Analysis of Snapchat Mesh Triangulations in Facial Detection and Mapping

Nika Korchok

Tufts University: COMP163: Computational Geometry

Author Note

This report documents research and results for a project for COMP163 at Tufts University, taught in the Fall 2016 semester by Prof. Greg Aloupis.

Acknowledgments: My mum, for always giving me feedback on drafts of papers and drawings related to this project, and for supporting me through a year-long project on triangles.

Further inquiries can be directed to nikakw@gmail.com

Abstract

This project was commenced as an investigation into the structure and development of one particular triangulation -- that used by Snap Inc. to render real-time photo filters of user faces in their app Snapchat -- as it relates to the application of popular facial detection algorithms in rendering facial meshes. The project attempts to clearly define, for the reader, the main procedures of facial recognition and mapping algorithms, in order to illuminate a seemingly opaque technology and to open a critical discussion of the problems and advantages of the utilized algorithms and their efficacy in rendering persistent facial meshes.

In short, this paper hopes to (1) clearly explain how the Snapchat technology renders a triangular facial mesh, (2) explain and demonstrate an alternate program for computing a user's triangular facial mesh, utilizing an overall parallel algorithmic structure with several marked differences in procedural decisions, and (3) provide an analysis of the logistical and ethical implications of both the algorithms used within the Snapchat technology and the technology itself.

Part I is divided into four sections, each explaining a major module of the program, to provide the reader with conceptual background in the utilized computer vision algorithms. Utilizing references from both the U.S. Patent of the technology itself, as well as supplemental texts, Part I explains how the Snapchat technology (I) recognizes a face, (II) calculates facial landmarks on a face, (III) uses these landmarks to calculate an initial triangulation of the point set defined by these landmarks, (IV) refines this triangulation using Steiner vertices, in order to generate an optimal facial triangulation.

Part II is divided in the same four subsections as Part 1, the difference being that Part 2 explains a novel approach by the author to replicate the Snapchat technology in generating images with a triangular facial mesh; the output from Part 2 serves as as a tool for further analysis of the triangulation structure in Snapchat.

Part III proposes a critical analysis of the algorithms and methodologies defined therein, allowing for a dialogue concerning the choices made in algorithmic design as they relate to facial detection and mapping.

Keywords: POINT LOCATION, NEAREST NEIGHBOR SEARCH, FACIAL RECOGNITION, FACIAL MAPPING, MESH GENERATION, DELAUNAY TRIANGULATION, VORONOI DIAGRAM, MESH REFINEMENT

A Critical Examination of Snapchat Mesh Generation and its Implications

As a means for illustrating and connecting several key concepts in the field of Computational Geometry -- namely Point Location, Point Set Triangulation, Delaunay Triangulation and Voronoi Diagrams, and Edge-Flipping in the context of constraints for conforming triangulations -- the Snapchat Triangulation appears an ideal model. Appearing only briefly -- and infrequently -- to users, the triangulation that subdivides a detected face on the Snapchat app is deceptively elegant. Yet it is the appearance of this structure at all which raises important questions about the algorithms central to the projection of a filter onto a photo within the app: How is the face, and its corresponding facial landmark points, detected within the app? Which algorithms for facial detection and landmark detection are employed as part of this technology and why were they chosen over other available options? Are the initial detected facial landmarks retained as persistent markers or are they modified in subsequent procedural steps in the algorithm? Could, or should, these facial landmarks be used as input for a nearest neighbor search for refining the point set before triangulation can take place? Which triangulation is used for the facial mesh? Why?

Most importantly, perhaps, the question arises as to why Snapchat even decided to reveal the triangulation to a user at all. Formally titled in its U.S. Patent as a “Method for real-time video processing involving changing features of an object in the video,” the end result of the Snapchat technology is a filter, or a Snapchat dubbed “Lens,” with some unique feature such as puppy ears or a clever pair of spectacles, being projected onto the face of the user.

Considering that the development of a facial mesh is comprised of myriad steps (of rather staggering complexity, for the average user), and that the facial mesh itself is not even the end result presented to a user on the app -- it is not immediately evident as to the necessity of presenting the triangulation to a user, and in such a fleeting and sporadic manner.

Introduction

It must be noted that Snapchat, until recently, did not make available an open-source version of the software used for generating filters on user images. According to a report on PetaPixel, Snapchat bought the technology that is now used to render the triangulation and filter projection from Ukrainian company Looksery in 2015 for $150 million[1].

According to an article on Slate, as of December 16, 2017, Snap Inc. released a new major platform for Lens creation, called Lens Studio, where users can develop their own augmented reality Snapchat Lenses via the downloadable desktop app developed by Snap Inc. Lenses; once a Lens design has been submitted to, and approved by, Snapchat, a QR code allows for easy shareability within the app amongst a user's contacts.[1] Templates available on the Lens Studio website include Static Objects, Animated Objects, Interactive Tap, Interactive Look At, Interactive Approach.[2]

Already, the simplicity of the selfie triangulation has been extended into the seemingly infinite world of Augmented Reality. The proliferation of the app's usage, amongst a

[1] "Prepare for More Dancing Hot Dogs."
[2] Lens Studio. "Templates." https://lensstudio.snapchat.com/templates/

wide-ranging user demographic -- most notably amongst millennials but also increasingly within the communities of baby boomers (see @BobSaget for further proof) -- already warrants a closer examination of the technology that yields, via a triangulation, a morphed facial filter (henceforth referred to in following the company standard term “Lens”). The necessity for examination of the technology has increased with the release of Lens Studio.

This project stems from my own investigative search for the methodology behind the Snapchat triangulation. It presents a naive algorithm which, when presented with a user-supplied photograph of a human face, returns a modified image of that face with the applied “Lens” being the Voronoi Diagram and/or Delaunay Triangulation of the point set of the face. This Lens was used as a tool for analysis of the Snapchat Lens technology, as a means to recreate the algorithmic processes of the technology and provide a basis for a more nuanced understanding of its methodologies and implications.

Project Methodology

Participants

Participants in this paper are limited to the author (myself), as I used my own image for all of the subsequent work in attempting to rebuild the SnapChat triangulation on a supplied image. The singularity of the target audience for this technology also supported this choice -- i.e. each Snapchat user is encouraged to participate in, what is known colloquially as “selfie culture,” by turning the camera on their own personal image to transform their face using a filter (what Snapchat refers to as a “Lens”). Future participants could be extended to anyone wishing to use the program I wrote, in order to render their own unique images from the triangulation frameworks. The author also extends the invitation to the reader to become a participant, by reflecting upon, and questioning, the ubiquitousness of Snapchat's Lens technology and by recognizing the methodological choices made in its design and opening a dialogue for their evolution and re-interpretation.

A Note on Images

All images included in this paper are my own, generated as output from the programs I wrote for this project. Figures 1 and 2 were screenshots taken of the Snapchat screen, capturing the triangulation that briefly and inconsistently appears before a user can apply a Lens to an image capture. All other images are modified versions of the same original input image. This input image, in keeping with the spirit of the technology itself, is a selfie.

All associated code for this project can be found on my Github repo:
https://github.com/nikaroxanne/comp163project

Additional images generated from this project can be found on my website:
https://nikaroxanne.github.io/

Assessments and Measures

The paper is divided into three sections: (I) An overall survey of several common image processing techniques, used by Snapchat in their Lens technology -- including, but not limited to, face detection, facial landmark detection, and mesh generation; (II) a presentation of my own triangulation software, based on that developed by Snapchat, used as a tool for analysis of the Snapchat triangulation; (III) a comparative analysis of the two algorithms via their methodologies and results.

I. The Snapchat triangulation algorithm

Definitions

Facial detection: In computer vision, a facial detection algorithm returns a boolean value corresponding to the presence of a human face in an input file -- true if yes, false if not. Facial detection algorithms can also implement other functionality, such as returning (x,y) coordinates of where these detected "faces" are located in the image. The OpenCV module, for example, returns a Java Rect() object, representative of the area around each detected face; the Rect() object has data members for the x and y coordinates for the upper left corner of each bounding box, and the box width and height. A facial detection algorithm can take a user-supplied input file in a variety of formats, depending on the implementation of the algorithm -- a still JPEG, PNG or other photographic format, a recorded video file or a live-stream video.

Facial recognition: Facial recognition is the process by which a detected face is used as a key for searching in a supplied library. If the key -- here an algorithm-defined specification of distinctive markers, contours, etc. that describe a face and its shape -- is found, then the identity of that face can be returned. Facial recognition is used in domains such as the criminal justice system, wherein facial recognition can match captured images against a database.

Facial mapping: In computer vision, facial mapping refers to a multilayered approach to image processing: facial detection and then persistent detection and reconfiguration over time to maintain a facial structure, utilizing the face and its features, or landmark points. Facial mapping is related to facial tracking -- which, herein, is defined as persistent feature detection, irrespective of shifts in light, orientation or relational distance between camera and subject.

Triangulation: A partition of a point set, polygon or planar straight-line graph (PLSG) into triangles (see Figure 3).

