CrossMark

Caricature generation utilizing the notion of anti-face

Vlasis Gogousis[1] · Anastasios Tefas[1]

Abstract The production of caricatures is a particularly interesting field of art, because it aims to highlight the very essence of a given face. Caricature generation systems traditionally rely on two approaches: they either follow extracted rules through learning algorithms, or follow rules that were directly programmed by experts. This paper attempts to reduce the reliance on heuristic methods, by proposing a novel method that provides a set of well-defined rules, which can be put to use for the purpose of caricature generation. The method is based on the notion of *anti-face* in conjunction with *unbiased* distortions. In addition, we indicate the usefulness of the anti-face as a means to perceive, for our own sake, the degree to which our face seems peculiar to others. Finally, we deploy a reverse variant of the method in order to attain beautification.

Keywords Caricature generation · Automatic · Anti-face · Average face · Unbiased distortions · Face oddity · Beautification

1 Introduction

The creation of caricatures constitutes one of art's most demanding and difficult fields. Making a caricature does not involve the mere deforming of facial features. It concerns the distortion of carefully selected features using deterministic methods that are derived from fuzzy rules. Selection of the wrong features or application of non-deterministic deformation rules produce outcomes that will most likely fail to result in accurate caricatures. For this reason, the development of systems that can ensure the realization of accurate caricatures is a major challenge.

Over the past few decades various interesting approaches have been proposed in the field. One of the first and most important attempts is that of Brennan's [2]. In her master thesis, Brennan developed an interactive system that helps the user select the dominant features of a

✉ Vlasis Gogousis
  vgogousis@gmail.com

  Anastasios Tefas
  tefas@aiia.csd.auth.gr

[1] Department of Informatics, Aristotle University of Thessaloniki, 54124 Thessaloniki, Greece

given face and then exaggerates them by comparing with the average face. Another well-known system is that of PICASSO [5]. The PICASSO system computes an average face and then produces caricatures based on the difference between the input and the average face. In "Making Caricature with Morphing", Akleman [1] designed a system that provides tools, which help a user to manually create caricatures. A different approach has been followed by Chang et al. in [3], where a feature database has been used, from which the final output is synthesized, after classifying the input to certain patterns. Towards this direction lies also the work of Liang el al. in [8], who used an artist's samples, in order to train a system to recognize exaggeration patterns and apply them to upcoming input. A noteworthy effort was made by Gooch et al., in [4]. Although their work focused on the generation of sketches from images, their program helped the user distort a face and produce funny sketches by providing a grid. P. Chiang et al. in [3] developed a method that processes the input, compares it with the average face and exaggerates features according to certain rules. The output is a face grid that is applied (warped) to an artist's sketches of various faces. G. Z. Xu et al. in [14] performed Principal Component Analysis (PCA) on the input face, to synthesize the output for various exaggeration degrees. A useful contribution has been presented by Mo et al. in [11]. The authors suggested that the variance of a population must also be considered in the process of caricature generation. Another probabilistic approach has been proposed by J. Liu et al. in [9], who suggested a method that trains a system to map inputs to outputs, in order to generate caricatures by using PCA and Support Vector Regression (SVR) learning. In a similar way, Chien-Chung Tseng et al. in [13] performed PCA and statistical analysis of the input, in order to determine prominent features and exaggerate them using sketches. Finally, J. Liu et al. in [10] tried to optimize their previous effort, by using a Manifold algorithm for dimension reduction, instead of SVR learning, during the training process.

As understood from the above, all efforts have focused on the development of systems that follow distortion rules, which are either *extracted* from the systems themselves –via learning– or are *provided* to them a priori, as the outcome of thought and experience of an expert.

In the first case, these rules can be expressed (e.g. as weighted vectors) but cannot be *described*. This kind of approach ultimately leads to an approximation of a space of caricatured faces, which cannot ensure accurate results for all inputs. Also, it requires training, which apart from being time and resource consuming, ties us to the quantity and quality of our samples, and limits us to the personal taste and style of the artists involved.

In the second case, the rules do not conform to proper scientific standards and are thus arbitrary. Whether it be tools provided to an artist, or a system that was programmed with the assistance of an artist, the dependence on the human factor has a significant impact on the output.

