	No
Works on videos	No	Yes	Yes
Independent on resolution and format	No	Yes	Yes
Poses	0 degree	Up to 5 degrees	Up to 30 degrees
Require ML	No	No	Yes
Expression independent	No	No	Yes
Works with partial of a face	No	No	Yes
Times constraint	No	No	Yes

Notice that with the implementation level, we measured the computational from experiments. With a PC of CPU Core 2 3.06GHz, 8GB RAM, running Windows 7 and the experiments were performed by Matlab 2010b, AsymMethod requires less than 1 millisecond to compute asymmetry level of an input image, ExpressMethod takes around 1 second on each image instances after the ASM points were extracted, while ModelMethod requires in average 30s from each frame for reconstructing 3D model and analyzing the evolution. The details of all experiments will be introducted in the next chapter.

Chapter 4

Experimental Results

In this chapter, we present experimental results on the proposed methods, which show that using these methods, CG faces can be detected from both still images and videos in reliable and accurate ways. The organization of the collected datasets and evaluation metrics are also introduced.

However beautiful the strategy, you should occasionally look at the results.

Winston Churchill

Acknowledgements

- Portions of the research in this chapter use the CASIA- 3D FaceV1 collected by the Chinese Academy of Sciences Institute of Automation (CASIA).

4.1 Datasets

In order to evaluate the performance of the proposed methods, various sources have been collected and created to have a complete evaluation. The datasets are grouped into two categories:

- Benchmark datasets: these are the common datasets which are used in other studies.
- Collected datasets: we also created and collected other images and videos for our simulations on the situations which are not available in the benchmark datasets, e.g., faces with different poses or photorealistic human faces in high quality.

In the next sections, details of these datasets are introduced.

4.1.1 Benchmark Datasets

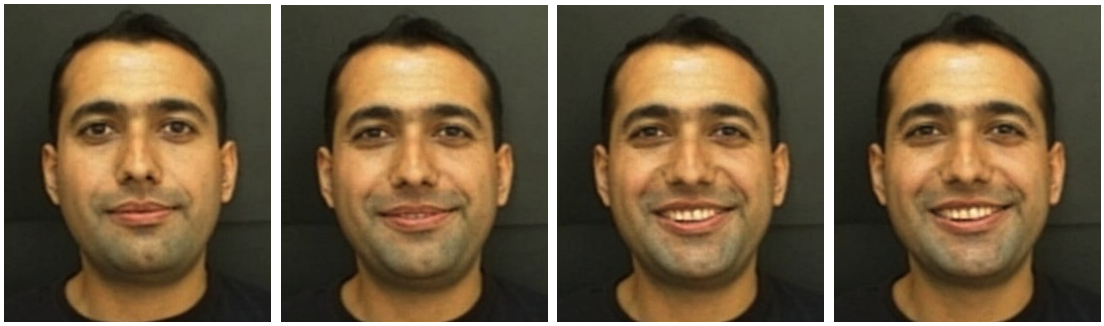
We used several common datasets in facial analysis together with multimedia forensics for our evaluation:

4.1.1.1 BUHMAP-DB

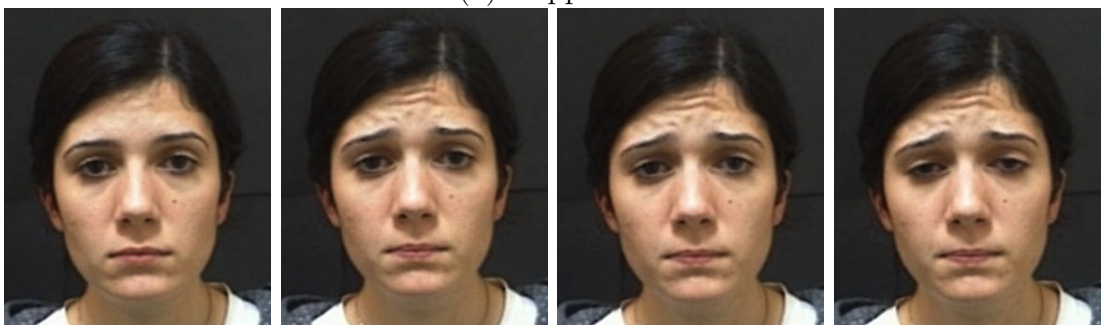
Boğaziçi University Head Motion Analysis Project Database (BUHMAP-DB) [1] contains 440 videos of 11 people (6 female, 5 male) performing 5 repetitions on 8 different gestures. BUHMAP-DB is used in experiments of both Method 2 and Method 3. Shown in Figure 4.1 are two examples of the faces extracted from BUHMAP-DB.

4.1.1.2 JAFFE

The Japanese Female Facial Expression Database (JAFFE)[35] contains 213 images of 7 facial expressions posed by 10 Japanese female models.



(a) Happiness



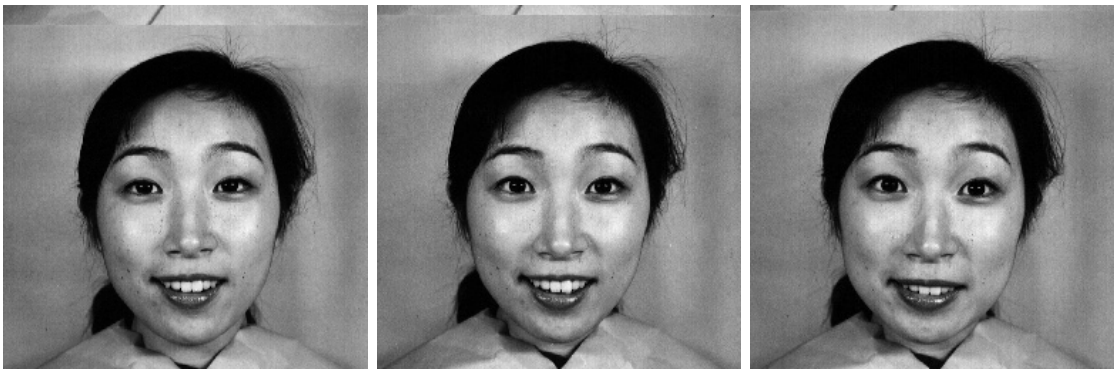
(b) Sadness

Figure 4.1: Examples of extracted faces from BUHMAP-DB.

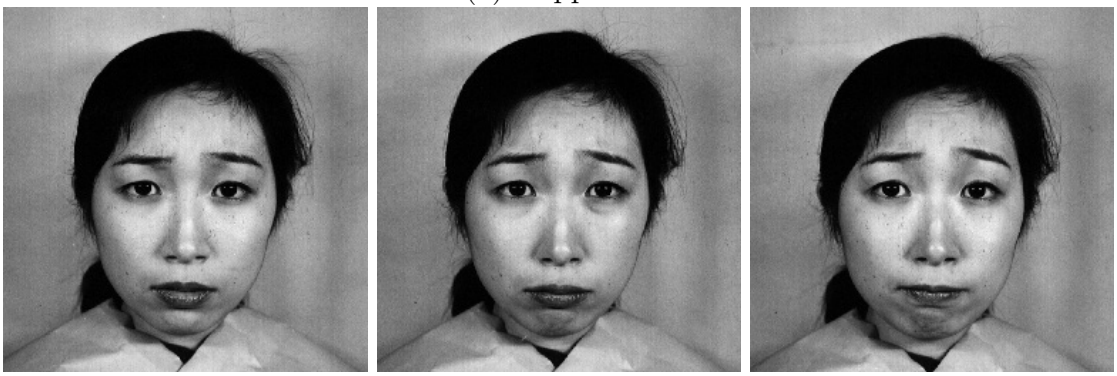
This dataset is used in both Method 2 and Method 3. Shown in Figure 4.2 are some examples of the faces extracted from this dataset.

4.1.1.3 CASIA-3D FaceV1

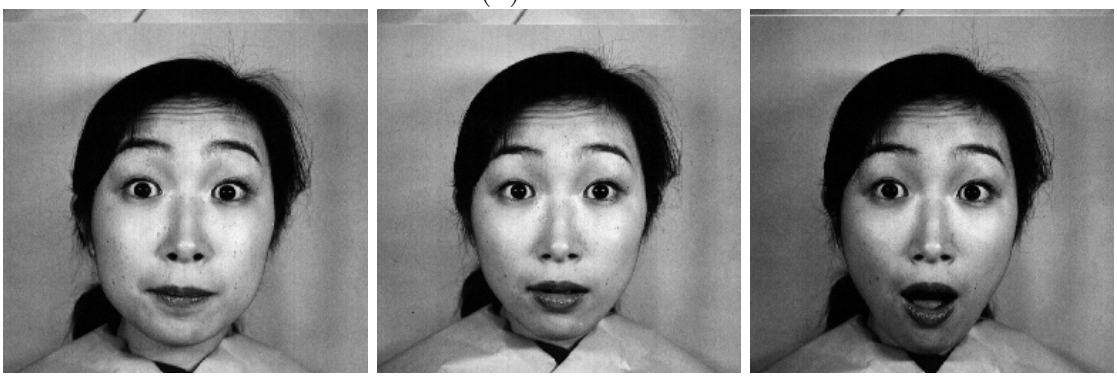
Casia-3D FaceV1 [51] consists of 4624 scans of 123 people using the non-contact 3D digitizer. For each person, they collected 37 - 38 images, both in 3D and 2D, in different poses, expressions, and lighting conditions. The 3D-PDM reference model of Method 3 is built based on this dataset. Shown in Figure 4.3 are some examples of the faces from this dataset.



(a) Happiness



(b) Sadness



(c) Surprised

Figure 4.2: Examples of faces from JAFFE.