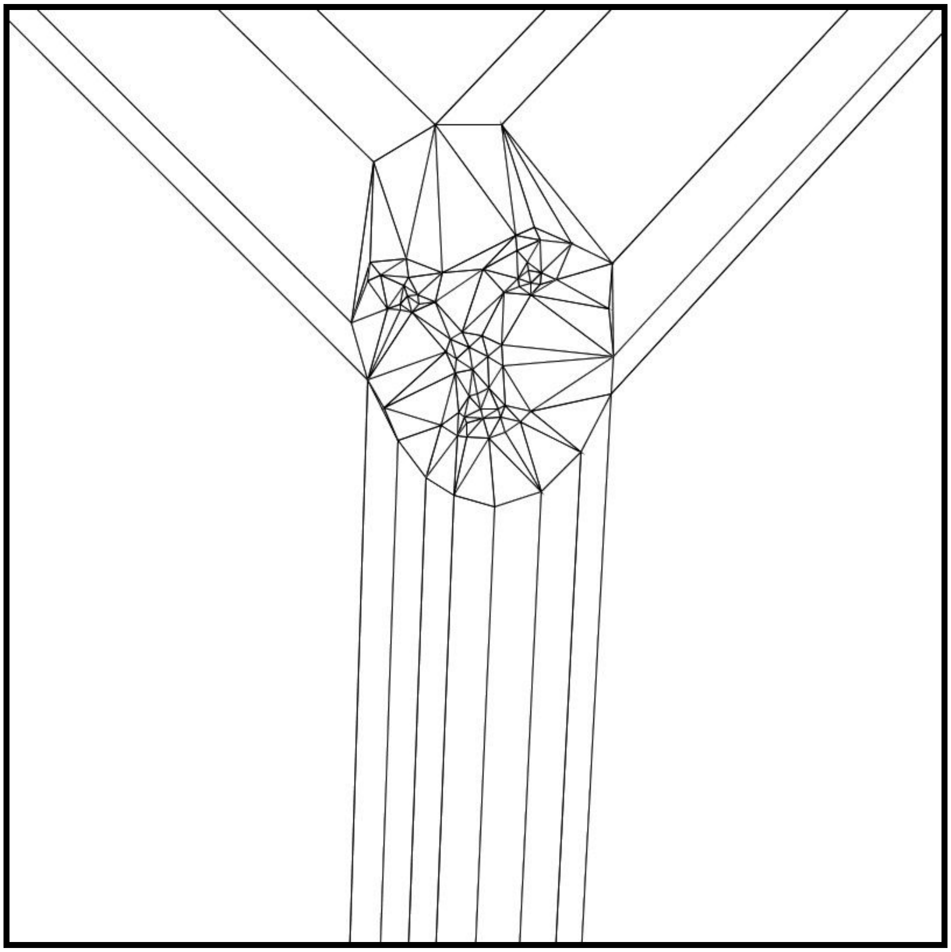


Figure 1: Triangulation of a point set of facial landmarks

Delaunay Triangulation: A triangulation in which each triangle is defined by the empty circle property, that is that the circumcircle, formed by the three vertices of a Delaunay triangle, contains no other input point. The Delaunay triangulation is the dual of the Voronoi diagram. A Delaunay triangulation also maximizes the minimum angle in each triangle, so as to avoid “skinny” triangles, which is one property which makes it an optimal triangulation in some cases. (Again, see Figure 1).

Voronoi Diagram: A Voronoi diagram is a nearest-neighbor division of a space— it determines the delineations between points such that every Voronoi edge between two input points is equidistant from those two points. (See Figure 6)

Figure 2: A Voronoi Diagram of Facial Landmark Points

Conforming Delaunay Triangulation: Referring to Bern et al, "A conforming Delaunay triangulation of a PSLG [Planar Straight-Line Graph] is a triangulation in which extra vertices—Steiner points—are added to the input, until the Delaunay triangulation of the vertices "conforms" to the input, meaning that each input edge is a union of Delaunay edges."[3]

[3] Marshall Bern et al. "Chapter 29: Triangulations and Mesh Generation," Preliminary version (July 28, 2017). To appear in the "Handbook of Discrete and Computational Geometry," J.E. Goodman, J. O'Rourke, and C. D. Tóth (editors), 3rd edition, CRC Press, Boca Raton, FL, 2017. https://www.csun.edu/~ctoth/Handbook/chap29.pdf

Steiner point: A point that is not part of the original input set but that is added retroactively.

Mesh: A mesh is similar to a triangulation, in that it is subdivides a form into polygonal partitions, frequently triangles or rectangles. In the chapter “Triangulations and Mesh Generations” in the preliminary version of “Handbook of Discrete and Computational Geometry,” authors Marshall Bern, Jonathan R. Shewchuck and Nina Amenta define a mesh as “a decomposition of a geometric domain into elements, usually triangles or quadrilaterals in R^2” and a conforming mesh covers all of the input domain.[4]

Introduction

Exploiting geometric series in image-based Markov processes

Facial Landmark detection can essentially boil down to the same logic employed in a deterministic selection algorithm -- exploiting the structure of a geometric series, one can break up the problem into smaller pieces by discarding unnecessary image portions and concentrating on smaller pieces; in this way, an efficient algorithm can detect a face and its landmarks with high accuracy.

Much of facial detection and facial landmark detection relies on the very principle exploited by Claude Shannon in his seminal paper, “A Mathematical Theory of Communication” wherein,

[4] Marshall Bern et al. “Chapter 29: Triangulations and Mesh Generation,” Preliminary version (July 28, 2017). To appear in the “Handbook of Discrete and Computational Geometry,” J.E. Goodman, J. O'Rourke, and C. D. Tóth (editors), 3rd edition, CRC Press, Boca Raton, FL, 2017. https://www.csun.edu/~ctoth/Handbook/chap29.pdf

describing sequences of letters “There exist a finite number of possible “states” of a system…[and] a set of transition probabilities. To make this Markoff process into an information source we need only assume that a letter is produced for each transition from one state to another. The states will correspond to the “residue of influence” from preceding letters.”[5]

A Markov process is a stochastic process, as defined by Papoulis in “Probability, Random Variables and Stochastic Processes,” and by Eric Weisstein on Wolfram Math, who also cites Papoulis in his definition. A Markov process is “A random process whose future probabilities are determined by its most recent values.”[6]

In the case of an image, each pixel contains a data value for pixel brightness constrained to a range of 0-255; a value of 0 corresponds to a black pixel, a value of 255 corresponds to a white pixel. Rather than having an object with a state corresponding to one of 26-letters in an English alphabet, pixels each have 2^8 states.

According to Martin Janžura, in his paper “Image Processing, Markov Chain Approach”, the division of an image into a Markov chain approximation can be explained thus: “By image we understand a configuration of states $x_s = (x_s)_{s \in S} \in x_s = {}_{s \in S} \otimes X_s$, where S is a finite set of indices (sites), and X, for each s $\in$ S is a finite state space. Usually, the set S is a rectangular

[5] Claude Shannon. “A Mathematical Theory of Communication.” Page 9.
[6] Eric Weisstein. "Markov Process." Wolfram MathWorld. http://mathworld.wolfram.com/MarkovProcess.html

area in the two-dimensional integer lattice Z^2, and each X_s, is a copy of some X_o. Then the image is an array of states."[7]

Obviously, and as Janžura notes, Markov processes must be evaluated in the context of image processing while taking into consideration several strong assumptions about the image. An image made entirely of noise, for example, is, by its definition, comprised of pixels, each of which taking a random value (in this case, for simplicity, we will only consider the brightness value.) This pattern — or perhaps non-pattern — does not lend itself particularly well to image processing following a Markov model, simply because each pixel's value is not dependent on preceding values in a "chain" — in this case, the array of pixels comprising an image.

In the context of facial detection, however, Markov processes can help to illuminate a pattern of brightness values which define a face, as it is recognized by a computer program. In the same way that humans perceive depth by shifts in value, and the way that artists are able to render realistic depictions of forms from life, knowing where light meets shadow is at the very nature of how any form can be discerned from another. The bridge of the nose, for example, will typically have a lighter value (a higher brightness value) than the sides of the nose, because it is where light strikes first; the sides of the nose will appear comparatively darker, the adjacent eye sockets and nostrils will have an even darker value, all corresponding to their relative placement as neighbors in the same space.

[7] Martin Janzura. "Image Processing, Markov Chain Approach." 1996. Page

Understanding these shifts in tone, or brightness value, can help us to recognize and establish patterns of predictability in certain forms. As in Shannon's crossword predictions, knowing a brightness value at a particular position on a face, can help to predict the subsequent value; akin to first-order word approximations, knowing that a nose bridge must have a brightness pattern of light-dark-light can help to establish something of a canonical form to check against.

It is from this understanding that we can begin our breakdown of each step in the process of Snapchat's Lens technology. Establishing patterns within neighbourhoods of pixels can allow for processes such as facial detection, facial landmark detection, and eventual refinement of a point set to render a persistent facial mesh.