In a more general scope, all the aforementioned methods have a great reliance on the human touch, which intensely affects and shapes the end results. To avoid the development of intuitive systems that can lead to biased or inaccurate caricatures, we made an attempt to rule out the need for perspective. Separating subjective from objective in art is by itself a very ambitious task. Yet by logically analyzing the whole caricature generation process, we attempted to address this matter and propose a generic method that provides a set of well-defined rules, which can be put to use for the purpose of caricature generation.

2 Data preparation

To implement caricature generation systems, data preparation is required. Firstly, the way of representing facial information needs to be determined. Moreover, devising methods that are

able to process and compare faces in a meaningful manner must also be addressed. In this direction, some kind of normalization of the input data and a metric are to be specified, in order to decide what facial features should be exaggerated. In this section, we present the conventions and techniques that were used for pre-processing face data in our study.

2.1 Face representation

In our approach, the input data -that is, the face- are represented as a *set of points* on the plane, in what ultimately constitutes a face graph. These points are selected in such a way, as to optimally provide the information of the facial features that is required for analysis. This information is of geometrical nature (eye distance, face width, mouth height, etc.) and depends on the artistic guidelines followed by any given system. Thus, every face is expressed as a set of points $\{p_i \in Z^2 \mid p_i$ specifies the position of a facial feature on a face$\}$, which are then subject to analysis and processing.

In the process of caricature generation with the method developed herein, a face is taken as input, then it is compared with the average face, and afterwards an output face is obtained as the final outcome. The described process can be seen in Fig. 1. In order to compare faces, facial features are represented as non-dimensional normalized numbers (namely scalars), say x_i, where i is a feature index. *Non-dimensional*, in the sense that features are expressed as distance ratios. So, to be more precise, we do not perform comparisons between features, but rather between relations among features (ratios). *Normalized,* in the sense that these relations are determined according to a reference face, that is, the average face. Examples of feature relations are the ratios of the face height to face width, nose height to nose width, width of mouth to width of face, and so on. For instance for the

Comparison

Original face

Average face

Enhancement of characteristics
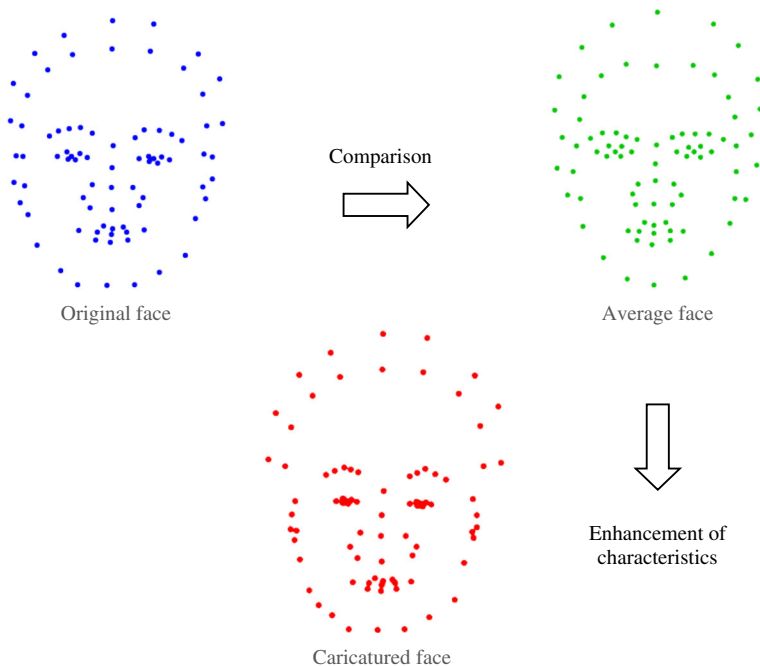
Caricatured face

Fig. 1 Caricature generation process. The original face is compared with the average face in order to determine what features ought to be enhanced. The result of enhancing those features is the caricatured face

feature "length of nose" we denote:

$$x_i = \text{Length of Nose Ratio} = \frac{\text{Length of Nose}}{\text{Length of Face}},$$

to express the length of the nose and compare and alter it, later on. Obviously $x_i \in [0, +\infty)$.