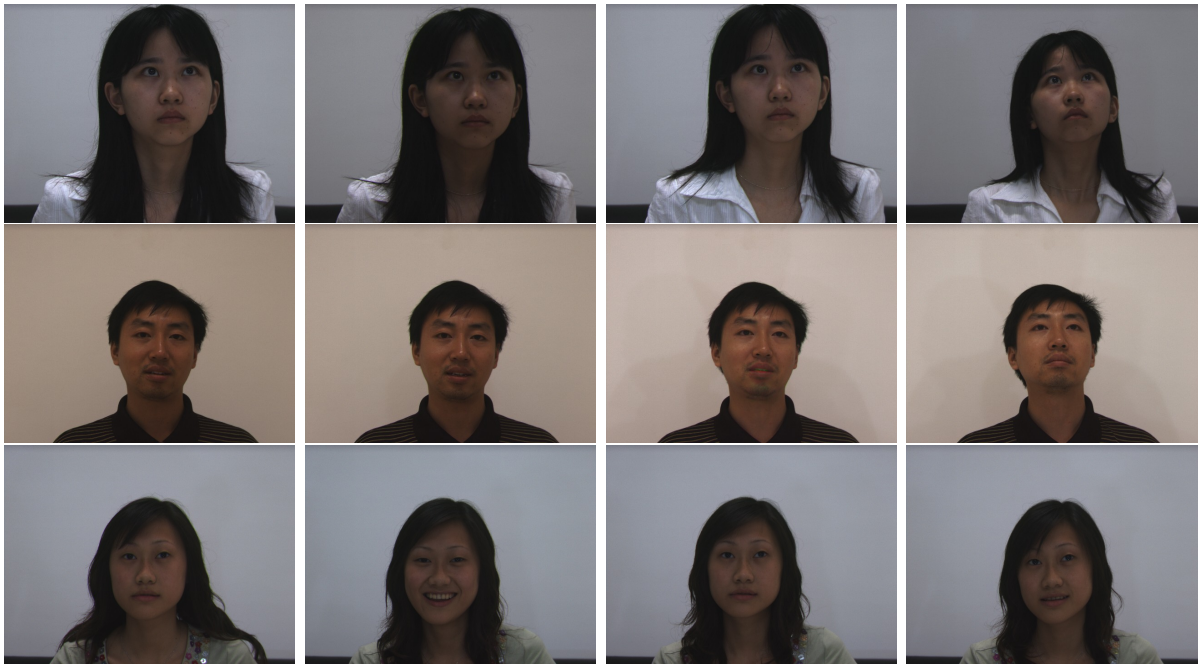


Figure 4.3: Sample images from CASIA-3D FaceV1 dataset.

4.1.1.4 Star Trek

We collected *Star Trek Aurora*[1] movie, a fully-animated product, and *Star Trek Odyssey*, a live action movie from *Star Trek: Hidden Frontier* series[2] in order to test Method 2. In Star Trek Aurora, two graphics applications, namely Poser and Cinema 4D, are used to create the entire 3D world and characters while *Star Trek Odyssey* is a normal movies perform by real actress. Some examples of these two movies are shown in Figure 4.4

4.1.2 Collected Datasets

We created and collected other images and videos for our simulations on the situations which are not available in the benchmark datasets, e.g., faces with different poses or photorealistic human faces in high quality. In the following sections, details of these datasets are presented.

[1]http://auroratrek.com
[2]http://www.hiddenfrontier.com

(a) Star Trek Aurora, a fully-animated movie.

(b) Star Trek Odyssey, a live action movie.

Figure 4.4: Examples of human happiness faces extracted from Star Trek movies.

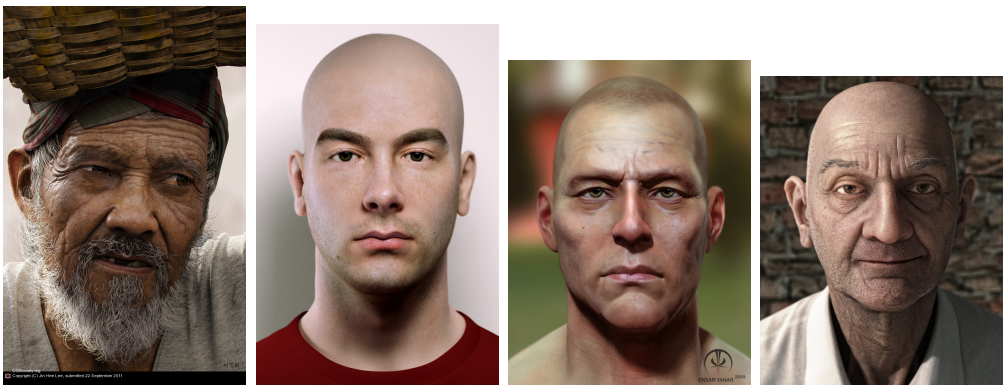
4.1.2.1 D1. Synthetic Human Faces

We have collected the computer generated images from the *Society of Digital Artist*[3] and downloaded football player face images from the database of *Faces for Pro Evolution Soccer 2012 (PES 2012)*[4]. All of the computer generated images are confirmed that they are purely created by computer. We have also collected other images from Karolinska[5] database, which contains hundreds of frontal face images. For the natural images, real people and football players images were collected from various sources on the internet. From these images, we grouped them into two sub-datasets:

- Dataset D1.1 contains very realistic CG images, which are almost undetectable by human, together with other natural images;
- Dataset D1.2 contains more images of the football players from PES 2012, and real faces as described above.

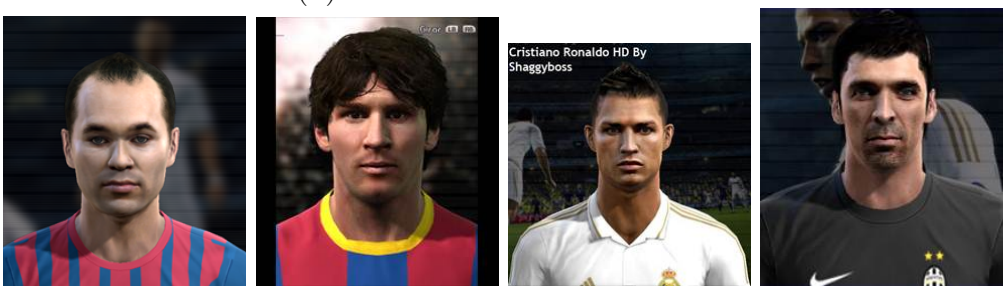
The number of images from these two subsets are reported in Table 4.1 while some examples are shown in Figure 4.5.

[3]http://CGSociety.org
[4]http://www.pesfaces.co.uk/
[5]http://webscript.princeton.edu/~tlab/databases/database-2-karolinska-dataset/



(a) CG faces of Dataset D1.1, downloaded from CGSociety.org.

(b) Real faces of Dataset D1.1



(c) CG faces of Dataset D1.2, downloaded from PES 2012 faces database

(d) Real faces of Dataset D1.2.

Figure 4.5: Examples of images in dataset D1.

Table 4.1: Number of images in Dataset D1

	Computer Generated	Photographics
Dataset D1.1	40	40
Dataset D1.2	200	200

4.1.2.2 D2. Synthetic Expression

The experiments of Method 2 are performed on expressions, hence starting from the 11 people of BUHMAP-DB, we created 11 CG characters by using FaceGen [24] and morphed all of them into both happy and sad faces. FaceGen is a powerful tool which can be used in building complex face structures from one to three images. In our case, we pass a 'neutral' image to FaceGen in order to build the face structure, then we use Morph options to generate happiness and sadness expressions on the new generated face. Thus, we obtained 110 sets of happy and sad faces, where each model has 55 sets corresponding to happiness and 55 sets corresponding to sadness. Shown in Figure 4.6 are two examples of the CG versions and the original faces from BUHMAP-DB.

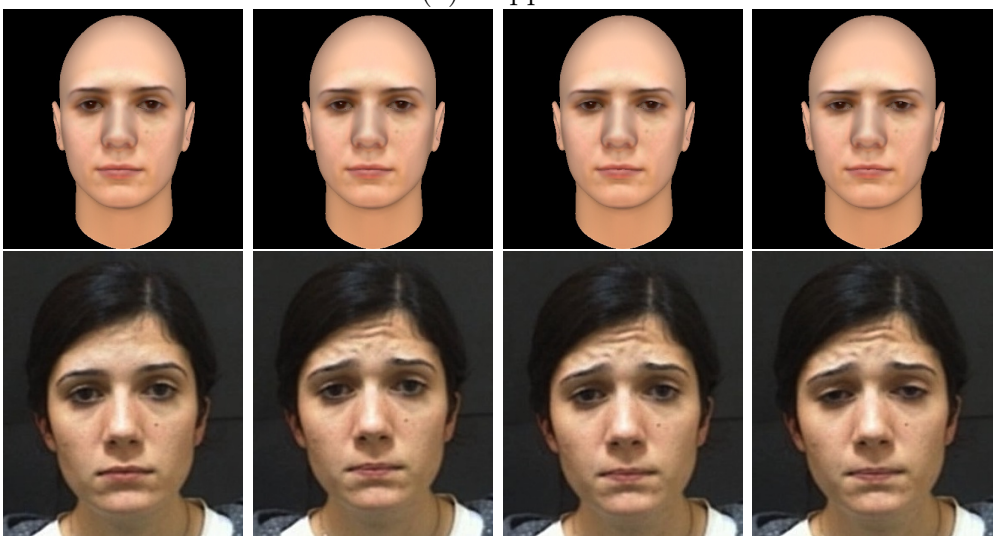
The same process is applied on JAFFE datasets to have 6 models with all types of expressions (see examples in Figure 4.7).

4.1.2.3 D3. Synthetic Animations

For this dataset, we created face models in 3D then create synthetic animations based on those models. We also collected videos from Internet for the most realistic characters. These videos are used in the experiments of Method 3.