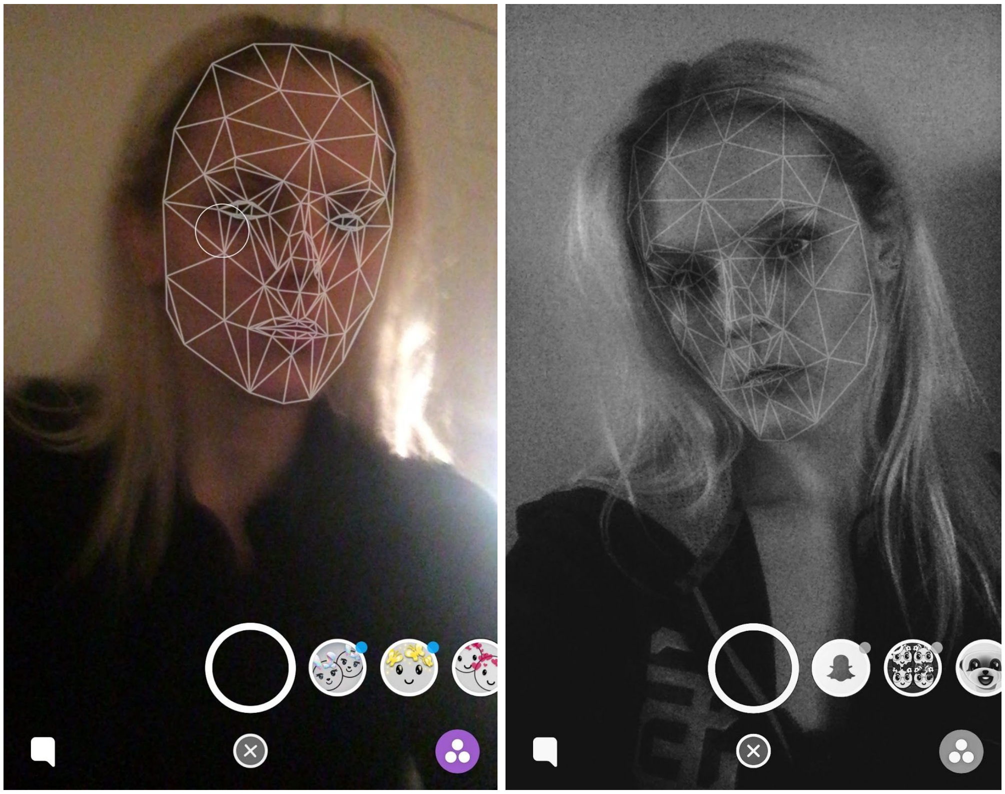


Figure 3 and 4: Screenshots of Snapchat triangulation
Both images were captured after hundreds of unsuccessful attempts, due to the brevity of the triangulation on the screen and its inconsistency as a feature

Section 1: Face Detection

The algorithm utilized by Snapchat for facial detection (as detailed in the notes in the patent for "Method for real-time video processing involving changing features of an object in the video"), is that which has become somewhat the canonical facial detection algorithm, at the very least an industry standard in computer vision and image processing: The Viola-Jones algorithm.

The Viola-Jones algorithm utilizes pixel group-processing to detect features; this exploits the Markov property present in neighbourhoods of image pixels. The algorithm employs a methodology akin to playing an off-kilter version of chess: it scans an image file, analyzing the brightness of each pixel within a rectangular neighbourhood, checking to see if certain patterns appear which indicate that a face/facial structure is present in the image. If patterns appear, the algorithm moves forward. If not, it returns, as though check-mated.

This pattern scan is done using Haar-like features[8] — rectangles with divisions which indicate different shifts of light and dark. The input image is scanned and checked against these rectangles to calculate shifts in pixel intensity. This is done in several phases, each phase further refining the pattern in the Haar-like features to correspond to smaller areas; this multi-phase process (a simplified explanation of cascading classifiers) also uses the Adaboost learning algorithm[9] -- both of which are described in much greater detail in the paper itself, to which the reader is referred for a more thorough understanding of this specific algorithm. If the image passes all rounds of these checks, indicating that all Haar-like features have been confirmed in the image, the algorithm responds with an implementation-defined output: i.e. a simple TRUE (for some defined Boolean that represents Facial Detection in the input image); number of detected faces; even a bounding box drawn around each detected face.

In OpenCV, an open-source computer vision library which implements the Viola-Jones algorithm for its opencv.detect() function, the implementation returns such a response: the function returns a Rectangle object, with data members for the width, height and top-left (x,y)

[8] Paul Viola and Michael J. Jones. “Robust Real-Time Face Detection.” International Journal of Computer Vision 57(2), 137–154, 2004. Kluwer Academic Publishers.
[9] Paul Viola and Michael J. Jones. “Robust Real-Time Face Detection.” International Journal of Computer Vision 57(2), 137–154, 2004. Kluwer Academic Publishers.

coordinate corresponding to each detected face, allowing for a clear visual representation of these detected faces(see Figure 5.)

From here, having a defined region, or regions, of interest, further image processing can be refined to take into account these ROIs, allowing an analysis of smaller subsections of the image for processes such as facial landmark detection.

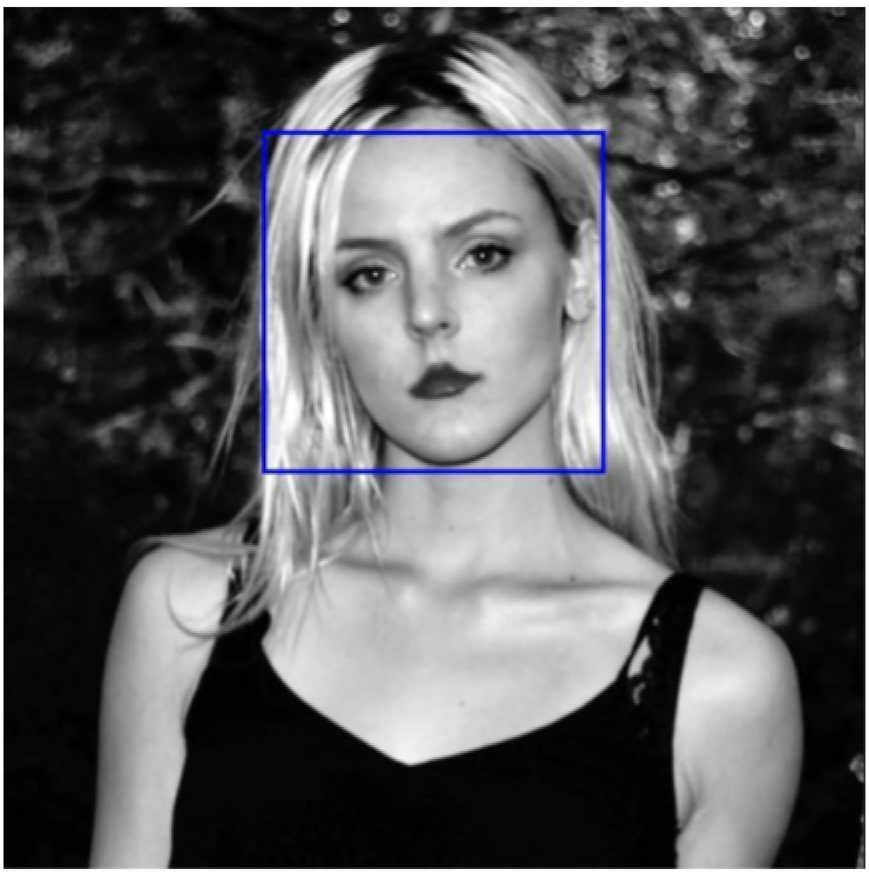


Fig 5: Bounding Box generated using OpenCV module

Section 2: Facial Landmark Detection

Active Search Model

In Snapchat's Lens technology, in following with what is described in the patent by Shaburova, facial landmarks are detected as key points using an Active Search Model.

In "An Introduction to Active Shape Models," Tim Cootes extrapolates on the methodology for this process, which revolves around the use of a "golden image" for template matching: a golden image is constructed from a set of training images; landmark points are identified in an image, typically those points with high variation, such as corners are strong candidates.[10] Each training image generates a landmark-defined shape, which then allows for the construction of a general shape. As explained by Cootes, "The approach is to translate, rotate and scale each shape so that the sum of distances of each shape to the mean ($D = \Sigma |x_i - \bar{x}|^2$) is minimised."

In the patent, Shaburova describes the use of the ASM as a two-fold procedure: (i) Suggest a tentative shape by adjusting the locations of shape points by template matching of the image texture around each point (ii) conform the tentative shape to a global shape model."[11] In the context of this paper, only a simplified general understanding of the ASM is established; it is left to the reader to read the associated cited papers to dig deeper into this concept. The actual procedure implemented in the Snapchat technology utilizes a multi-layered pyramid structure of

---

[10] Tim Cootes. "An Introduction to Active Shape Models." Page 5.

[11] Elena Shaburova. "Method for real time video processing for changing proportions of an object in the video." US20150221118A1. Page 3.

coarse-to-fine image processing, akin to that described in Cootes' paper, though forgoing a Gaussian model.

In the Snapchat technology, points initially detected are then further refined to a second set, wherein the landmark location are transformed via point moves. This second set is more precise in its predictions and allows for a more accurate mapping of landmarks to a unique face, rather than that predicted from a canonical model.

Figure I in the patent shows the locations of the landmark points detected from the ASM algorithm, where there are 76 landmark points on the face[12]. Below, in Figure 6, is an output image from the program outlined in Part II, showing the same placement of these 76 landmark points on an image from my own studies. This image serves to illustrate the placement of the

[12] Elena Shaburova. "Methods for real time video processing for changing proportions of an object in the video." US20150221118A1." Page 2.

facial landmarks in the Snapchat model, as that model was used as a template for this depiction.