In every face we select N important feature relations, like the one described above. Thus, every face is mapped to a N-dimensional vector:

$x = [x_1, x_2, \ldots, x_N]$, where x_i are the previously described features.

2.2 The significance of the average face

The "Reference Face", as it is usually called, is a very important concept for caricature generation systems. The reference face is the face selected by the designers, with which all input must be compared, in order to determine the appropriate distortions for the face during succeeding stages of the overall process.

For the purpose of the present work, we choose the *average* face as our reference face, so as to determine the deviation of the input from the average face, and then amplify or de-amplify the differences. By "deviation" we mean the distance between the input and reference faces, according to a specified norm which is defined with the help of the points of the various features of the input and reference faces.

The actual selection of an average face is in the hand of the designer. Undeniably this depends on geo-political and cultural factors. From a practical point of view, though, the geometrical information of the average face that is needed for caricature generation is approximately the same, regardless of these factors. As a result, the statistically precise selection of an average face does not play a significant role in the field of caricature generation.

2.3 Face normalization

In order to perform fair comparisons among faces, we first have to normalize our face vectors. To do so, we begin by dividing each feature relation x_i with the corresponding average feature relation. If $\mu = [\mu_1, \mu_2, \ldots, \mu_N]$ is the average face, then our initial vector is:

$$x_n = \left[\frac{x_1}{\mu_1}, \frac{x_2}{\mu_2}, \cdots, \frac{x_N}{\mu_N}\right] = [x_{n1}, x_{n2}, \ldots, x_{nN}], x_{ni} \in [0, +\infty),$$

and for the average face:

$$\mu_n = [\mu_{n1}, \mu_{n2}, \ldots, \mu_{nN}] = [1, 1, \ldots, 1]$$

If we subtract the average vector from the input vector, then we have the following "diversity" vectors:

$$x_\delta = x_n - \mu_n = [x_{n1} - 1, x_{n2} - 1, \ldots, x_{nN} - 1] = [x_{\delta 1}, x_{\delta 2}, \ldots x_{\delta N}], x_{\delta i} \in [-1, +\infty)$$

Assuming that every face can be described by a diversity vector, as described above, a *face representation space* occurs. Let's name this space a "diversity" *space*. In this space, we define an operation between two of its elements, as the half addition of these two elements. If $x_\delta^{(1)}$ and $x_\delta^{(2)}$ are

two elements (faces), then by combining $x_\delta^{(1)}$ and $x_\delta^{(2)}$ a third element, $x_\delta^{(3)}$, is produced:

$$x_\delta^{(3)} = \frac{x_\delta^{(1)} + x_\delta^{(2)}}{2} = \left[\frac{x_{\delta 1}^{(1)} + x_{\delta 1}^{(2)}}{2}, \frac{x_{\delta 2}^{(1)} + x_{\delta 2}^{(2)}}{2}, \dots, \frac{x_{\delta N}^{(1)} + x_{\delta N}^{(2)}}{2} \right] = \left[x_{\delta 1}^{(3)}, x_{\delta 1}^{(3)}, \dots, x_{\delta 1}^{(3)} \right]$$

According to the above, the diversity space constitutes a *group* with its neutral element being that of the average face vector. Indeed, any elements of this space can be combined in order to produce a third element, while combining an element with the neutral element results in itself.

In this manner, for any given face x, such that $x_{\delta i} \in [-1, 1]$, we are in a position to determine its opposite, x', by simply finding a diversity vector x_δ' with which $(x_\delta + x_\delta')$ / $2 = \mu_\delta = [0,0,\dots,0]$. As we will soon see, limiting ourselves to $[-1, 1]$ does not pose a problem for our purpose of caricature generation. For the resulting vector x_δ', it is quite easy to determine x' and represent the final actual face. The corresponding face of this opposite vector is called "anti-face" or "counter-face", and consists of a face with opposite features to the prior face, namely, if the initial face has a small nose then its anti-face has a big nose, and so on.

Let's name the pair of x and its anti-face, x', "opposite pair". For an opposite pair we obviously have that $(x_\delta')' = x_\delta$. An instance of a pair of opposite faces is displayed in Fig. 2.