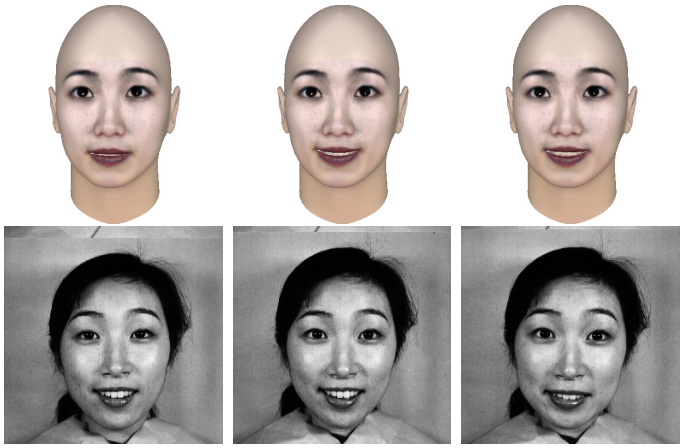
- **Set D3.1**: 10 face models in 3D, rotated with different poses and stored in 2D to have 60 images in total. See examples in Figure 4.8.
- **Set D3.2**: 60 sets of synthetic characters performing different animations, (5 males and 5 females with 6 animations for each person).

(a) Happiness

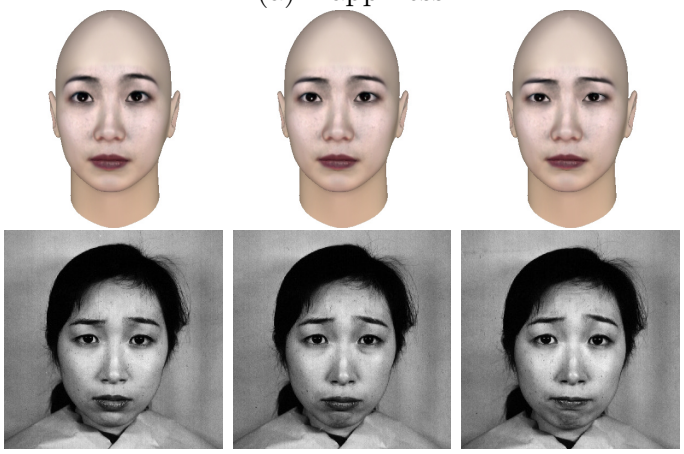


(b) Sadness

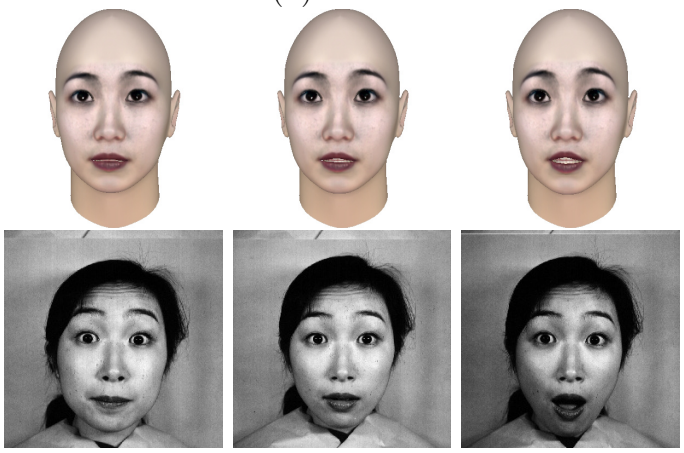
Figure 4.6: Examples of faces from BUHMAP-DB and the corresponding CG faces generated via FaceGen.



(a) Happiness



(b) Sadness



(c) Surprised

Figure 4.7: Examples of faces from JAFFE and the corresponding CG faces generated via FaceGen.

Figure 4.8: Examples of images from dataset D3.1.

For each set, 100 images has been created as a video of 10 seconds with the frame rate of 10 fps. Figure 4.9 shows some example of the images, exacted from each second. The model are built based on real people from BUHMAP-DB.

- **Set D3.3**: We collected videos from Internet for the most realistic characters (see the list in Appendix A) and extracted 24 short animations from those videos for our experiments, each animation last from 6 to 10 seconds. Shown in Figure 4.10 are some examples of this set.

Datasets information are summarized in Table 4.2, while in the next sections, details of the experiments on these datasets are introduced.

4.2 Evaluation Metrics

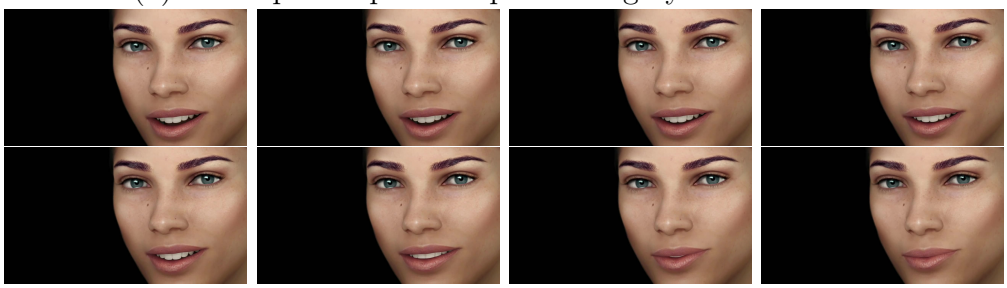
In this section, we present the common metrics that used in most of the work in this field, which is based on the typical precision/recall and confusion matrix. In particular, the current works mainly consider the problem

(a) Happiness.

(a) Sadness.

Figure 4.9: Examples of frames extracted from dataset D3.2.

(a) A complex expression performing by a real female.



(b) OnLive's MOVA Geni4, a CG female character.

(c) Paul Ekman talks about his work on the human face.

(d) A CG character from Activision R&D realtime demo.

Figure 4.10: Samples of real videos from dataset D3.

Table 4.2: Summary of datasets used in this doctoral thesis.

Dataset	CG Images	Natural Images	CG Videos	Natural Videos
BUHMAP-DB				440 short videos of 11 people perform 5 times repetitive 8 animations.
JAFFE		213 images of female in 6 expressions		
CASIA-3D FaceV1		4624 from non-contact 3D digitizer		
StarTrek			1 full movie	1 full movie
D1.1 D1.2	40 200	40 200		
D2	116 sets[a]	116 sets		
D3.1 D3.2 D3.3	60 in different poses		 60 from 10 people 24	 60 from 10 models 24 highly realistic

[a]110 sets extracted from BUHMAP-DB videos, 6 sets from JAFFE.

of distinguishing between photorealistic CG multimedia data and the natural ones. This is a typical binary classification problem. The performance of a binary classifier can be measured by a classification confusion matrix at an operating point of the classifier as follows:

$$\begin{bmatrix} p(C = positive|L = positive) & p(C = negative|L = positive) \\ p(C = positive|L = negative) & p(C = negative|L = negative) \end{bmatrix} \tag{4.1}$$

where the two classes are respectively identified as positive and negative while C and L respectively represent the assigned label (by the classifier) and the true label. The probability $p(C = positive|L = negative)$ is known as false positive rate, and $p(C = negative|L = positive)$ false negative rate.

The averaged classification accuracy can be computed as

$$\frac{p(C = positive|L = positive) + p(C = negative|L = negative)}{2} \tag{4.2}$$

The operating point of a classifier can be adjusted by shifting the decision boundary or threshold values which in turn adjust the balance of the false positive and false negative rates. Equal error rate is often used for evaluating a biometric system [40]. Equal error rate refers to the false positive rate or the false negative rate of a classifier when it functions at an operating point where the two rates are equal.

To have a deeper analysis on this type of result, the area under curve (AoC) of the ROC curve of true positive and true negative are often analyzed in recent works.

4.3 Results of Experiments on AsymMethod

In order to evaluate AsymMethod, we performed three experiments on dataset D1 and compared the results with the state-of-the-art method in [36], [15], and [8].

In our first experiment, we analyze the proposed approach using only asymmetry information achieving 67.5% of accuracy on dataset D1.1 and 89.25% on dataset D1.2. Shown in Figures 4.11 and 4.12 are the ROC chart of False Positive and True Positive rates on Dataset 1 and 2, respectively, while in Tables 4.3 and 4.4 corresponding confusion matrices are reported. These results show that geometry information, in this case the asymmetry of human faces, can be effectively used to discriminate computer generated from the natural faces.

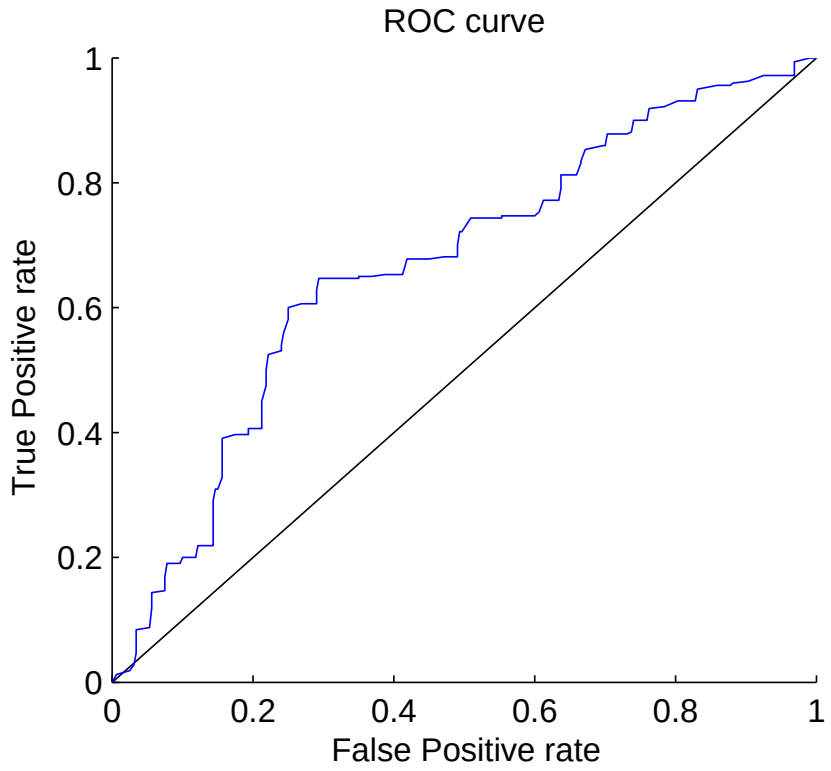


Figure 4.11: ROC curve of AsymMethod on dataset D1.1.

Table 4.3: Confusion matrix on dataset D1.1.