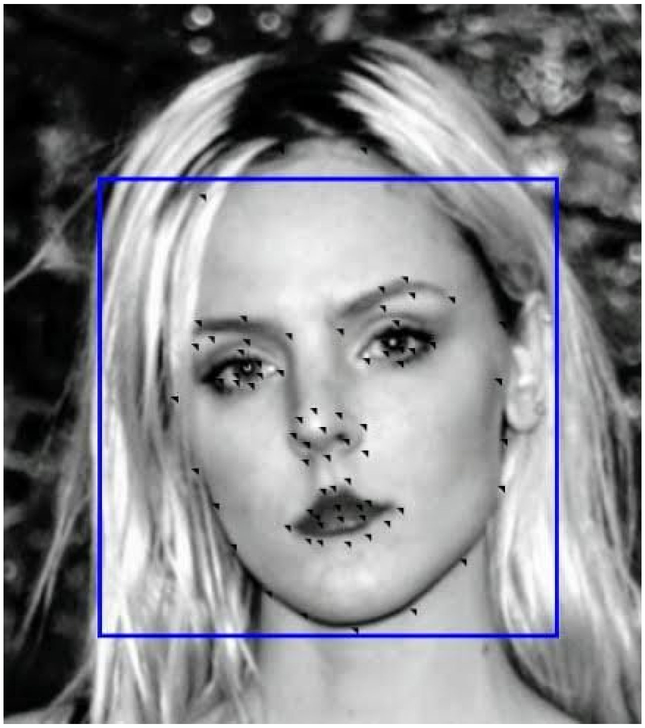


Figure 6: The 76 facial landmark points on input image, following template proposed in US Patent 20150221118A1 by Shaburova

Section 3: Triangulation/ Mesh Generation

The patent documentation only refers to the term "mesh," and does not make use of the term "triangulation," with respect to the point set of landmark locations. The mesh, however, is comprised of triangular faces and qualifies as a triangular mesh. Nonetheless, the terminology of the patent makes clear that the application of a mesh is a critical component of the process.

Notably, the patent refers to the utilization of Candide-3, which, according to Jörgen Ahlberg of Image Coding Group in the Department of Electrical Engineering, Linköping University, Sweden, is "a parameterised face mask specifically developed for model-based coding of human faces. Its low number of polygons (approximately 100) allows fast reconstruction with small computing power."[13]

The use of Candide-3 marks an important distinction in this technology: utilizing Candide-3 as a framework could theoretically optimize performance. Any changes from the original Candide-3 model would be simple local reconfigurations on a pre-existing framework, which could lead to a lower computational cost overall.

In the case of Candide-3, the application of the facial model allows for textured mapping, an important component of the face morphing that is common to many instances of the Snapchat Lens. This does indicate, however, that there is no triangulation of the original detected point set. Any triangulation rendered would be a resultant reconfiguration of the mesh to suit the specificity of an individual face.

Section 4: Mesh Refinement

It must be noted that the Candide model comes in several flavors: 75 landmark points in the original model, 79 landmark points in the "de-facto standard model," 160 landmark points in Candide-2, and 113 landmark points in Candide-3.[14] Any variations from those established

[13] Jörgen Ahlberg. "Candide-3 - An Updated Parameterised Face." Linköping University, Sweden. January 2001. Page 1.
[14] Jörgen Ahlberg. "Candide-3 - An Updated Parameterised Face." Linköping University, Sweden. January 2001. Page 1-2.

models indicates that reconfiguration, likely retriangulation, did become an integral process in the application and adaptation of the model.

As will be shown in Part II, the 76 initial landmark points and the 96 final vertices in the facial mesh in the Snapchat Lens, indicate that re-triangulation and reconfiguration, utilizing additional Steiner points, were an important procedural step in the generation of the final mesh.

II. Adapting the Snapchat Lens

Technologies Utilized

The majority of the code was written in processing.js, and run in the Processing IDE. Initially, this was a decision based off of the reputation of Processing as a programming language for the visual arts. However, as the scope of this program expanded, the Processing language, IDE and libraries provided more of a hindrance than a support; this was due to the limited availability of built-in functions for computer vision within the language and its associated libraries.

The OpenCV library for Processing has significantly fewer functions for processes i.e. little to no built-in functionality for processes such as Edge Detection, Corner Detection, Facial Landmark Detection and many other procedural elements utilized by Snapchat for their Lens technology. While the OpenCV library does provide powerful tools, which were an invaluable resource in this project, the OpenCV Processing library has minimal wrapped functionality.[15] Building more

[15] Greg Borenstein. "ReadMe : OpenCV for Processing." https://github.com/atduskgreg/opencv-processing

complex tools in Processing requires the utilization of unwrapped OpenCV Java objects functions which, while not necessarily overly difficult to do, was frequently more time-intensive than implementing functions from scratch in Processing.

I utilized the ToxicLibs library for Processing for the Voronoi/Delaunay renderings. The library is generally good as a computational geometry library, though its implementation for the Sutherland-Hodgman clipper left something to be desired.

As such, many modules were either implemented from scratch, or were significantly adapted and repurposed to serve similar functionalities while also fitting to the time/emotional labor constraints related to the enormity of the task which I self-allocated.

Procedural Steps

In attempting to implement a naive recreation of the Snapchat triangulation for the scale of this project, it was necessary to understand which processes and procedures present in the original technology could be exploited in parallel in this context. Thus, the chosen processes were facial detection (section 1) and triangulation/mesh generation(section 3 and 4, respectively). Facial landmark detection was adapted in this context to better suit the need to accurately replicate the Snapchat triangulation, as is described in Section 2.

Section 1: Facial Detection

For this module, I utilized the OpenCV implementation of the Viola-Jones algorithm to generate an initial Bounding Box for the face. The implementation made several false-positive detections on faces, so I implemented a check on the minimum size of bounding boxes for detected face regions (see Figure 5). This was also in keeping with the nature of the Snapchat Lens: a user's face must take up a significant portion of the screen when taking a photograph in the app, otherwise a face will not be detected; in other words, a user must be relatively close to the camera, as to take up approximately ½-⅔ of the screen— based on experimentative use of the app and its Lens functionality.

Section 2: Facial Landmark Detection

(See Part III for reasoning and methodologic choices).

In referring to Shaburova and the methodology outlined for the detection of facial landmarks, "The mean shape is the mean of the aligned training shapes (which in the present disclosure are manually landmarked faces)."[16] Utilizing the figures from the patent documentation as templates, I employed the process of manually landmarking the photo of my own face with the same number and location of facial landmarks. This allowed me to have an accurate basis from which to render the triangulations for analysis.

(See Figure 6.)

---

[16] Elena Shaburova. "Method for real time video processing for changing proportions of an object in the video." US20150221118A1. Page 3.

Section 3: Triangulation of Facial Landmarks

Utilizing the ToxicLibs library in Processing, I rendered the Delaunay triangulation of the input set of landmark points (the manually landmarked points). Figure 7 shows the resulting image.

Figure 7: Delaunay Triangulation of initial point set of facial landmarks

This output image provided a basis for comparative analysis to the original screenshot of the Snapchat triangulation. The two images are considerably different: from a purely visual analysis, the two triangulations are not equal in number of edges, vertices or triangular faces, most notably, in regions such as the forehead or cheeks.

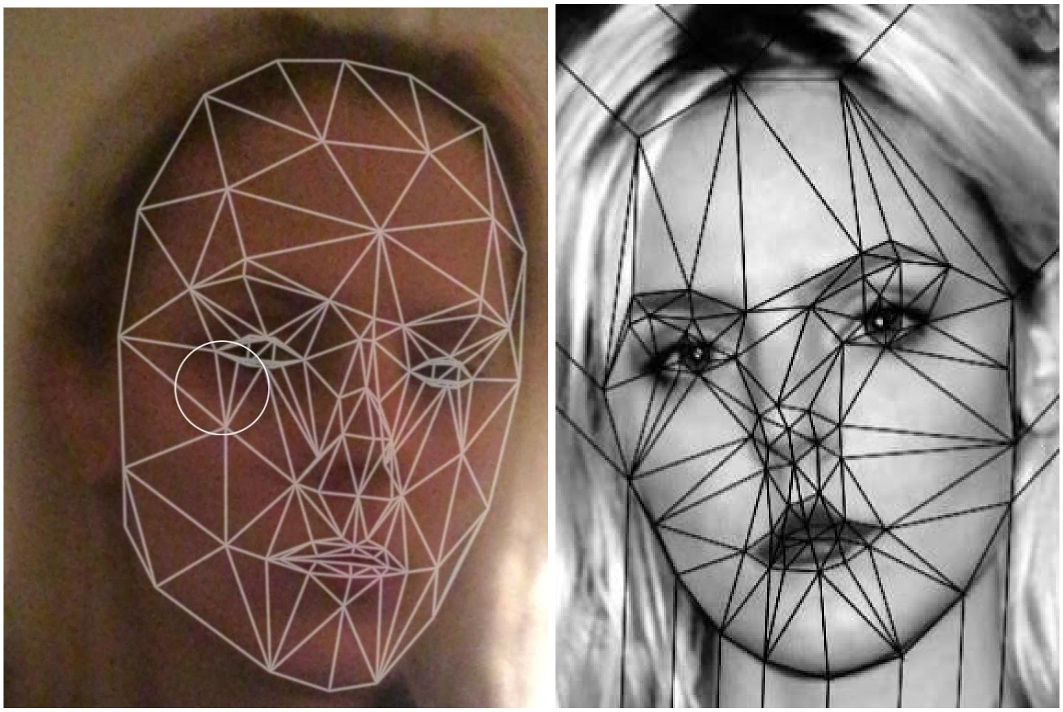


Figure 8: Screenshot of Snapchat triangulation (left) and output image from author's implementation, showing Delaunay Triangulation of initial point set of 76 facial landmark points (right)