2.4 The anti-face as a means of self-perception

Having presented the notion of anti-face, we would like to point out a realization regarding it, other than its key role in our model for caricature generation. As presented by Rhodes et al. in [12], we all hold an average face representation in our brain, based on our visual experiences. Since we are constantly exposed to images of ourselves and people very similar to us, it follows that this average is slightly shifted towards ourselves; hence our face seems more normal to us than others. To understand the degree to which our face seems peculiar to others, we can use the anti-face. By rendering our anti-face using a statistically derived average (rather than the biased average derived from our experience) we can generate a face that deviates from the average the *same amount* as our own face does, only that it deviates in the complete *opposite* direction. Thus, we get a very firm depiction of a seemingly irrelevant face, which causes the same reaction to others as our own face does. This gives us a valid sense of how our face is *perceived* by others.

Fig. 2 Face and anti-face. A pair in the face space with the exact opposite features. By taking the average of any pair of faces such as this, we have the average of the face space

x μ x'
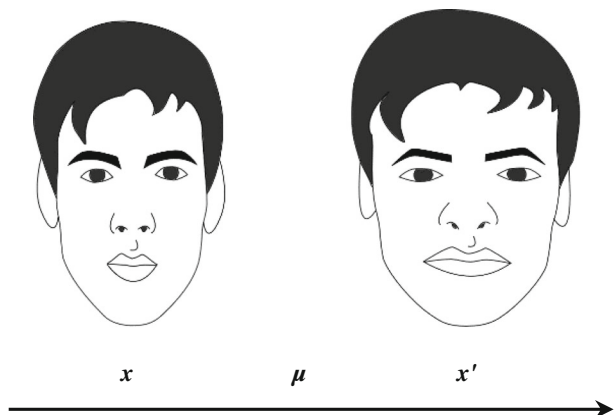
3 Determining facial feature exaggeration rules

Having analyzed how faces are represented and compared in this study, we now move on to the subject of how facial features are to be exaggerated. Let us begin so by defining the transformation of an element in the diversity space. If $f_i(x_{\delta i})$, $x_{\delta i} \geq -1$ is a transformation (deformation to be exact) of a feature $x_{\delta i}$, then:

$$f(\boldsymbol{x_\delta}) = [f_1(x_{\delta 1}), f_2(x_{\delta 2}), \dots, f_N(x_{\delta N})]$$

which is a mapping of $\boldsymbol{x_\delta}$, based on the functions f_i. The f_i functions can be either the *same* or *different*. If they are identical for all features, then the face is uniformly distorted. If they are different, then some features may be exaggerated easier or harder than others. In the latter case, this process is what constitutes an artist's "style", meaning his/her tendency to emphasize certain characteristics more than others.

Without loss of generality, from here on we will examine the case of using the same f_i functions, that is, the transformation of facial features will be uniformly applied. Therefore, when we refer to a transformation f, we are referring to the same rule, which is evenly applied to a face x, and thus the expressions $f(\boldsymbol{x_\delta})$ and $f(x_{\delta i})$ are an isomorphism, as described by Lewis and Papadimitriou in [7]. The presentation that follows is easily extended to cover the case of different f_i functions.

Before we move on to the proposed method, it is important to point out the significance of the average face and the anti-face in the process of caricature generation. As stated in [12], visual mechanisms code identity in relation to the average face. To perceive and understand a caricature, one's brain compares what is seen with the average face, in order to distinguish what features have been exaggerated. If an artist, for example, enlarges all noses in his or her work, an observer would be confused, because all caricatures would imply that all the corresponding faces have big noses. This is where the notion of anti-face comes in handy. The use of the anti-face not only helps sustain the average, but also helps provide balanced or "fair" distortions for the input faces. For all the aforementioned reasons, the use of unbiased functions for face distortion is crucial.

Let $\{\boldsymbol{x_\delta}, \boldsymbol{x_\delta'}\}$ be an opposite pair. The core of the method proposed herein relies on carrying out the following: In the process of selecting a transformation rule f for a caricature generation system, we impose the condition that the *transformation* of an opposite pair must be *itself* an opposite pair.