	Computer Generated	Photographics
CG	0.75	0.25
Photographics	0.4	0.6

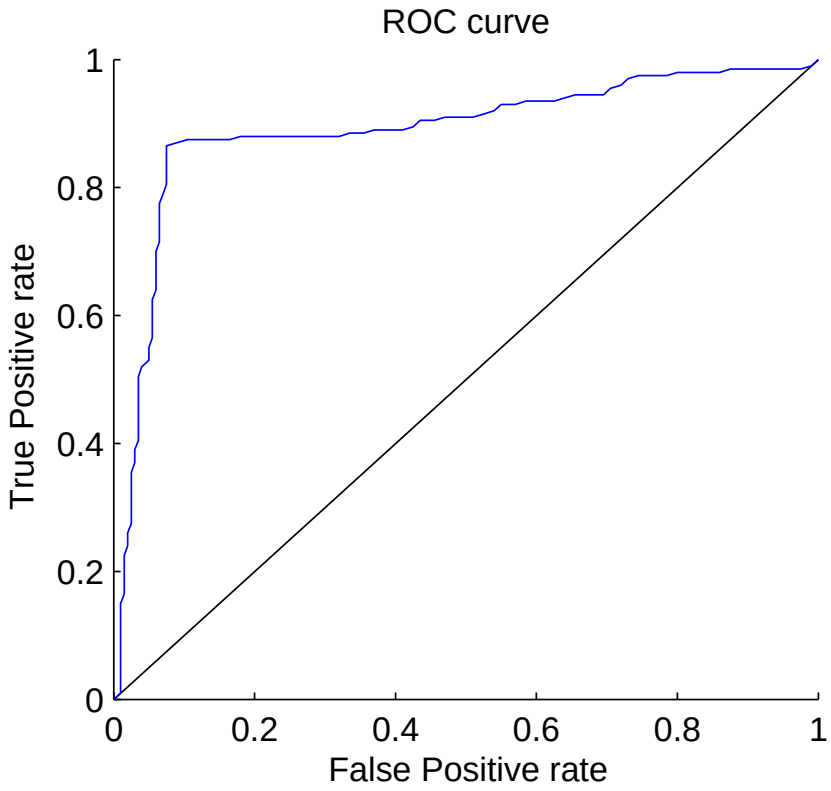


Figure 4.12: ROC curve of AsymMethod on dataset D1.2.

Table 4.4: Confusion matrix on dataset D1.2.

	Computer Generated	Photographics
CG	0.92	0.08
Photographics	0.135	0.865

In the second experiment, we compared our method with three state-of-the-art approaches, namely, [36], [15], and [8]. Here, we consider asymmetry information as a feature, and then use Support Vector Machine (SVM) for training and solving the binary classification problem. Shown in Figure 4.13 are results of comparing these methods using leave-one-out (LLO) cross validation method. It can be noticed that on the challenging dataset D1.1, the proposed approach achieves the best performances, while on dataset D1.2, there is not much difference among all approaches.

In the last experiment, we use asymmetry information as an additional

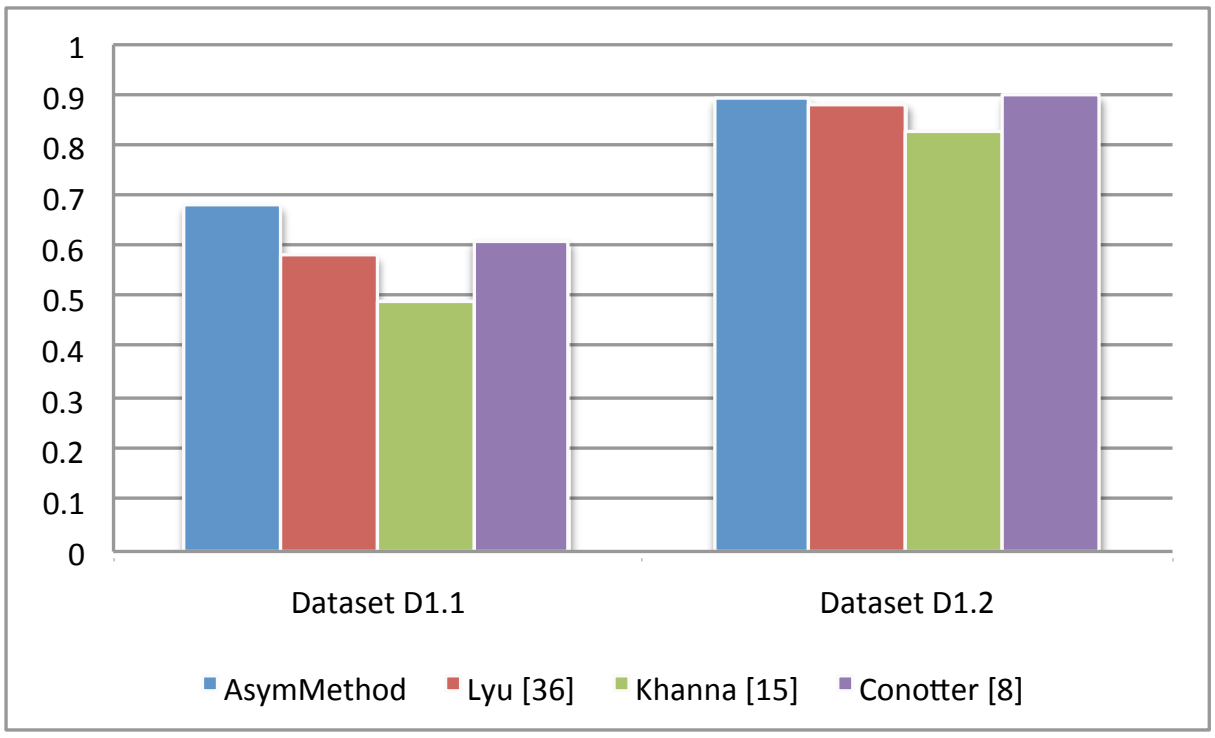


Figure 4.13: Performance of AsymMethod vs. SoA approaches.

.

Comparison of results of AsymMethod with [36], [15], and [8] on dataset D1.

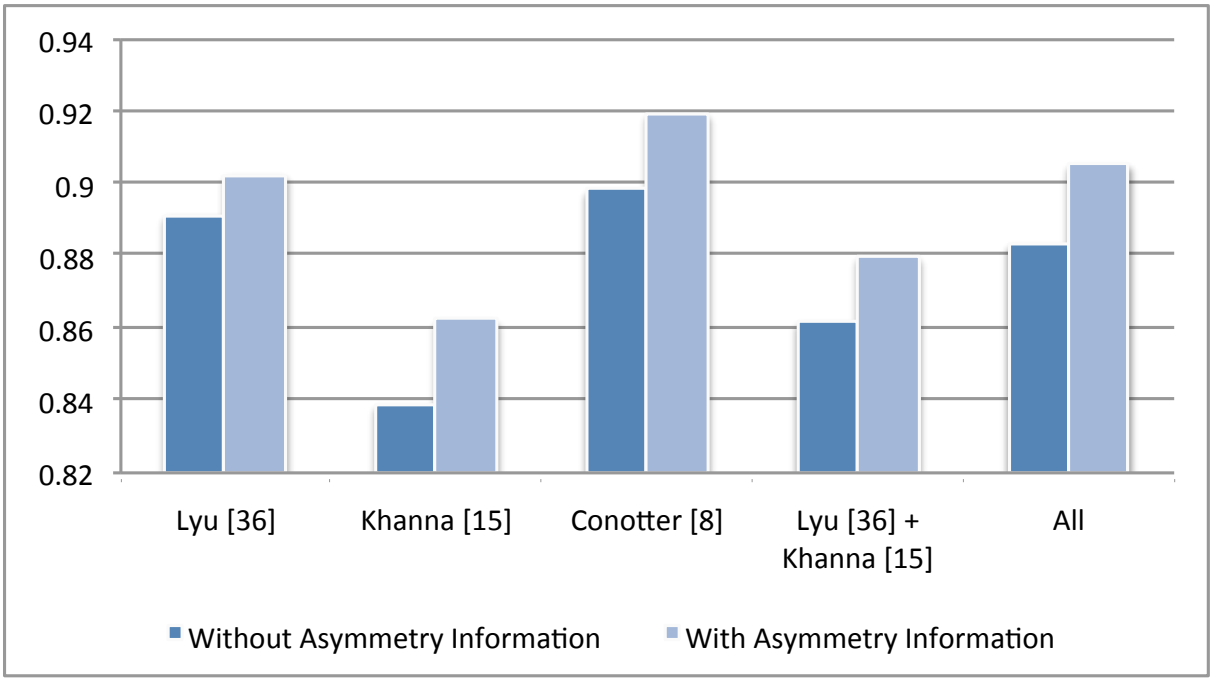


Figure 4.14: Results on the fusion of approaches on dataset D1.1

feature to [36], [15], and [8] and compare results using SVM binary classification (LLO validation). Figure 4.14 and 4.15 show results on dataset D1.1 and dataset D1.2, respectively. Performances of state-of-the-art approaches increase on average by 16.25% on the more challenging dataset D1.1 when fusing their features with the proposed asymmetry features.

These experimental results confirmed that AsymMethod can be used as a stand alone method or in combination with other information to improve state-of-the-art techniques on the problem of discrimination between CG and natural frontal human faces. In the next section, experiments on ExpressMethod is reported.