This visual analysis rendered a conclusivity about the Snapchat triangulation: the triangulation is not Delaunay, nor is it a retriangulation of a Delaunay triangulation, with respect to the original point set. If the Snapchat triangulation was simply a retriangulation of an initial Delaunay triangulation, utilizing edge flips or point moves, then the resulting triangulation would have the same number of vertices, the name number of edges. This retriangulation, in theory would also not be computationally expensive. In considering the paper "Reconfiguring Triangulations with Edge Flips and Point Moves" by Greg Aloupis et al, we know that "with O(n log n) edge flips and point moves, we can transform any geometric near-triangulation on n points to any other geometric near-triangulation on n possibly different points,"[17] -- near-triangulations being defined as not having a triangular outer face. Yet in the case of the Snapchat triangulation, in this context, we can see (from a visual analysis) that our number of points, n, changes between the initial facial landmark detection phase and the final mesh generation phase.

Thus, the final Snapchat triangular mesh is not a resultant retriangulation but a triangulation on a set of points of size (n + c), where n is the number of initial points, c is a constant (in the case of this experiment, 20). In following this, Steiner points must be required to create the final triangulation.

Section 4: Addition of Steiner Points

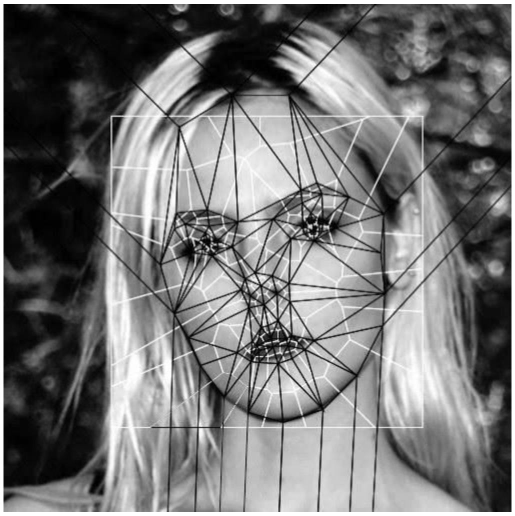
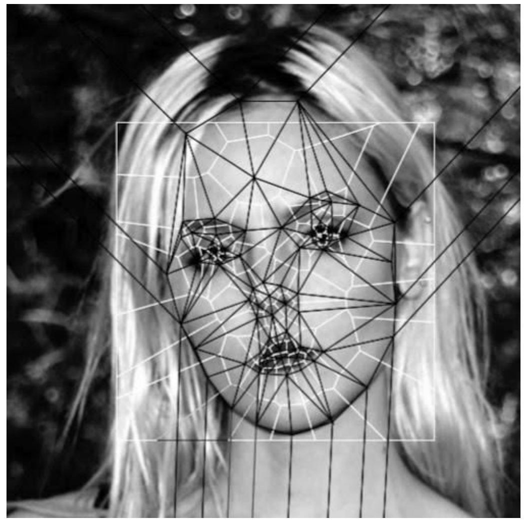
To recreate the final Snapchat triangulation on my own implementation, Steiner points were incrementally added to each portion of the face. Each insertion of a Steiner point triggered a local, but not global, retriangulation.

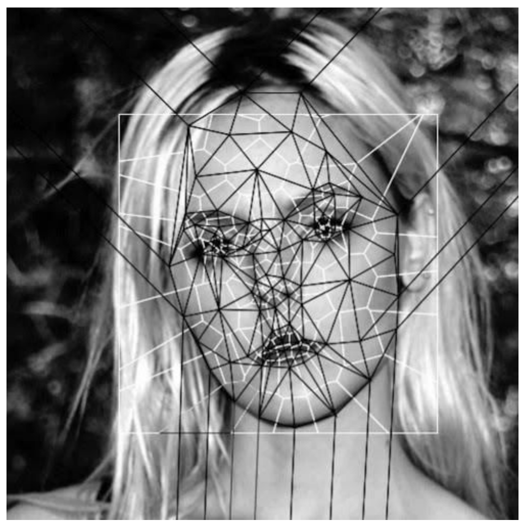
From the initial construction of the Voronoi diagram and Delaunay Triangulation, with a lower bound $\Omega($ nlogn$)$, each added Steiner point can be inserted by utilizing the structure of the old cell into which it is inserted; this gives a lower bound of $O(|c|)$, where c = size of cell,[18] for the retriangulation after the insertion of each Steiner point.

[17] Greg Aloupis, Prosenjit Bose and Pat Morin. "Reconfiguring Triangulations with Edge Flips and Point Moves." Page 2.
[18] Greg Aloupis. "163 - Chapter 10b." Lecture Notes. COMP163: Tufts University. 2016. http://www.cs.tufts.edu/comp/163/lectures/163-chapter10b.pdf'

Figure 7 shows six stages of this process of incremental insertion of points. In total, there were 20 steiner vertices required to modify the original mesh to its final state.

Figure 9: Voronoi Diagram (white edges) and Delaunay triangulation (black edges) of the point set at six stages of the incremental insertion of Steiner points

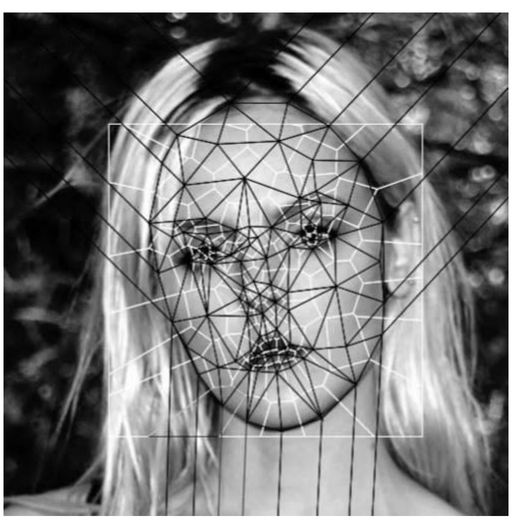
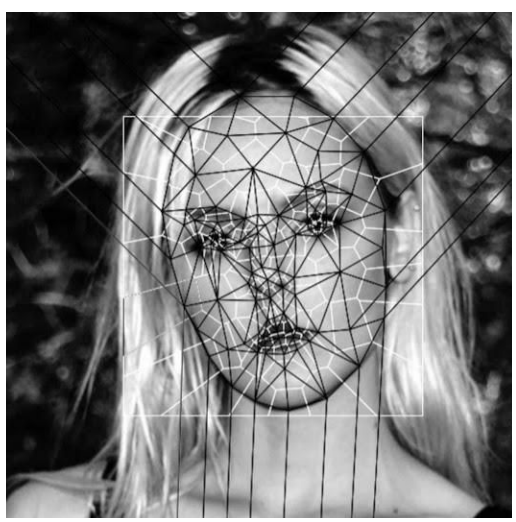



Figure 10 shows the final triangulation.

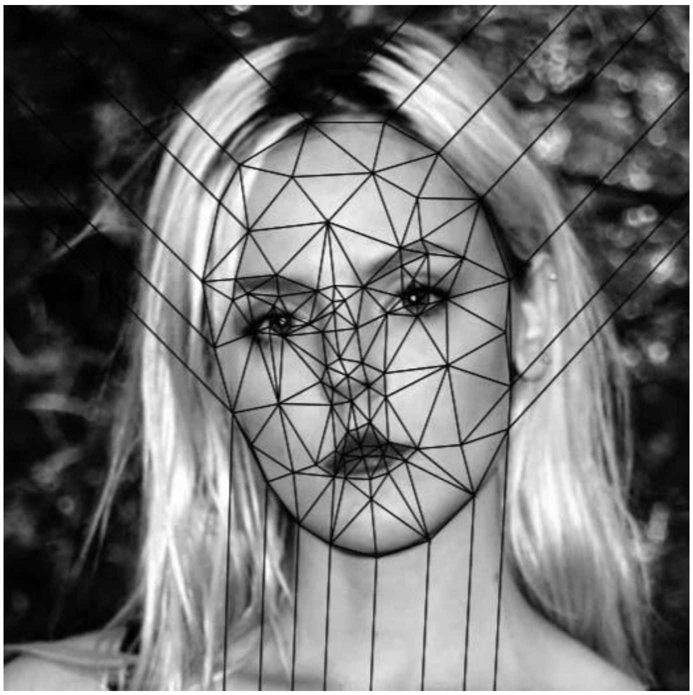


Figure 10: Final triangulation to approximate the Snapchat triangulation, rendered after addition of 20 Steiner points

Results

Outcome 1

The Snapchat triangulation makes heavy use of Steiner points to generate the final mesh; this is done by refining an original template generated from the original Candide-3 model, after using an ASM approach to detect an initial set of facial landmark points.