This condition can be stated with respect to the anti-face, $\boldsymbol{x'}$, of a face, $\boldsymbol{x}$, as follows:

The transformation of $\boldsymbol{x_\delta}$, $f(\boldsymbol{x_\delta})$, should coincide with the anti-face, $f(\boldsymbol{x_\delta})'$, of the transformation of the anti-face of $\boldsymbol{x_\delta}$, $f(\boldsymbol{x_\delta'})$.

Therefore we demand that the transitions in Fig. 3 hold:

From the condition described, we also have that a transformation rule f must not *shift* the average. By accepting this condition, we have that:
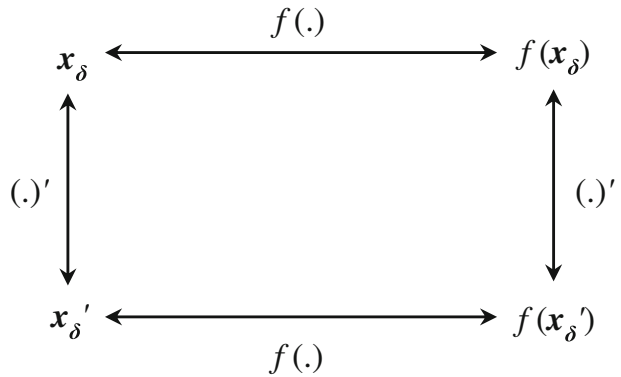
$$\frac{f(\boldsymbol{x_\delta}) + f(\boldsymbol{x_\delta'})}{2} = 0 \rightarrow f(\boldsymbol{x_\delta}) = -f(\boldsymbol{x_\delta'})$$

But, $\frac{f(\boldsymbol{x_\delta}) + f(\boldsymbol{x_\delta'})}{2} = 0 \rightarrow \boldsymbol{x_\delta} = -\boldsymbol{x_\delta'}$, so:

$$f(\boldsymbol{x_\delta}) = -f(-\boldsymbol{x_\delta})$$

From the last equation and knowing that $x_{\delta i} \geq -1$, we conclude that f must be odd and thus:

Fig. 3 State diagram showing the transitions between the space of original faces and distorted faces

- (α) $f(0) = 0$.
- (β) $x_{\delta i} \in [-1, 1]$, since $x_{\delta i} \geq -1$ and f is odd
- (γ) $f(x_{\delta i}) \in [-1, 1]$, because $f(x_{\delta i}) + 1 = f(x_i)$, and $f(x_i)$ is a distance ratio, which means that $f(x_i) \in [0, +\infty)$

We come to the conclusion that all f rules must both be defined and take values in $[-1, 1]$. The values -1 and 1 of features are *physical boundaries*. A value of -1 means that a feature does not exist, since it has 0 length. A value of 1 means that a feature is twice as large as the average, which is unnatural. For these extreme cases the application of a rule f has no purpose. Thus, for the depiction of such cases, if a feature is larger than 1 we leave it the same (since it is exaggerated on its own), while if a feature is smaller than -1 it is cut off at -1.

Note that due to f being odd, it follows that the whole process of caricature generation proposed herein is reversible, that is, there is no loss of information and thus we can regain the original face of a caricatured one by applying f^{-1} of Fig. 3, thus forming a cyclic model of transformations.

If we visualize all the above, we conclude that the rule we should select must belong to the space shown in Fig. 4, and must be anti-symmetrical with respect to the bisector of the axes:

Using curves such as in Fig. 4, we can define different artistic styles by determining the easiness with which separate facial features will be emphasized.

Since f is odd, the dashed lines on the left and right of (1,1) are of equal length. For $x_{\delta} = 0$ a feature remains unchanged, while for $x_{\delta} \neq 0$ a feature is amplified according to $f(x_{\delta})$.

Ultimately, the essence of all the analytical methods above could be reduced to the expression: "In order to produce caricatures that are perceived correctly, one must distort a given face in the same manner as he or she would distort its anti-face". Hence the need for symmetric transformations occurs, so as to maintain the average of both original and caricature faces equal and unchanged. Consequently, the enhanced features of a caricature can better indicate the deviation of the initial features of a face from those of the average.