4.4 Results of Experiments on ExpressMethod

In order to evaluate ExpressMethod, several experiments were performed on BUHMAP-DB, JAFFE, Star Trek, and D2 datasets (see Section 4.1 for

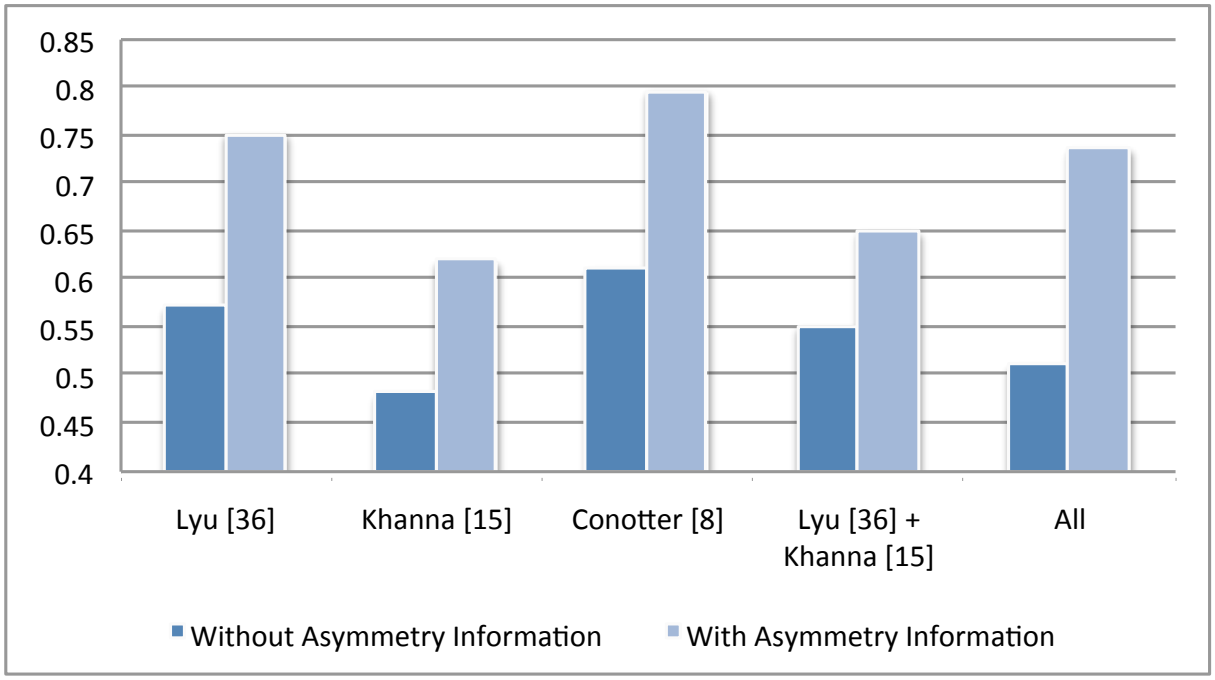


Figure 4.15: Results on the fusion of approaches on dataset D1.2

details).

The goal of the first experiment is to analyse the differences from CG models with the natural faces in order to confirm the idea of the proposed method. The analysis is performed as follows: for each video sequence in BUHMAP-DB, 10 frames are uniformly extracted and similarly for each CG model in D2, 10 images are selected. Then, the sets of images are analysed and the corresponding *Expression Variation Values* computed as described in Section 3.2.5. In this case since the expressions are already known, we implement the method from step C. In this step, we use Microsoft Face SDK [44] to extract the ASM models. Finally, we apply step D and E to get the results.

Shown in Figure 4.16 are EVV_1 values computed on the 55 sets of CG and the 55 sets of natural happy faces. These values are well separated between CG and natural. There is only one miss classification using the threshold $\tau_1 = 0.45$. The accuracy, therefore, is 99% (equals 109/110).

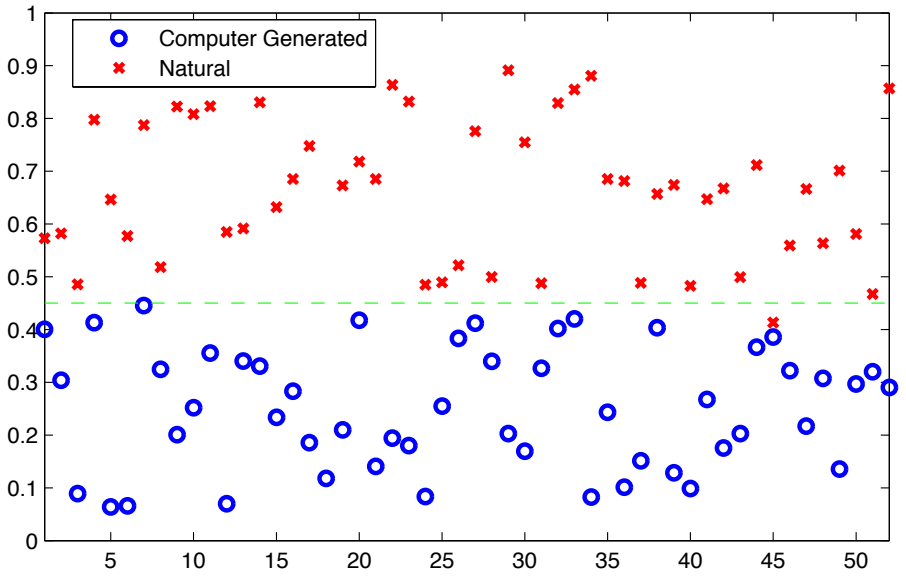


Figure 4.16: Facial Expression Values computed on happiness expression.

The threshold value τ_1 is 0.45. The separation between CG and natural EVV_1 is clearly visualized with only one miss classification.

On sadness expression, the result is even better, with 100% of accuracy using the threshold $\tau_2 = 0.6$. The EVV_2 values for CG and natural characters are perfectly separated, as shown in Figure 4.17.

Our second experiment is performed on the JAFFE dataset, which contains all six expressions. Also in this case we used FaceGen [24] to create the CG models reproducing the JAFFE models (see Figure 4.7 for some examples). For each model in this dataset (dataset D2, in particular), we reproduced all 6 expressions. Therefore, we perform the second test on 120 sets of images, 60 sets of CG and 60 sets of JAFEE real faces. The complete proposed approach described in Section 3.2 is applied as a classification approach on these sets.

Shown in Figure 4.18 is the average EVV_ξ for each expression ($\xi = 1, ..., 6$). The inner blue boundary represents the EVV_ξ computed from CG sets of images, and the outer red boundary represents the natural

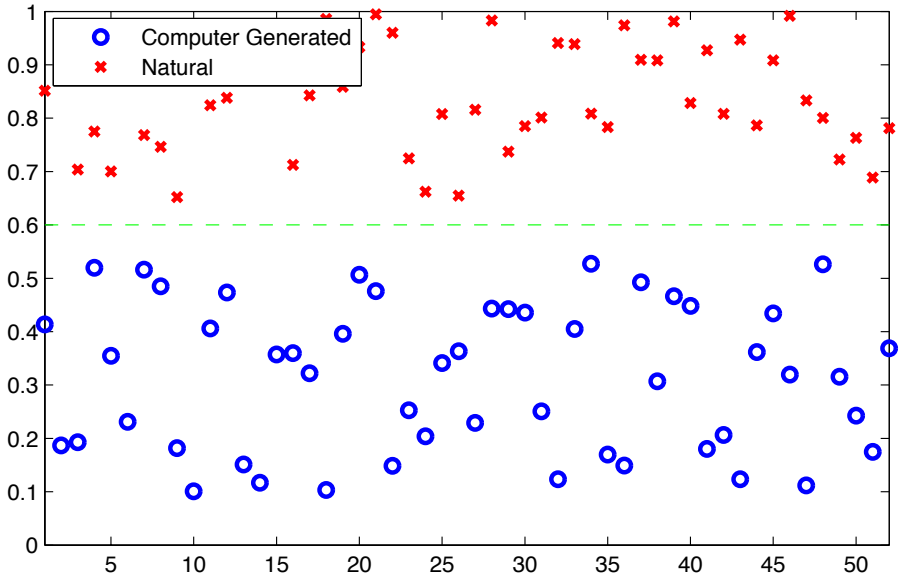


Figure 4.17: Facial Expression Values computed on sadness expression.

The threshold value τ_2 is 0.6. CG and natural EVV_2 are clearly separated.

EVV_ξ. Results show that CG and natural *Expression Variation Values* can be differentiated by using and comparing with a set of thresholds τ_ξ, visualized by the green boundary. The classification performance of this experiment is in average 96.67%. Details for each expression are reported in the confusion matrices, Table 4.5.

The last experiment is performed by comparing two movies in Star Trek datasets. We extracted 4 female characters in each movie and selected frames that contain happy expression of those characters. Happy faces are then confirmed by using Rosa application [45]. Some examples of two characters in happy emotion are shown in Figure 4.4. Finally, EVVs are computed and compared. Using the same threshold as in the first experiment ($\tau_1 = 0.45$), all EVV_1 calculated for the 4 characters of Star Trek Aurora are smaller than τ_1 while all of the EVV_1 from Star Trek Odyssey are over τ_1, i.e., the CG characters can be recognized and separated from

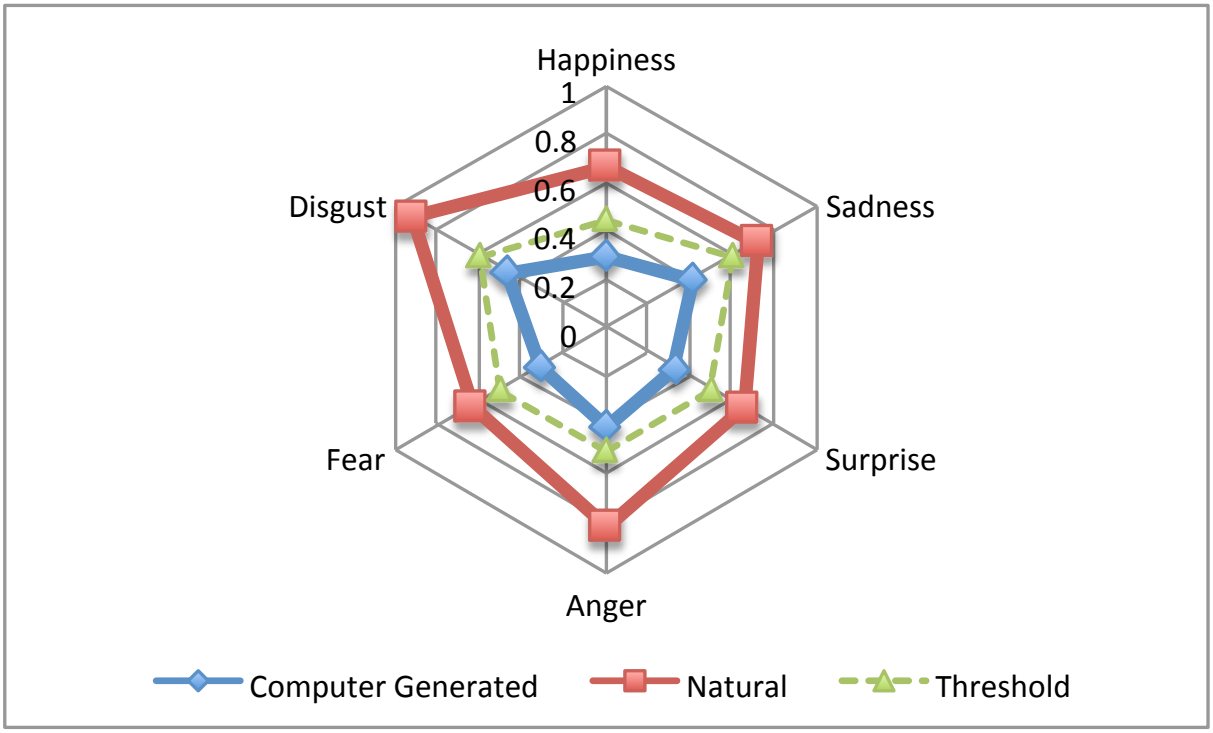


Figure 4.18: Average of *Expression Variation Values* analysed for all expressions.