Outcome 2

In this isolated experiment, the similarity between the resulting image from the procedure and the Snapchat triangulation used as a template, appears to show that the Snapchat triangulation is a Delaunay triangulation, albeit a conforming Delaunay triangulation, due to its use of Steiner points.

Part III: Analysis

Nearest-Neighbor Classification and Facial Landmark Detection

At the onset of this project, a major goal was to develop a better system for landmark detection. Inspired by the work of Boiman et al, in the paper "In Defense of Nearest-Neighbor Classification," I was interested in the possibility of forgoing Machine Learning technologies for the purpose of location facial landmarks within an image.

My interest in algorithmic transparency was the leading factor in this search for an alternative method; understanding that there are complex systems of racism, sexism, and other forms of institutional oppression built into many algorithms, specifically those that rely on a training set whose very selection reveals the biases of its creators, I was curious as to whether or not a purely geometric approach to the problem could serve as a solution, to combat the bias in CV technologies such as facial detection.

Again, returning to the paper by Cootes, there is an acknowledgement of the limitations of the ASM methodology: "In addition, it should be noted that the accuracy to which [Active Shape Models] can locate a boundary is constrained by the model. The model can only deform in ways observed in the training set. If the object in an image exhibits a particular type of deformation not present in the training set, the model will not fit to it."

Ian Tucker's piece "'A white mask worked better': why algorithms are not colour blind"[19] in The Guardian, and Claire Garvie and Jonathan Frankle's piece "Facial Recognition Software Might Have a Racial Bias Problem"[20] in the Atlantic are two examples of how the limitations of training sets, and in built-in bias in algorithmic design, can have dramatic, negative societal impact.

Utilizing Voronoi for location-based nearest-neighbor search and refinement of input data

In following with the model outlined by Boiman, one could consider improvements to the methodology utilized in the Snapchat technology for refinement from its initial set of facial landmark points. One example of this would be an application of Voronoi diagrams for pruning input data, as seen in the documentation for the US Patent of the technology "Method and System for Facilitating the Selection of Icons." The technology relies on the Voronoi diagram structure to determine the nearest-neighbor icon to the input (x,y) location of a cursor on a screen.

As stated in the section entitled "Best Mode for Carrying Out the Invention," the inventor states the methodology for choosing between multiple icons if one is to fall on the … "If an icon

---

[19] Ian Tucker. "'A white mask worked better': why algorithms are not colour blind." The Guardian. 28 May, 2017.
[20] Claire Garvie and Jonathan Frankle. "Facial Recognition Software Might Have a Racial Bias Problem." The Atlantic. April 7, 2016.

is on the boundary of two or more polygons, then any one of of the polygons may be selected based on predefined criteria. One example of predefined criteria would be a probability of use as determined from past history of use…”[21]

This research opens the possibility for additional critical examination of the Snapchat triangulation. Rather than using the Candide-3 model, could the original 76 landmark points, detected from an ASM approach, simply be refined using a Voronoi-based approach? The generation of the Voronoi diagram from the original 76 landmark points could be later utilized for the retriangulation for the final mesh generation. In following with what is outlined in Part II, Part 4, local updates to an Voronoi diagram/Delaunay triangulation could be relatively low-cost computations to generate the end result mesh.

Triangulation and Refinement

The utilization of Steiner points is reflective of the goals of the technology itself. The title of US20150221118A1 is, after all, “Method for real time video processing for changing proportions of an object in the video.” The goal of the technology in the context of Snapchat is not simply for facial tracking, or for identification in images. The Snapchat Lens is built around concepts of using Augmented Reality to morph a user’s image in real-time.

[21] Bertrand M. Grossman et al. “Method and System for Facilitating the Selection of Icons.” US Patent 5564004A Page 7.

The patent outlines several ways in which modification of the face can take place: "changing proportions of the object defined by the user" by altering thinness/fatness of the face, nose width, eye width.[22] Considering these operations, the placement of the Steiner points underscores the necessity for functionality within a desired aesthetic; the decision to add specific points is dependent on which features are deemed important to change, and in what ways they should be changed, according to the algorithmic design of the technology, i.e. which morphing options are available to a user.

Conclusion

An article from the New Yorker's December 18 & 25, 2017 issue extrapolates the subtle but pervasive power and influence exerted by computer vision apps in present society. In the piece, entitled "China's Selfie Obsession," reporter Jiayang Fan connects present beauty trends in China to physical characteristics that are enhanced, adjusted or eliminated in popular selfie filter apps available on the Meitu platform:

"Now people have shifted to what he described as "Euro-American wave," a tacit acknowledgment of the fact that the apps have a way of making people look more Western—for instance, by replacing single eyelids, which are typical, though not universal, among East Asians, with a double eyelid fold. There is even a new filter on BeautyPlus called "mixed blood," used to

[22] Elena Shaburova. "Method for real time video processing for changing proportions of an object in the video." US20150221118A1. Page 3 and 6.

achieve a Eurasian appearance. Earlier this year, there was a spate of outrage on social media after international users pointed out that increasing beauty levels in the app invariably resulted in a lightening of skin color…”[23]

If the societal implications of selfie filters were not present when the Snapchat Lens technology first debuted, there can be no doubt that the multitude of such apps, i.e. Meitu, have increased the necessity for critical examination of their algorithmic design. While the addition of a Steiner point may seem innocuous in the grand scheme of a mesh generation -- adjusting 76 input points to 96 points -- the already-present public reaction of filter effects shows differently. The placement of Steiner points, the meshes generated from the Snapchat Lens or other filter-based apps, are reflective of the facial modifications that are deemed desirable or, even necessary, by designers of these technologies. While there cannot be one singularity of moral judgement to evaluate these choices -- after all, “beauty” is subjective -- it is absolutely necessary that they come with an increased consciousness on the part of both the user and the engineer. Methodologies in the geometric reconfigurations of a face demand an algorithmic transparency.

A Note on This Project

This project became a year-long investigation into the minutiae of image processing in relation to one specific triangulation which captured my attention last year. The difficulty of this project centered around synthesizing the research I did (months of reading research papers, articles, etc.) and in applying this knowledge to create effective tools for analysis. Many tools

[23] Jianyang Fan. “China’s Selfie Obsession.” The New Yorker: December 18 & 25, 2017.

developed (edge detection, corner detection, histogram visualizations, etc.) were not included in this final paper but were a considerable part of the research process. Some of the difficulty of the project centered around having to learn about image manipulation in relation to topics such as eigenvalues, eigenvectors, matrix operations, etc., without having taken a course in Linear Algebra (I made heavy use of Youtube tutorials to catch myself up on this).

The work of this project is still open-ended. A very affirmative acknowledgment is made that my experimentation and analysis was centered around a model of one singular image, and that no concrete conclusivity can be derived from its findings, simply due to the very limitations of these experiments. I do not presume in any way to have an empirical solution for improvements to any issues with the Snapchat technology, nor do I doubt that the technology itself is a remarkable and incredibly effective, if not unparalleled, commercial tool for Augmented Reality image processing.

I do believe that this paper offers some interesting insights as to the links between Computational Geometry and an incredibly popular social media app. I also believe that this paper poses important questions about the methodology of the algorithms utilized in the mesh generation for Snapchat; these questions could extend to algorithmic design in software engineering today, and the choices made by engineers in trying to balance optimality with performance and accuracy.

References

Ahlberg, Jörgen. “Candide-3: An Updated Parameterised Face.” Image Coding Group: Dept. of Electrical Engineering, Linköping University, Sweden. January 2001. http://www.bk.isy.liu.se/publications/LiTH-ISY-R-2326.pdf

Alexa, Marc et al. “Computing and Rendering Point Set Surfaces.” Institute of Electrical and Electronics Engineers. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). January 2013. http://www.sci.utah.edu/~shachar/Publications/crpss.pdf

Aloupis, Greg, Prosenjit Bose and Pat Morin. “Reconfiguring Triangulations with Edge Flips and Point Moves.” Algorithmica, vol.47, issue 4 (May 2007), p.367-378. Also in the LNCS proceedings of Graph Drawing 2004 (vol.3383, p.1-11).