4 Results

By implementing a system based on the proposed method and applying various exaggeration rates for the output, we can obtain the desired results. As mentioned in the previous section,

Fig. 4 Transformation rule space.
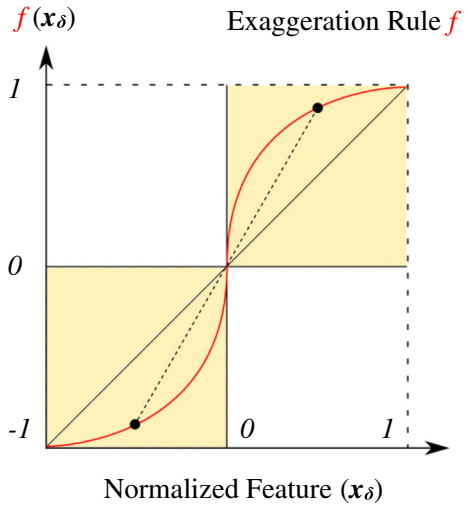
any exaggeration rule conforming to the described conditions of Fig. 3 can be applied for the purpose of caricature generation. For the demonstration that follows we chose a sigmoid function as our exaggeration rule:

$$f(\boldsymbol{x}_\delta) = \frac{2}{1 + e^{-\alpha x_\delta}} - 1$$

where α is the exaggeration factor. The sigmoid function is adjusted to fit the desired transformation rule space of Fig. 4. An appropriate polynomial or trigonometric function could have been used as well. Due to the properties of the above sigmoid function, values of α near 2 result to an approximate linear curve which leaves the input unchanged. For values of α greater that 2 exaggeration is achieved, while for values smaller than 2 features are driven toward the average.

Some examples of the resulting caricatures are depicted in column (b) of Fig. 5. Exaggeration factors ranging from 5 to 7 were applied, which produce an effective slope for the exaggeration rule that was used. As expected, the results not only conform to the proposed theoretical background, but also to our intuition. Indeed, we observe that the closer to the average a characteristic lies, the more it remains unchanged, while on the contrary, the further it deviates from it, the more it is enhanced towards the direction of that deviation.

In [6], Langlois and Roggman demonstrate that the average face is commonly considered to be attractive. In our case, we pursued the rendering of input faces into output ones with more attractive features, without the loss of the identity of any given face. Due to the properties of the distortion rules that were previously described, for curves similar to that of the reflection of f with respect to the bisector (f^{-1} of Fig. 4), de-amplification is obtained. This attenuation of features drives the input face towards the average, thus resulting in quasi beautification (see column (c) of Fig. 5).

Resuming the discussion over the resulting caricatures of our model, it is important to point out that the output can be parameterized and set to follow various *artistic styles*. An artistic style is when an artist has the tendency to highlight certain aspects of the face

Fig. 5 (a) Original faces, (b) Car-icatures generated with the pro-posed method, (c) Beautified faces

more than others. For instance, an artist may want to focus and emphasize on the deviation of people's eyes in his or her work. This means that the deviation of a person's eyes from the average is exaggerated more than the deviation of other characteristics. In

terms of our method, this means that different exaggeration rules are applied to various characteristics (i.e. rules like that of Fig. 4 but with different slopes). The caricatures in