CG and natural EVV_ξ are separated for all $\xi = 1, ..., 6$.

Table 4.5: Confusion matrices on CG and Natural faces

ξ	Expression		CG	Natural
1	Happiness	CG	**100%**	0%
		Natural	0%	**100%**
2	Sadness	CG	**100%**	0%
		Natural	0%	**100%**
3	Surprise	CG	**100%**	0%
		Natural	0%	**100%**
4	Fear	CG	**90%**	10%
		Natural	0%	**100%**
5	Anger	CG	**100%**	0%
		Natural	0%	**100%**
6	Disgust	CG	**80%**	20%
		Natural	10%	**90%**

The results are computed on JAFFE and D2 datasets.

the natural ones.

In the next section, experimental results of our last proposed method is presented.

4.5 Results of Experiments on ModelMethod

Three different groups of experiment were performed in order to evaluate ModelMethod. The first group was performed to measure the accuracy of the reconstruction step (step B, see Section 3.3.2). Using the best configuration obtained from these experiments, we ran our method on CG facial expression identification problem and compared the performance to ExpressMethod. Finally, in the last group of experiment, we applied our method on more challenging videos to evaluate the efficiency when it is applied to identify synthetic animations. BUHMAP, JAFFE, CASIA-3D FaceV1 and D3 datasets were used in these experiments.

4.5.1 3D Face Reconstruction

The first experiment is performed to measure the accuracy of the reconstruction step (step B) of the proposed method. We used a commercial software, namely Luxand FaceSDK [34], to extract 2D ASM feature points. In order to compute 3D PDM, we used 20 models in 3D of each person from 30 people from CASIA-3D FaceV1. Shown in Figure 4.3 some examples of images of CASIA-3D FaceV1. Algorithm 1 in Section 3.3.2 is applied to build the 3D PDM, i.e., $\overline{S}^{3D}$ and φ^{3D}. Notice that the angel pose for all faces are smaller than 30 degrees, which is the limitation of Luxand FaceSDK.

Given a 2D input facial image, 2D ASM is extracted on each image, then Algorithm 2 is applied to estimate the 3D shape S^{3D}. Finally, the error

err between the estimated shape and the referenced shape is computed:

$$err = \frac{1}{d}\sum_{i=1}^{d} ||\rho_i^{estimated} - \rho_i^{ref}|| \tag{4.3}$$

where $\rho_i^{estimated} \in S^{3D}$ and ρ_i^{ref} are manually marked and d is the number of feature points, which is mentioned in equation (**??**).

We used another 30 people from CASIA-3D FaceV1, 20 images on each person for testing. The result was obtained with the average difference of **5.83** pixels. Notice that face resolution is scaled into 400×400, hence the error is less than 1.5%.

Since CASIA-3D FaceV1 does not contain images of all poses (the poses are in the ranges of 0 - 15 degree or 80 - 90 degree), a test on a wider range of poses is necessary. Thus, we ran a second test on dataset D3.1 where images ranging from 0 to 30 degree. Shown in Table 4.6 are the results using 66 facial feature points, extracted by [34]. Notice that all poses are computed in Euler angles. It shows that yaw angle affects more than pitch angle. In the range of angle from 0 to 15, the average error on each pixels is 5.5915 (notice that for each face, the 2D image is scale into 400×400). Some examples are also illustrated in Figure 4.19. The third experiment

Table 4.6: Average errors err on different face poses.

pitch \ yaw	0 - 15	15 - 20	20 - 25	25 - 30
0 - 15	5.5915	7.8925	10.3567	15.3425
15 - 20	5.7617	7.9667	10.4215	15.5142
20 - 25	6.7624	9.3415	10.7141	17.1251
25 - 30	7.3215	9.1524	12.5214	25.1481

The errors are computed based on 66 points.

is performed to measure the accuracy with different setups of landmarks. The 2D facial features extraction returns 66 points for each input image.

However, the position of points at eyebrows or chin is often not very accurate due to occlusion or illumination conditions. Therefore, reducing the number of selected points usually allows to improve the performance of the reconstruction step.

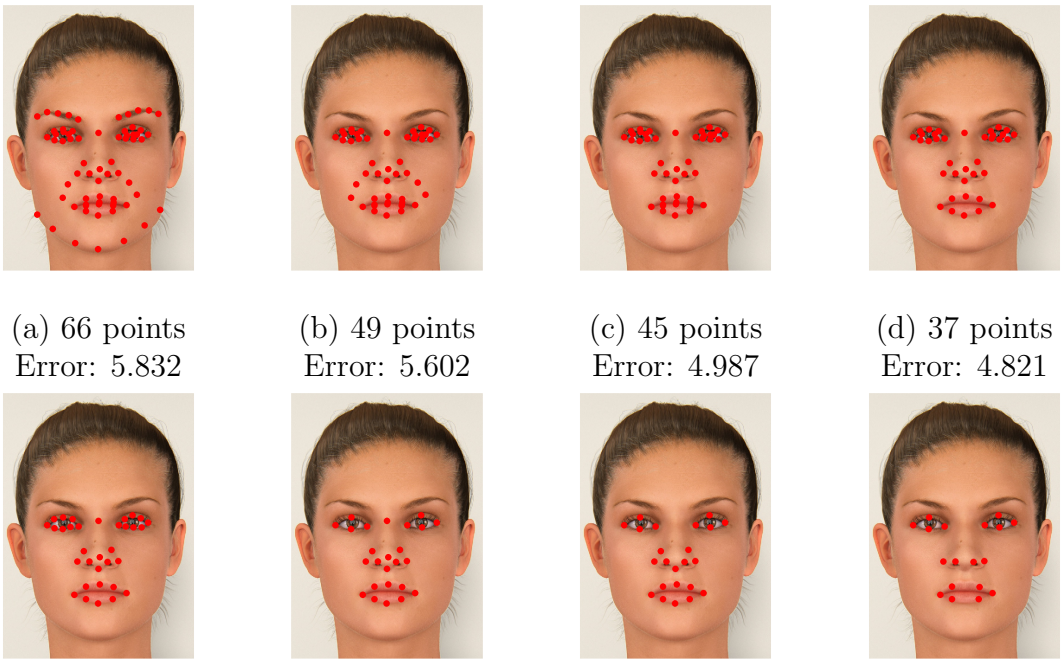
We used 30 people (20 images for each person) from CASIA-3D FaceV1 dataset to test some different setups, ranging from 66 points to 18 points. An illustration of all setups is shown in Figure 4.20 which also shows that setup f with 25 points provides the best solution with an average error of 4.1986 pixels (approximately 1%). This setup also provides the best solution on synthetic faces from set D3.1 with an average error of 4.142 pixels. Hence we used this configuration for all following experiments.

pitch = -14.37	pitch = -15.13	pitch = 10.03	pitch = 0.07
yaw = 0.13	yaw = 17.08	yaw = 1.08	yaw = -29.18
roll = 0.82	roll = 5.79	roll = 0.38	roll = -1.70

Figure 4.19: Samples of different poses for face reconstruction.

4.5.2 Computer Generated Facial Expression Identification

We ran our method on the BUHMAP-DB and JAFFE datasets and compare the results with ExpressMethod. Notice that in this case animations will be mainly consist of single expressions like happiness or sadness. In ExpressMethod, we compared different state of an expression of the same person, and the chronological order does not play any role in the method. Thus, we ran our proposed method with the windows size l in equation



(a) 66 points
Error: 5.832

(b) 49 points
Error: 5.602

(c) 45 points
Error: 4.987

(d) 37 points
Error: 4.821

(e) 33 points
Error: 4.749

(f) **25 points**
Error: **4.199**

(g) 23 points
Error: 4.869

(h) 18 points
Error: 5.2055

Figure 4.20: Different setups of facial landmark positions.

In our evaluations, setup (f) with 25 points provides the more accurate solution.

(3.15) equal $|V|$, where V is the input video, and $|\cdot|$ is the cardinal number of a set, i.e., extract features on the whole video.

We obtained an interesting result, in which the performance of the proposed approach was comparable with ExpressMethod only using value of σ^2 (see Section 3.3.3), i.e., without any support from machine learning models. Shown in Figure 4.21 (a) and (b) are sample results of σ^2 values on BUHMAP-DB happiness and sadness and on all expressions on JAFFE (Figure 4.21 (c)). Table 4.7 shows the explicit results on the whole BUHMAP-DB and JAFFE datasets.

Table 4.7: ModelMethod with σ^2 versus ExpressMethod

	Happiness	Sadness
ExpressMethod	67.5	**72.5**
ModelMethod with only σ^2	**71.82**	69.09

Accuracies are displayed in percentage.

