The integration of functional data analysis in conditional image generation

Cédric Beaulac

Simon Fraser University and University of Victoria

April the 20th 2022

Shape and functional data analysis for image generation

Content (image) Generation

Brief introduction to the problem.

What is content generation ?

- A typical example of content generation is art asset generation in video game application.
- The generation allows for infinite content and minimal storage requirements.
- Procedural content generation is the old algorithmic approach for content generation.
- Uses designed models (no learning) to generate specific content.
- Struggle with generalization and is time consuming.
- A model learning from a data set would be helpful.

Image generation: An example

- A typical example is the generation of non-playable character (NPCs) faces.
- We want the faces to look like faces (realistic).
- We want everyone to look different.
- We want control to create region-specific features and control their expression.
- From now on, we consider *faces* to be object we want to generate. x is the face variables and $p(x)$ is the distribution of faces.

Generative Adversarial Networks (GANs)

Generative Adversarial Networks (GANs)

- Fairly recent family of *generative model* introduced by Goodfellow (2014).
- Generative models: learns a distribution $p_g(x)$ given a data set x, allowing us to sample from that distribution.
- Regression learns $p(y|x)$ (supervised), so it is not a generative model.
- Gaussian Mixture Models (unsupervised) learns $p(\pi = k)p(x|\pi = k)$, thus is generative.

Generative Adversarial Networks (GANs)

- We can train very accurate *Discriminative* models (D)
- Better than humans at identifying content of an image. (In medical application for instance)
- GANs emerged from the recent success of these NN for classification.
- We can train a Discriminative model (D) to discriminate real from fake (generated) image.

GANs: The Concept

- If the Discriminative Model D cannot identify that our generated data is indeed fake, then it must be realistic-looking.
- We train the Generative Model G to *fool* the Discriminative Model D.
- It creates an Adversarial dynamic where D learns to discriminate between real and fake data (say image) and G tries to fool D.
- The better D becomes at distinguishing true from fake, the better G has to become at creating realistic images.

GANs: Objective function

- $p_g(x)$ is the generator distribution over $\mathcal{X}$
- Built using a prior input noise $p_z(z)$ $z \in \mathcal{Z}$
- and a mapping $G_{\theta_g} : \mathcal{Z} \to \mathcal{X}$ identified with $G(z; \theta_g)$ in the literature.
- G takes a random noise z as input and outputs an image x.

GANs: Objective function

- We also define $D_{\theta_d} : \mathcal{X} \to [0,1]$ a discriminative function, identified with $D(x;\theta_d)$ in the literature.
- D takes as input an image x and returns the probability that it is a TRUE image.
- We train D by maximizing $\mathbf{E}_{\text{true } x}[\log D(x)] + \mathbf{E}_{\text{fake } x}[\log(1 - D(x))]$

GANs: Objective function

- Simulteanously, we train G to generate *fake* x classified as true by D
- $\min_G \max_D \mathbf{E}_{x \sim p_{\text{data}(x)}}[\log D(x)] + \mathbf{E}_{z \sim p_z(z)}[\log(1 - D(G(z))]$

GANs: results

Figure: Faces generated by original GAN (2014).

GANs: results

Figure: Faces generated by GAN modern architectures (2019).

GANs: Common problems

- (1) Mode collapsing
- (2) Instability
- (3) Randomness

Active Appearance Models (AAM)

Functional and shape data analysis

- Shape, form, appearance, outline, curves.
- Child learns about shape of objects before alphabet, etc..

Figure: *Hey, that's a face right there!*

Functional and shape data analysis

- Shapes are invariant to: translation, rotation and scaling
- Shape registration, quantification of shape similarities, shape classification, etc...
- A challenge of FDA remains: Infinite dimensions of the problem

Functional and shape data analysis

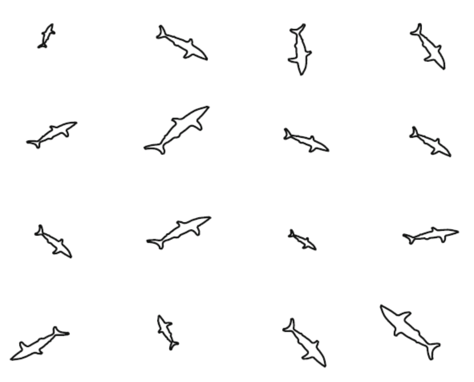


Figure: Same shapes, different scale, rotation, translation.

Active Appearance Models: Introduction

- Active Shape Model (ASM) is a statistical shape model; a 2-dimensional functional model.
- It's a parametric approach to fit and identify shapes in an image.
- The Active Appearance Model (AAM), is an ASM with an additional layer: appearance (coloring).
- It's a parametric approach to fit and identified colored shapes in an image.