Fig. 6 Caricatures focusing on the deviation of the initial faces' (a) eyes (b) noses and (c) head shapes
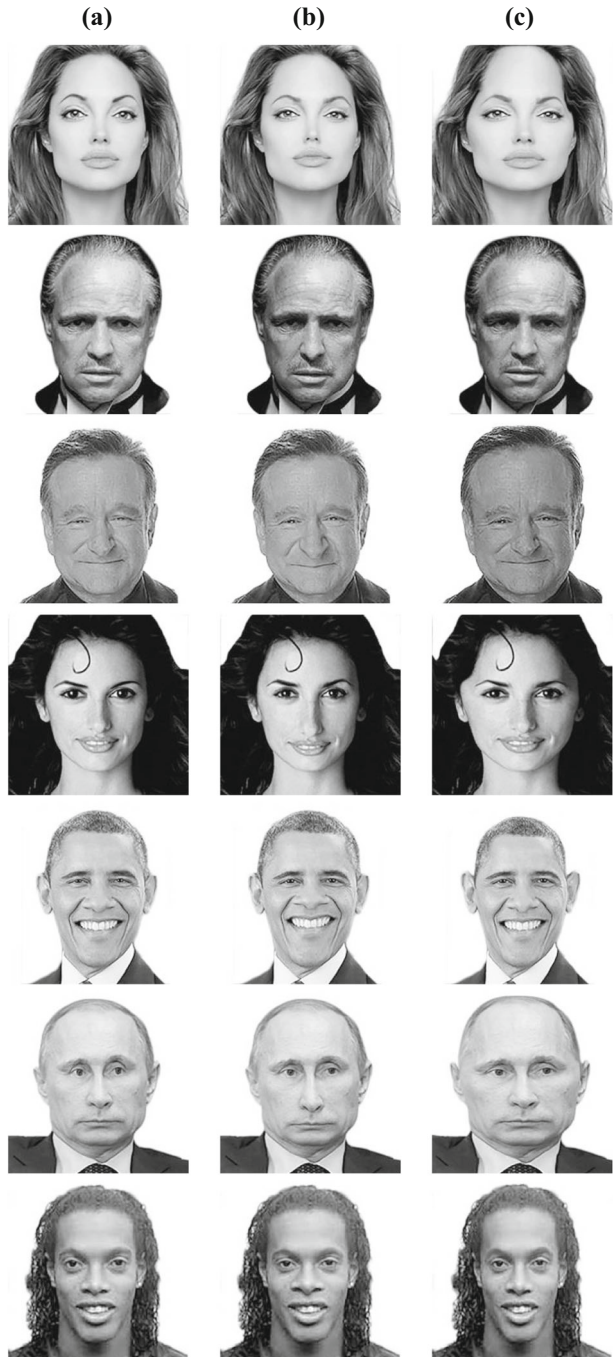
Fig. 5 were produced by uniformly applying the same rule to all facial features. The caricatures in Fig. 6 demonstrate an example of three different styles. Columns (a), (b) and (c) focus on the deviation of eyes, noses and head shapes, respectively, by using more sensitive exaggeration rules for these features. With respect to the sigmoid function that was used to produce these caricatures, each of the focused features in columns (a), (b) and (c) is exaggerated by a factor of $\alpha = 4$ (which is a moderate exaggeration), while the rest of the facial features are exaggerated by a factor of $\alpha = 2.5$ (which is a very subtle exaggeration).

By observing the caricatures of Fig. 6, it is clear that the main feature of interest is more easily exaggerated in every case. What's important, according to the scheme of Fig. 3, is that for any given style, the output face space is consistent as to the input face space.

5 Conclusions

Traditional methods for caricature generation are either highly dependent on human interaction and design or statistical analysis and machine learning. In some cases, this leads to transformation rules that are either biased or unaccented regarding feature exaggeration, as discussed in section 3. In other cases, this produces unpredictable or ambiguous results.

The purpose of this study was to try and setup a theoretical framework for selecting transformation rules that guarantees successful exaggerations of facial features. Towards this direction, we suggested that the consideration and employment of the average face in conjunction with the anti-face is effective and beneficial for the field of caricature generation. This consideration helps provide transformation rules that do not shift the average, and thus leads to the creation of non-arbitrary and non-heuristic caricatures. The rules that are defined by the proposed method are characterized by consistency and produce satisfying results. Furthermore, they allow for the easy handling of factors that determine the exaggeration rate, namely the achievement of various kinds of artistic styles of the output.

Efforts on building caricature systems like the one made herein, do not imply that human caricaturists can or should be replaced by automatic caricature generators. Computer generated artwork in all its forms can often be distinguished from that of humans' and is generally considered less appealing. Hopefully, methods like this can form the basis for producing alternative styles of caricatures, or assisting artists in their decisions regarding what features to select and how to exaggerate them.

In the framework of this study, we also indicated that the anti-face can be used as a means to understand the oddity with which our face is perceived by others. Lastly, we pointed out that a suitable variation of our method produces attractive faces by yielding output faces shifted towards the average.