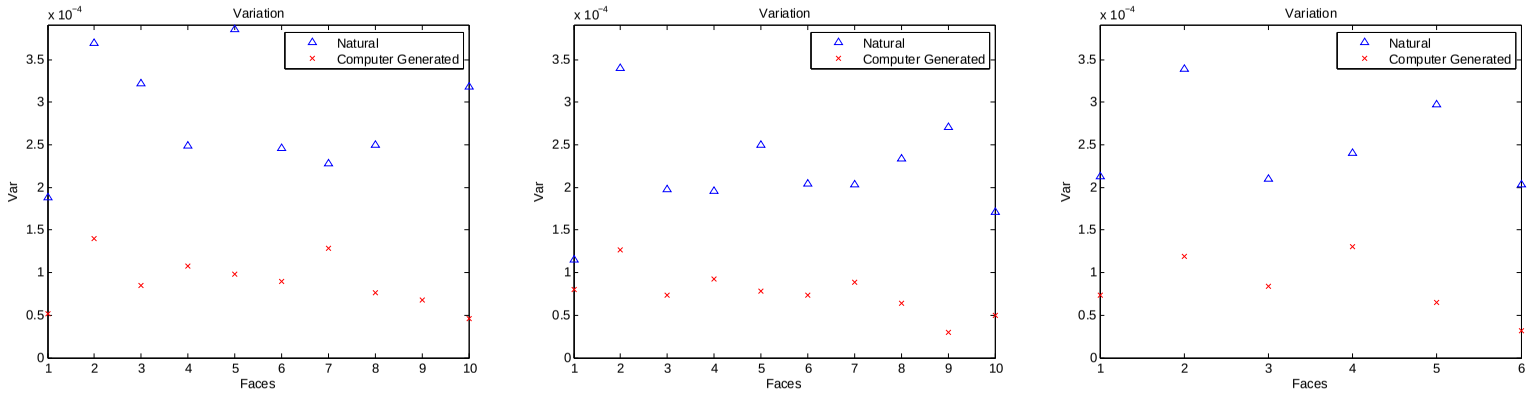
The last experiment in this section is performed by using the full configuration of the proposed method, i.e., all properties mentioned in section 3.3.3 are extracted. Windows size $l = 4$ and number of components $c = 3$ were chosen. Support vector machine was used as a binary classification and LOO (Leave one out) cross validation was applied in the test. The achieved accuracy outperformed results in ExpressMethod, as shown in Table 4.8.

Table 4.8: Comparing between ModelMethod and ExpressMethod.

	Happiness	Sadness
ExpressMethod	67.5	72.5
ModelMethod	**97.5**	**87.5**

Accuracies are displayed in percentage.

To summarize, ModelMethod outperformed ExpressMethod when using machine learning model. Without machine learning model, compara-



(a) BUHMAP happiness. (b) BUHMAP sadness. (c) JAFFE six expressions.

Figure 4.21: Sample results of σ^2 of ModelMethod.

The results are computed on BUHMAP and JAFFE datasets. According to these results, using the same threshold for all expressions, $T = 1.5 \times 10^4$, the accuracy obtained is comparable with ExpressMethod, which requires different thresholds for each expression.

ble results are obtained with a single threshold for all expressions, while ExpressMethod requires different thresholds for different expressions. Another advantages of the proposed approach is that no analysis of expressions is required, hence it can be used in the analysis of more general animations as tested in the last set of experiments.

4.5.3 Synthetic Animation Identification

In this experiment, we used 60 animations from BUHMAP-DB and 60 synthetic animations, reported as dataset D3.2 in Section 4.1. Notice that in this case, animations are more complicated since they consist of both expressions and other gestures of the faces.

We ran our proposed approach with different sets of features, i.e., different values of l and c, in order to determine the best configuration. SVM is used as a binary classification and LOO cross validation is applied in the test. The proposed method obtained the best result with the accuracy of 91.93% on the windows length $l = 4$ with the features extracted from the first 3 components, i.e., $c = 3$. The details are shown in Table 4.9 where columns are the numbers of components c and rows are the length l of the analyzed windows.

Table 4.9: Accuracy performance of ModelMethod on different configurations.

l \ c	1	2	3	4	5	6
2	66.57	76.08	73.54	69.37	45.81	49.11
3	75.36	84.93	90.40	74.65	65.00	49.82
4	70.20	88.19	**91.93**	85.25	62.29	54.25
5	73.32	88.15	90.27	70.12	61.05	52.59

Columns are the numbers of components and rows are the length of the analyzed windows. Accuracies are displayed in percentage.

The last experiment is performed to test the proposed method on more challenging videos, where we extracted 24 animations from 8 highly realistic

computer generated characters and collected another 24 animations from real persons (Dataset D3.3 in 4.1). Shown in Figure 4.10 are some examples of the realistic CG animations and the videos of real persons.

The same 3D PDM, which is computed as described in section 4.5.1, is used to extract 3D models. The configuration of $l = 4$ and $c = 3$ is used. SVM is again used as a binary classification. Using K-fold cross validation ($K = 6$ in the experiment) the achieved accuracy is **60.42%** while with LOO validation is **72.9%**.

Chapter 5

Conclusions

With the development of innovative multimedia technologies, the realism of computer generated characters has achieved a very high quality level. Non-existing subjects or situations can be easily generated. Thus, in a daily life context, it raises the need of advance tools supporting users in the identification of artificial data which may not represent reality. Although many interesting methods have been proposed to discriminate between CG and natural multimedia data, most of these methodologies do not achieve satisfactory performance in the detection of CG characters. Hence, in this doctoral study, we proposed efficient techniques to distinguish between CG and natural on this special kind object. Our methods are developed based on geometric-based forensic techniques, which exploit the measure on facial shapes formation and the evolution of facial animations. These solutions can be applied both for images and videos, in a wide situations and contexts.

In this document, our proposed methods were presented in Chapter 3, in which three methods are fully explained and discussed in details.

For the evaluation, we ran our experiments on various public datasets together with our own data, which are highly realistic CG characters from the computer graphics society and synthetic characters from advance ap-

plications for designers. Experiments were ran in different situations, from still images to video, from neutral expression to complex animations (see details in Chapter 4). The results confirmed that our methods performed highly accurately for distinguishing between CG and natural characters. The methods are not only work with both still pictures and video but also can overcome the difficulties in facial analysis caused by different face poses, occlusions, or lighting inconsistencies.

We also presented a complete picture of the State-of-the-Art approaches in this problem in Chapter 2. An overview on visual realism of computer graphics and the way that synthetic facial animations are created were also introduced in this chapter.

Since each proposed method can be applied in different situations, future work should consider the problem of fusing the three methods together. Automatic threshold selection is also a problem that could be taken into account. Extensions of these methods can be applied on other similar 'objects', which are deformable and are created by following rules or patterns, for example to the whole body of the human characters. The interval between transitions of expression could be also a promising discriminative feature. Computer graphics community can also gain some knowledge based on the proposed methods, hence they can create more complicated synthetic characters. In further work, the speech of the characters should be considered as mutual information. Other biological information, which are not presented in CG characters are also taken into account for future work.

Acknowledgements

One of the best moments of a long journey is to sit and look over the old-time that full of inspirers, friends and kinsmen/family who always go along with me and fulfill my road.

My heartfelt thanks go to Professor Giulia Boato and Professor Francesco G. B. De Natale, the inspirers of every successes, not only for their guidance but also for encouraging and helping me to shape my interest and directions. Thank you, Giulia, for everything, I owe you my most sincere gratitude.

I would like to thank the remaining members of my dissertation committee: Professor Fernando Pérez-González, Professor Alessandro Piva, and Dr. Raffaele De Amicis for their time and all the technical stuff as well as the secretary for making things running smoothly over the past years. What they gave to my thesis are not only the intellectual contributions but also the perfection.

As a member of MMLab, I have been surrounded by wonderful colleagues and friends. You have helped me to exploit new ideas as well as guided me how to enjoy the lifestyle in this wonderful country. Thank you all of you, MMLab members.

To my friends, over the years, you guys are always appear at the right time and helping me regardless of the reason. Thank you for just standing by my side.

Most of all, I would like to thank my parents and my family for encouraging and watching me with their endless love.

And last but certainly not least, to my dearest who shares love and life with me, thanks for being the kindler of my dream.

Bibliography

[1] O. Aran, İ. Ari, M. A. Güvensan, H. Haberdar, Z. Kurt, H. . Türkmen, A. Uyar, and L. Akarun. A database of non-manual signs in Turkish sign language. In *Signal Processing and Communications Applications*, 2007.

[2] Vassilis Athitsos and Michael J. Swain. Distinguishing photographs and graphics on the world wide web. In *IEEE Workshop on Content-Based Access of Image and Video Libraries*, pages 10–17, 1997.

[3] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In *ACM Computer Graphics and Interactive Techniques*, pages 187–194, 1999.

[4] Shen-Chi Chen, Chia-Hsiang Wu, Shih-Yao Lin, and Yi-Ping Hung. 2d face alignment and pose estimation based on 3d facial models. In *IEEE International Conference on Multimedia and Expo*, pages 128–133, 2012.

[5] Wen Chen, Yun Q. Shi, and Guorong Xuan. Identifying computer graphics using hsv color model and statistical moments of characteristic functions. In *EEE International Conference on Multimedia and Expo*, pages 1123–1126, 2007.

[6] Erika Chuang, Hrishi Deshpande, and Christoph Bregler. Facial expression space learning. In *Pacific Conference on Computer Graphics and Applications*, pages 68–76, 2002.

[7] V. Conotter and G. Boato. Analysis of sensor fingerprint for source camera identification. *Electronics Letters*, 47(25):1366–1367, 2011.

[8] V. Conotter and L. Cordin. Detecting photographic and computer generated composites. In *SPIE Symposium on Electronic Imaging*, 2011.

[9] Sintayehu Dehnie, Husrev T. Sencar, and Nasir Memon. Digital image forensics for identifying computer generated and digital camera images. In *IEEE International Conference on Image Processing*, pages 2313 – 2316, October 2006.