Aloupis et al. “Meshes Preserving Minimum Feature Size.” CRM Documents, vol. 8, Centre de Recerca Matemàtica, Bellaterra (Barcelona), 2011 Page 205 - 208. http://www3.uah.es/egc2011/files/ActasEGC2011.pdf

Aloupis, Greg. “163 - Chapter 10b.” Lecture Notes. COMP163: Tufts University. 2016. http://www.cs.tufts.edu/comp/163/lectures/163-chapter10b.pdf”

Aurenhammer, Franz and Rolf Klein. “Voronoi Diagrams.”
http://www.pi6.fernuni-hagen.de/downloads/publ/tr198.pdf

Bélair, François. “Everything You Always Wanted to Know About Alpha Shapes But Were Afraid to Ask.”
http://cgm.cs.mcgill.ca/~godfried/teaching/projects97/belair/alpha.html

Bern, Marshall, Jonathan R. Shewchuk, and Nina Amenta. “Chapter 29: Triangulations and Mesh Generation,” Preliminary version (July 28, 2017). To appear in the “Handbook of Discrete and Computational Geometry,” J.E. Goodman, J. O'Rourke, and C. D. Tóth (editors), 3rd edition, CRC Press, Boca Raton, FL, 2017. https://www.csun.edu/~ctoth/Handbook/chap29.pdf

Boiman, Oren, Eli Shechtman and Michal Irani. “In Defense of Nearest-Neighbor Based Image Classification.” IEEE Conference on Computer Vision and Pattern Recognition. June 2008.
http://www.wisdom.weizmann.ac.il/~irani/PAPERS/InDefenceOfNN_CVPR08.pdf

Bonnington, Christina. “Prepare for More Dancing Hotdogs.” December 16, 2017. Slate.
http://www.slate.com/articles/technology/technology/2017/12/snapchat_releases_augmented_reality_lens_studio_meaning_memes_are_coming.html

Cade, DL. “A Look at How Snapchat’s Powerful Facial Recognition Works.” PetaPixel. 30 June 2016.
https://petapixel.com/2016/06/30/snapchats-powerful-facial-recognition-technology-works/

Chazelle, Bernard “Approximation and Decomposition of Shapes” in “Advances in Robotics 1: Algorithmic and Geometric Aspects of Robotics,” (J.T. Schwartz, C.K. Yap, eds), Lawrence Erlbaum Associates (1987), 145-185.
https://www.cs.princeton.edu/~chazelle/pubs/ApproxDecompShapes.pdf

Chazelle, Bernard. “Computational Geometry: A Retrospective.” Appears in “Computing in Euclidean Geometry” 2nd edition (D.-Z. Du and F. Hwang, eds), World Scientific Press (1995), 22-46. Prelim. version in STOC 1994 (invited).
https://www.cs.princeton.edu/~chazelle/pubs/ComputGeomRetrospective.pdf

Chazelle, Bernard.“Finding a Good Neighbor, Near and Fast.” CACM 51 (2008), 115.
https://www.cs.princeton.edu/~chazelle/pubs/cacm08.pdf

Chazelle, B., C. Seshadhri, Journal ACM 58 (2011), 1-32. Prelim. version in SoCG 2006.
https://www.cs.princeton.edu/~chazelle/pubs/Onlinerecons11.pdf

Chazelle, Bernard, O. Devillers, F. Hurtado, M. Mora, V. Sacristan, M. Teillaud. “Splitting a Delaunay Triangulation in Linear Time.” Algorithmica 34 (2002), 39-46. Prelim. version in ESA 2001

https://www.cs.princeton.edu/~chazelle/pubs/splitting.pdf

Chazelle, Bernard and D.P. Dobkin, “Optimal Convex Decompositions” in “Computational Geometry” (G.T. Toussaint, ed), North-Holland (1985), 63-133. Prelim. version in STOC 1979.

https://www.cs.princeton.edu/~chazelle/pubs/OptimalConvexDecomp.pdf

Chazelle, Bernard, D. Liu, A. Magen, “Sublinear Geometric Algorithms” in SIAM J. Comput. 35 (2006), 627-646. Prelim. version in STOC 2003.

https://www.cs.princeton.edu/~chazelle/pubs/sublinearSIAM.pdf

Chazelle, Bernard. “Triangulating a Simple Polygon in Linear Time.” Department of Computer Science, Princeton University, Princeton NJ. Discrete and Computational Geometry 6: 485-524 (1991).

https://www.cs.princeton.edu/~chazelle/pubs/polygon-triang.pdf

Cootes, Tim. “An Introduction to Active Shape Models.” Appears as Chapter 7: ”Model-Based Methods in Analysis of Biomedical Images” in “Image Processing and Analysis”, Ed.R.Baldock and J.Graham,Oxford University Press, 2000. Page 223-248.

http://www2.compute.dtu.dk/courses/02511/docs/asm_overview.pdf

Deo, Dhanannjay and Dibakar Sen. "Mesh Processing for Computerized Facial Anthropometry."Journal of Computing and Information Science in Engineering: Vol. 10, Issue 1. March 10, 2010. doi:10.1115/1.3330420 https://computingengineering.asmedigitalcollection.asme.org/mobile/article.aspx?articleid=1401813

Deniz, Oscar, G. Bueno, J. Salido, and F. De la Torre. "Face Recognition Using Histogram of Oriented Gradients." Pattern Recognition Letters: Volume 32, Issue 12. 1 September 2011. Page 1598-1603. DOI=http://dx.doi.org/10.1016/j.patrec.2011.01.004 https://www.researchgate.net/publication/220646162_Face_recognition_using_Histograms_of_Oriented_Gradients

Eberly, David. "Skeletonization of 2D Binary Images." Geometric Tools, Redmond WA 98052. Created: June 7, 2001. Last Modified: March 2, 2008. https://www.geometrictools.com/Documentation/Skeletons.pdf

Fisher, Robert, Simon Perkins, Ashley Walker and Erik Wolfart. "Histogram Equalization." The Hypermedia Image Processing Reference. 2003. http://homepages.inf.ed.ac.uk/rbf/HIPR2/histeq.htm

Fisher, Robert, Simon Perkins, Ashley Walker and Erik Wolfart. "Intensity Histogram."

The Hypermedia Image Processing Reference. 2003. http://homepages.inf.ed.ac.uk/rbf/HIPR2/histgram.htm

Fisher, Robert, Simon Perkins, Ashley Walker and Erik Wolfart. "Unsharp Filter." The Hypermedia Image Processing Reference. 2003. http://homepages.inf.ed.ac.uk/rbf/HIPR2/unsharp.htm

Fisher, Robert, Simon Perkins, Ashley Walker and Erik Wolfart. "Skeletonization/ Medial Axis Transform." The Hypermedia Image Processing Reference. 2003. http://homepages.inf.ed.ac.uk/rbf/HIPR2/skeleton.htm

Fisher, Robert, Simon Perkins, Ashley Walker and Erik Wolfart. "Roberts Cross Edge Detector." The Hypermedia Image Processing Reference. 2003. http://homepages.inf.ed.ac.uk/rbf/HIPR2/roberts.htm

Fisher, Robert, Simon Perkins, Ashley Walker and Erik Wolfart. "Canny Edge Detector." The Hypermedia Image Processing Reference. 2003. http://homepages.inf.ed.ac.uk/rbf/HIPR2/canny.htm

Fisher, Robert, Simon Perkins, Ashley Walker and Erik Wolfart. "Sobel Edge Detector." The Hypermedia Image Processing Reference. 2003. http://homepages.inf.ed.ac.uk/rbf/HIPR2/sobel.htm

Fisher, Robert, Simon Perkins, Ashley Walker and Erik Wolfart. "Gaussian Smoothing." The Hypermedia Image Processing Reference. 2003.
http://homepages.inf.ed.ac.uk/rbf/HIPR2/gsmooth.htm

Fisher, Robert, Simon Perkins, Ashley Walker and Erik Wolfart. "Gaussian Smoothing." The Hypermedia Image Processing Reference. 2003.
http://homepages.inf.ed.ac.uk/rbf/HIPR2/gsmooth.htm

Fong, Joss and Dion Lee. "Snapchat filters: the engineering behind augmented-reality selfies." Vox. 28 June 2016.
https://www.vox.com/2016/6/28/12046792/how-snapchat-filters-work

Fong, Joss and Dion Lee. "How Snapchat's Filters Work." Vox Observatory: Vox. 28 June 2016.
https://www.youtube.com/watch?v=Pc2aJxnmzh0&feature=youtu.be

Fortune, Steven. "Chapter 27: Voronoi Diagrams and Delaunay Triangulations," Preliminary version (July 17, 2017). To appear in the "Handbook of Discrete and Computational Geometry," J.E. Goodman, J. O'Rourke, and C. D. Tóth (editors), 3rd edition, CRC Press, Boca Raton, FL, 2017.https://www.csun.edu/~ctoth/Handbook/chap27.pdf

Freund, Yoav and Robert E. Schapire. "A Short Introduction to Boosting." AT&T Labs Research Shannon Laboratory. Florham Park, New Jersey. Journal of Japanese Society for Artificial Intelligence, 14(5):771-780, September, 1999. (In Japanese, translation by Naoki Abe.) https://cseweb.ucsd.edu/~yfreund/papers/IntroToBoosting.pdf

Garvie, Claire and Jonathan Frankle. "Facial-Recognition Software Might Have a Racial Bias Problem." The Atlantic. April 7, 2016. https://www.theatlantic.com/technology/archive/2016/04/the-underlying-bias-of-facial-recognition-systems/476991/

Goodman, Rob and Jimmy Soni. "The bit bomb: It took a polymath to pin down the true nature of 'information'. His answer was both a revelation and a return." ed. Sally Davies. Aeon Magazine. 30 August, 2017. https://aeon.co/essays/how-a-polymath-transformed-our-understanding-of-information?utm_source=Aeon+Newsletter&utm_campaign=66b269deae-EMAIL_CAMPAIGN_2018_01_15&utm_medium=email&utm_term=0_411a82e59d-66b269deae-69405713

Grossman, Bertrand M., James G. McLean, Clifford A. Pickover, Michael S. Schwartz, Daniel J. Winarski. "Method and system for facilitating the selection of icons." US Patent 5564004A. Filed April 14, 1994 and issued October 8, 1996. https://patents.google.com/patent/US5564004A/en?oq=5564004