References

1. Akleman E (1997) Making caricature with morphing. In: Proc. of ACM SIGGRPH, p 145
2. Brennan S (1982) Caricature generator. Master's thesis, Cambridge, MIT

3. Chiang P-Y, Liao W-H, Li T-Y (2004) Automatic caricature generation by analyzing facial features. In: 2004 Asian conference on computer vision, Jeju Island, Korea, January 27–30
4. Gooch B, Reinhard E, Gooch A (2004) Human facial illustrations: creation and psychophysical evaluation. ACM Trans Graph 23(1):27–44
5. Koshimizu H, Tominaga M, Fujiwara T et al (1999) On Kansei facial processing for computerized facial caricaturing system Picasso. In: Proceedings of IEEE International Conference on Systems, Man, and Cybernetics, Tokyo, p 294–299
6. Langlois JH, Roggman LA (1990) Attractive faces are only average. In: Psychological Science, vol. 1, no. 2, p 115–121
7. Lewis H, Papadimitriou C (1998) Elements of the theory of computation, 2nd edn. Pearson Education, Inc, Prentice Hall ISBN 0132624788
8. Liang L, Chen H, Xu Y-Q, Shum H-Y (2002) Example-based caricature generation with exaggeration. In: IEEE Proceedings of the 10th Pacific conference on computer graphics and applications
9. Liu J, Chen Y, Gao W (2006) Mapping learning in Eigenspace for harmonious caricature generation. In: 14th ACM international conference on multimedia, Santa Barbara, USA, October 22-27, 2006, p 683–686
10. Liu J, Chen Y, Gao XW (2009) Semi supervised learning of caricature pattern from manifold regularization. In: 15th international multimedia modeling conference, Sophia-Antipolis, France, January 7–9, 2009, p 413–424
11. Mo Z, Lewis JP, Neumann U (2004) Improved automatic caricature by feature normalization and exaggeration. In: Proceedings of ACM SIGGRAPH Conf. on Abstracts and Applications, New York
12. Rhodes G, Maloney LT, Turner J, Ewing L (2007) Adaptive face coding and discrimination around the average face. In: Vision Research vol. 47, p 974–989
13. Tseng CC, Lien JJ (2007) Synthesis of exaggerative caricature with inter and intra correlations. In: 8th Asian conference on computer vision, Tokyo, Japan, November 18–22, 2007, Proceedings, Part I, p 314–323
14. Xu GZ, Kaneko M, Kurematsu A (2004) A synthesis of facial caricature using Eigenspaces. In: Proceedings of electronics and communications in Japan. Part 3, vol. 87(8)

Vlasis Gogousis was born on December 13, 1992 in Thessaloniki, Greece. He received his B.Sc. degree in informatics, with specialization in digital media, in 2014 (with highest honors), from the Aristotle University of Thessaloniki, Greece. He fulfilled his military duty as 2nd Lieutenant Reserve of the Research and Informatics Corps from November 2014 to February 2016. In 2014 he participated in the Johns Hopkins University Center for Talented Youth summer program as a Teaching Assistant in the "Fundamentals of Computer Science" course. He has had various distinctions and awards in local and national drawing and sketching contests since his early childhood and has illustrated the cover pages of three international conference booklets (ACSTAC 2013, 2014, 2016). His artistic portfolio includes character design, caricatures, comic strips, and short animations for scientific, engineering, and educational purposes. He currently works as a Sketch Artist and Illustrator for university textbooks, while pursuing postgraduate studies in his field of interest, Computer Animation.

Anastasios Tefas (S'97-M'04) received the B.Sc. in informatics in 1997 and the Ph.D. degree in informatics in 2002, both from the Aristotle University of Thessaloniki, Greece. Since 2013 he has been an Assistant Professor at the Department of Informatics, Aristotle University of Thessaloniki. From 2008 to 2012, he was a Lecturer at the same University. From 2006 to 2008, he was an Assistant Professor at the Department of Information Management, Technological Institute of Kavala. From 2003 to 2004, he was a temporary lecturer in the Department of Informatics, University of Thessaloniki. From 1997 to 2002, he was a researcher and teaching assistant in the Department of Informatics, University of Thessaloniki. Dr. Tefas participated in 12 research projects financed by national and European funds. He has co-authored 74 journal papers, 178 papers in international conferences and contributed 8 chapters to edited books in his area of expertise. Over 3600 citations have been recorded to his publications and his H-index is 32 according to Google scholar. His current research interests include computational intelligence, pattern recognition, statistical machine learning, digital signal and image processing and computer vision, biometrics and security.