[10] Ahmet Emir Dirik, Sevinc Bayram, Husrev Taha Sencar, and Nasir Memon. New features to identify computer generated images. In *IEEE International Conference on Image Processing*, volume 4, pages IV–433 – IV–436, 2007.

[11] P. Ekman, Wallace V. Friesen, and J.C. Hager. *Facial Action Coding System (FACS): Manual*. A Human Face, 2002.

[12] Paul Ekman and Wallace V. Friesen. Constants across cultures in the face and emotion. *Journal of Personality and Social Psychology*, 17(2):124–129, 1971.

[13] Paul Ekman and Wallace V. Friesen. *Facial Action Coding System: A Technique for the Measurement of Facial Movement*. Consulting Psychologists Press, 1978.

[14] I. S. Penton-Voak et al. Symmetry, sexual dimorphism in facial proportions and male facial attractiveness. *Biological Sciences / The Royal Society*, 268(1476):1617–1623, 2001.

[15] N. Khanna et al. Forensic techniques for classifying scanner, computer generated and digital camera images. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1653–1656, 2008.

[16] T. Chen et al. Total variation models for variable lighting face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(9):1519–1524, 2006.

[17] T. T. Ng et al. Columbia photographic images and photorealistic computer graphics dataset. In *ADVENT Technical Report - Columbia University*, pages 203– 2004–3, 2005.

[18] T. T. Ng et al. Physics-motivated features for distinguishing photographic images and computer graphics. In *ACM International Conference on Multimedia*, pages 239–248, 2005.

[19] X. Xie et al. Normalization of face illumination based on large- and small-scale features. *IEEE Transactions on Image Processing*, 20(7):1807–21, 2011.

[20] H. Farid and M.J. Bravo. Perceptual discrimination of computer generated and photographic faces. *Digital Investigation*, 8:226–235, 2012.

[21] A. Gallagher and T. Chen. Image authentication by detecting traces of demosaicing. In *IEEE Computer Vision and Pattern Recognition Workshop*, 2008.

[22] Glenn Healey and Raghava Kondepudy. Radiometric ccd camera calibration and noise estimation. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 16(3):267–276, 1994.

[23] Tzvetanka I. Ianeva, Arjen P. de Vries, and Hein Rhrig. Detecting cartoons: a case study in automatic video-genre classification. In *IEEE International Conference on Multimedia and Expo*, pages 449–452, 2003.

[24] Singular Inversions. Facegen modeller 3.5. In *http://www.facegen.com*, 2004.

[25] ISO. *ISO/IEC 14496-2:1999: Information technology - Coding of audio-visual objects - Part 2:Visual*. pub-ISO, 1999.

[26] Eric Kee and Hany Farid. A perceptual metric for photo retouching. *Proceedings of the National Academy of Sciences*, 108(50):19907–19912, 2011.

[27] Matthias Kirschner, Meike Becker, and Stefan Wesarg. 3d active shape model segmentation with nonlinear shape priors. In *International Conference on Medical Image Computing and Computer Assisted Intervention*, volume 6892, pages 492–499, 2011.

[28] Jean-François Lalonde and Alexei A. Efros. Using color compatibility for assessing image realism. In *IEEE International Conference on Computer Vision*, pages 1–8, 2007.

[29] Chan-Su Lee and Ahmed Elgammal. Facial expression analysis using nonlinear decomposable generative models. In *IEEE International Workshop on Analysis and Modeling of Faces and Gestures*, pages 17–31, 2005.

[30] Chan-Su Lee and D. Samaras. Analysis and synthesis of facial expressions using decomposable nonlinear generative models. In *IEEE International Conference on Automatic Face Gesture Recognition*, pages 847–852, 2011.

[31] Lin Liang, Rong Xiao, Fang Wen, and Jian Sun. Face alignment via component-based discriminative search. In *Proceedings of the 10th European Conference on Computer Vision: Part II*, ECCV '08, pages 72–85, Berlin, Heidelberg, 2008. Springer-Verlag.

[32] Y. Liu, K. L. Schidt, J. F. Cohn, and S. Mitra. Facial asymmetry quantification for expression invariant human identification. *Computer Vision and Image Understanding Journal*, 91(1/2):138–159, 2003.

[33] Jan Lukáš, Jessica Fridrich, and Miroslav Goljan. Detecting digital image forgeries using sensor pattern noise. In *SPIE Symposium on Electronic Imaging*, 2006.

[34] Luxand. Luxand FaceSDK. In *http://luxand.com/facesdk/*, June, 2013.

[35] Michael J. Lyons, Shigeru Akamatsu, Miyuki Kamachi, and Jiro Gyoba. Coding facial expressions with gabor wavelets. In *IEEE International Conference on Automatic Face and Gesture Recognition*, pages 200–205, 1998.

[36] S. Lyu and H. Farid. How realistic is photorealistic? *IEEE Transactions on Signal Processing*, 53(2):845–850, 2005.

[37] S. G. Mallat. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):674693, 1989.

[38] Ann. M. McNamara. Exploring perceptual equivalence between real and simulated imagery. In *ACM Symposium on Applied Perception in Graphics and Visualization*, pages 123–128, New York, NY, USA, 2005. ACM.

[39] Gary W. Meyer, Holly E. Rushmeier, Michael F. Cohen, Donald P. Greenberg, and Kenneth E. Torrance. An experimental evaluation of computer graphics imagery. *ACM Transactions on Graphics*, 5(1):30–50, January 1986.

[40] TT Ng and SF Chang. Discrimination of computer synthesized or recaptured images from real images. *Digital Image Forensics*, pages 1–36, 2012.

[41] Feng Pan, JiongBin Chen, and JiWu Huang. Discriminating between photorealistic computer graphics and natural images using fractal geometry. *Science in China Series F: Information Sciences*, 52(2):329–337, 2009.

[42] A.C. Popescu and H. Farid. Exposing digital forgeries by detecting traces of resampling. *Transaction on Signal Processing*, 53(2):758–767, February 2005.

[43] Paul Read and Mark-Paul Meyer. *Restoration of motion picture film.* Butterworth-Heinemann, 2000.

[44] Microsoft Research. Microsoft Research Face SDK. In *http://research.microsoft.com/en-us/projects/facesdk/*, May, 2012.

[45] Luigi Rosa. EigenExpressions for Facial Expression Recognition. In *http://www.advancedsourcecode.com/facialexpression.asp*, 2007.

[46] Sam T. Roweis and Lawrence K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, December 2000.

[47] Gopinath Sankar, H. Vicky Zhao, and Yee-Hong Yang. Feature based classification of computer graphics and real images. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1513–1516, 2009.

[48] Sebastian H. Seung and Daniel Lee. The manifold ways of perception. *Science*, 290:2268–2269, 2000.

[49] Patchara Sutthiwan, Xiao Cai, Yun-Qing Shi, and Hong Zhang. Computer graphics classification based on markov process model and boosting feature selection technique. In *IEEE International Conference on Image Processing*, pages 2913–2916, 2009.

[50] Patchara Sutthiwan, Jingyu Ye, and Yun Q. Shi. An enhanced statistical approach to identifying photorealistic images. In *Digital Watermarking*, pages 323–335, 2009.

[51] Biometric Ideal Test. CASIA-3D FaceV1. In *http://biometrics.idealtest.org/*, June, 2013.

[52] Emanuele Trucco and Alessandro Verri. *Introductory Techniques for 3-D Computer Vision*. Prentice Hall PTR, 1998.

[53] Paul A. Viola and Michael J. Jones. Rapid object detection using a boosted cascade of simple features. In *CVPR*, pages 511–518, 2001.

[54] Paul A. Viola and Michael J. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57(2):137–154, 2004.

[55] Norman Wang and Wendy Doube. How real is really? a perceptually motivated system for quantifying visual realism in digital images. In *IEEE International Conference on Multimedia and Signal Processing*, volume 2, pages 141–149, 2011.

[56] Y. Wang and P. Moulin. On discrimination between photorealistic and photographic images. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages II.161–II.164, 2006.

[57] Jie Wu, Markad V. Kamath, and W. F. Skip Poehlman. Detecting differences between photographs and computer generated images. In *ASTED international conference on Signal processing, pattern recognition, and applications*, pages 268–273, 2006.

[58] B. Xu, J. Wang, G. Liu, and Y. Dai. Photorealistic computer graphics forensics based on leading digit law. *Journal of Electronics (China)*, 28(1):95–100, 2011.

[59] Qingshan Zhang, Zicheng Liu, Baining Guo, Demetri Terzopoulos, and Heung-Yeung Shum. Geometry-driven photorealistic facial expression synthesis. *IEEE Transactions on Visualization and Computer Graphics*, 12(1):48–60, 2006.

[60] Zhengyou Zhang. A flexible new technique for camera calibration. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, November 2000.

Appendix A

Realistic Computer Generated Characters Sources

Here are the list of realistic CG videos used in our experiments.

1. **OnLive's MOVA** - Geni4

   http://www.youtube.com/watch?v=0fF2pAsaaiw, April 2013.

2. **Activision R&D** - Realtime Demo (Nvidia Face Works Tech Demo)

   http://www.youtube.com/watch?v=CvaGd4KqlvQ, June 2013.

3. **Janimation technology and IGN**

   http://www.youtube.com/watch?v=5oqxH7ut8hU, April 2013.

4. **Pendulum Studio** - Alter Ego Facial Animations

   http://www.youtube.com/watch?v=-wtv4bsLWvw, April 2013.

5. **Gravity Design Studio** - Virtual 3D Avatar, Spokesperson, 3D

   http://www.youtube.com/watch?v=nX8KitVCcZM, April 2013.

6. **Image Metrics** - Emily CG Facial Animation is Too Real

   http://www.youtube.com/watch?v=UYgLFt5wfP4, April 2013.

7. CG facial animation (unknown authors)

   http://www.youtube.com/watch?v=2WOQ8UEE6as, April 2013.

8. **Faceware Tech** - Demonstration and Overview in GDC 2011

   http://www.youtube.com/watch?v=WO9W56KcCb8, June 2013.