Herpers, R. and G. Sommer "An Attentive Processing Strategy for the Analysis of Facial Features." (1998) In: Wechsler H., Phillips P.J., Bruce V., Soulié F.F., Huang T.S. (eds) Face Recognition. NATO ASI Series (Series F: Computer and Systems Sciences), vol 163. Springer, Berlin, Heidelberg. https://link.springer.com/chapter/10.1007/978-3-642-72201-1_26

Hu, Peiyun and Deva Ramanan. "Finding Tiny Faces. " Robotics Institute Carnegie Mellon University. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). July 2017. http://openaccess.thecvf.com/content_cvpr_2017/papers/Hu_Finding_Tiny_Faces_CVPR_2017_paper.pdf

Husen, Bill. "Markov Processes." Department of Mathematics: Ohio State Univeristy. Math 571. https://people.math.osu.edu/husen.1/teaching/571/markov_1.pdf

Kimberling, Clarke. "Steiner Point." Encyclopedia of Triangle Centers. http://faculty.evansville.edu/ck6/tcenters/class/steiner.html

Lischinski, Dani. "Incremental Delaunay Triangulation" Graphics Gems IV. Paul S. Heckbert (Ed.). Academic Press Prof., Inc., San Diego, CA, USA. 1994. Page 47-59. http://www.karlchenofhell.org/cppswp/lischinski.pdf
doi: https://dl.acm.org/citation.cfm?id=180900

Mallick, Satya. "Histogram of Oriented Gradients." Learn OpenCV. December 6, 2016. http://www.learnopencv.com/histogram-of-oriented-gradients/

Mallick, Satya. "Image Recognition and Object Detection: Part 1." Learn OpenCV. November 14, 2016. http://www.learnopencv.com/image-recognition-and-object-detection-part1/

Martin, Janzura. "Image Processing, Markov Chain Approach." (1996) In: Prat A. (eds) COMPSTAT. Physica-Verlag HD. DOI: https://doi.org/10.1007/978-3-642-46992-3_8 https://link.springer.com/chapter/10.1007/978-3-642-46992-3_8#citeas

Mount, David M. "Kirkpatrick's Planar Point Location." Lecture Notes: CMSC 754 Computational Geometry. Department of Computer Science University of Maryland, Fall 2005. Page 21-25. http://www.cs.tufts.edu/comp/163/davidM/DM-kirkpatrick-point-location.pdf

Mount, David M. "Lecture 10: Voronoi Diagrams and Fortune's Algorithm." Lecture Notes: CMSC 754 Computational Geometry. Department of Computer Science University of Maryland, Fall 2005. Page 21-25. http://www.cs.tufts.edu/comp/163/davidM/DM-voronoi-delaunay.pdf

Mount, David M. “Monotone Polygons.” Lecture Notes: CMSC 754 Computational Geometry. Department of Computer Science University of Maryland, Fall 2005. Page 21-25. http://www.cs.tufts.edu/comp/163/davidM/DM-monotone.pdf

Osada, R., T. Funkhouser, B. Chazelle, D.P. Dobkin, “Shape Distributions.” ACM Trans. Graphics 21 (2002), 807-832. Prelim. version in SMI 2001. https://www.cs.princeton.edu/~chazelle/pubs/ShapeDistributions.pdf

Powell, Victor. “Image Kernels: Explained Visually.” Explained Visually. http://setosa.io/ev/image-kernels/

Powell, Victor. “Eigenvectors and Eigenvalues: Explained Visually.” Explained Visually. http://setosa.io/ev/eigenvectors-and-eigenvalues/

Razafindrazaka, Faniry Harijaona. African Institute for Mathematical Sciences (AIMS). “Delaunay Triangulation Algorithm and Application to Terrain Generation.” Supervised by: Doctor J W Sanders. International Institute for Software Technology: United Nations University, Macao. 22 May 2009. http://page.mi.fu-berlin.de/faniry/files/faniry_aims.pdf

Rennes, Hervé Jégou Inria and Andrew Zisserman, Dept. of Engineering Science, University of Oxford. “Triangulation embedding and democratic aggregation for image search.” The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). June 2014.

https://www.cv-foundation.org/openaccess/content_cvpr_2014/papers/Jegou_Triangulation_Embedding_and_2014_CVPR_paper.pdf

Rojas, Mario, David Masip, Alexander Todorov, and Jordi Vitria. "Automatic Prediction of Facial Trait Judgments: Appearance vs. Structural Models." August 17, 2011. PLOS ONE 6(8): e23323.doi: https://doi.org/10.1371/journal.pone.0023323 Retrieved from: http://journals.plos.org/plosone/article?id=10.1371/journal.pone.0023323#s2

Ruppert, Jim. "A Delaunay Refinement Algorithm for Quality 2-Dimensional Mesh Generation." NASA Ames Research Center. https://www.nas.nasa.gov/assets/pdf/techreports/1994/rnr-94-002.pdf

Shaburova, Elena. "Method for real time video processing for changing proportions of an object in the video." US Patent 20150221118 A1. Filed June 25, 2014, and issued August 6, 2015. https://patents.google.com/patent/US20150221118A1/en?oq=20150221118

Shamos, Michael Ian. (1978). *Computational Geometry* (Doctoral Dissertation). Yale University. http://euro.ecom.cmu.edu/people/faculty/mshamos/1978ShamosThesis.pdf

Shannon, Claude. "A Mathematical Theory of Communication." A mathematical theory of communication. SIGMOBILE Mob. Comput. Commun. Rev. 5, 1 (January 2001), 3-55. DOI=http://dx.doi.org/10.1145/584091.584093 http://delivery.acm.org/10.1145/590000/584093/p3-shannon.pdf?ip=130.64.25.59&id=584093&acc=ACTIVE%20SERVICE&key=AA86BE8B6928DDC7%2E4579F4D1C4C67060%2E4D4702B0C3E38B35%2E4D4702B0C3E38B35&__acm__=1516843499_f86ae40a089cc0e3cd5b8372667f6df8

Shewchuk, Jonathan Richard. "Delaunay Refinement Algorithms for Mesh Generation." Department of Electrical Engineering and Computer Science: University of California at Berkeley. May 21, 2001. https://people.eecs.berkeley.edu/~jrs/papers/2dj.pdf

Shi, Jianbo and Carlo Tomasi. "Good Features to Track." IEEE Conference on Computer Vision and Pattern Recognition (CVPR94), Seattle, June 1994. http://www.ai.mit.edu/courses/6.891/handouts/shi94good.pdf

Shiffman, Daniel. "Face It: A repository of Processing examples for ITP fall workshop about face detection, recognition, and miscellaneous tracking methods." https://github.com/shiffman/Face-It

Svozil, K. (1994) Extrinsic-Intrinsic Concept and Complementarity. In: Atmanspacher H., Dalenoort G.J. (eds) Inside Versus Outside. Springer Series in Synergetics, vol 63. Springer, Berlin, Heidelberg. DOI: https://doi.org/10.1007/978-3-642-48647-0_15 https://link.springer.com/chapter/10.1007/978-3-642-48647-0_15

Syed, Asem. September 20, 2017. “How do Snapchat filters work?” Technobyte. https://www.technobyte.org/2016/11/snapchat-filters-work/

“Templates.” Lens Studio by Snap Inc. Snap Inc. 2017. https://lensstudio.snapchat.com/templates/

Toussaint, Godfried. “The Relative Neighbourhood Graph of a Finite Planar Set.” Published in Pattern Recognition, Vol. 12, 1980, pp. 261-268. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.54.6448&rep=rep1&type=pdf

Tucker, Ian. “‘A white mask worked better’: why algorithms are not colour blind.” Computing: The Observer. The Guardian. 28 May 2017. https://www.theguardian.com/technology/2017/may/28/joy-buolamwini-when-algorithms-are-racist-facial-recognition-bias

Vandevenne, Lode. “Lode’s Computer Graphics Tutorial: Image Filtering.” Lode’s Computer Graphics Tutorial. 2004. http://lodev.org/cgtutor/filtering.html

Vantomme, Jan. “Voronoi Paintings with ToxicLibs.” Vormplus. 28 July, 2011. http://vormplus.be/blog/article/voronoi-paintings-with-toxiclibs

Viola, Paul and Michael J. Jones. “Robust Real-Time Face Detection.” International Journal of Computer Vision 57(2), 137–154, 2004. Kluwer Academic Publishers. http://www.vision.caltech.edu/html-files/EE148-2005-Spring/pprs/viola04ijcv.pdf

Wang, Chengfeng, and Marina L. Gavrilova. “Delaunay Triangulation Algorithm for Fingerprint Matching.” 2006 3rd International Symposium on Voronoi Diagrams in Science and Engineering. Date of Conference: 2-5 July 2006. Date Added to IEEE Xplore: 12 March 2007. DOI: 10.1109/ISVD.2006.19 http://ieeexplore.ieee.org/document/4124821/

Wolfrum, Philipp, Christian Wolff, Jörg Lücke, and Christoph von der Malsburg. “A Recurrent Dynamic Model for Correspondence-Based Face Recognition.” 13 August, 2008. Preprint, Journal of Vision, 2008, in press. https://fias.uni-frankfurt.de/~wolfrum/papers/WolfrumMalsburg08JoV.pdf