Active Appearance Models: Introduction

- Again, let us use faces as our colored shape.
- An AAM can model faces in a parametric manner.
- This is used for facial recognition.
- If we have a parametric representation for faces, maybe we can generate new ones.

ASM: Notations

- The shape is defined by the landmarks.

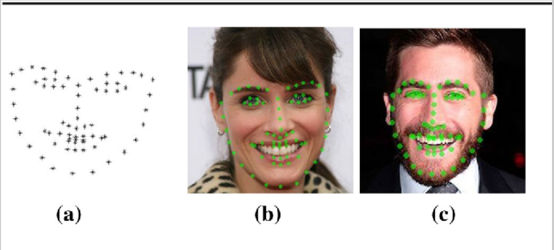


Figure: Face Landmarks

ASM: Notations

- Typically vertex locations of the mesh.
- The mesh is a surface built with triangles, the collection of vertices forms the shape
- Say $\mathbf{s}$ is the shape of an object with v vertices: $\mathbf{s}$ is represented as a vector of size $2v$

$$\mathbf{s} = (x_1, y_1, x_2, y_2, ..., x_v, y_v) \tag{1}$$

- (Already wondering if we can do better with a functional approach)

ASM: Mesh

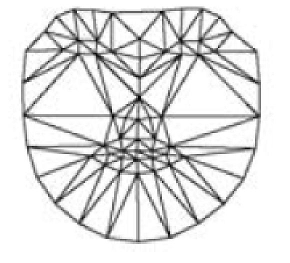


Figure: Face Landmarks

ASM: Shape representation

- For faces, the different shapes account for different individuals, poses, expression, etc...
- Standard assumption: We can express different shapes using a base shape $\mathbf{s}_0$ and a linear combination of k shape vectors $\mathbf{s}_i$.

$$\mathbf{s} = \mathbf{s}_0 + \sum_{i=1}^{k} p_i \mathbf{s}_i, \tag{2}$$

where the coefficients p_i are the shape parameters.

AAM: Learning the shape vectors

- Given a data with landmarks (vertices).

Figure: Data set with landmarks

AAM: Learning the shape vectors

- Given a data with landmarks (vertices).
- We register the shapes: taking into account translation, scaling and rotation
- This is done via a Procrustes Analysis (let's assume it's easy)
- After the Procrustes Analysis we can estimate $\mathbf{s}_0$ and $\mathbf{s}_i$'s
- Standard assumption: the shape vectors are orthonormal
- We apply PCA to the shapes to get the shape vectors $\mathbf{s}_i$

ASM: Shape vectors

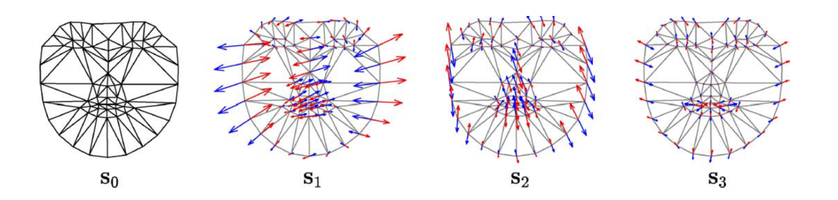


Figure: Base shape $\mathbf{s}_0$ and shape vectors $\mathbf{s}_i$

ASM: Shape reconstruction

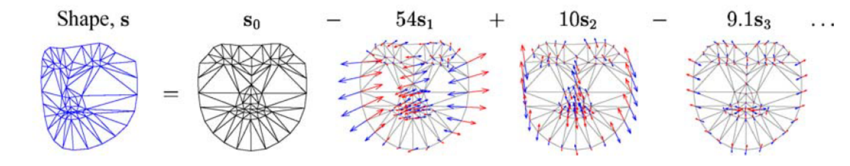


Figure: Reconstruction of an observed shape **s**

AAM: Learning the shape vectors

- From a functional data perspective, that's the good stuff.
- Let's quickly go over the Appearance model

AAM: Notations

- We consider appearance independently from shape.
- We learn appearances in within the base mesh $\mathbf{s}_0$
- The appearance of an AAM is $A(x)$ defined over pixels $x \in \mathbf{s}_0$

AAM: Representation

- Standard assumption: The appearance $A(x)$ can be expressed as a base appearance $A_0(x)$ plus a linear combination of m appearance images $A_i(x)$:

$$A(x) = A_0(x) + \sum_{i=1}^{m} \lambda_i A_i(x) \ \forall x \in \mathbf{s}_0, \tag{3}$$

where the coefficients λ_i are the appearance parameters.

- Standard assumption: the images $A_i(x)$ are orthonormal.

AAM: Representation

- Given a data after Procrustes Analysis and shape analysis.
- The process of mapping the colors on a shape $\mathbf{s}$ to the shape $\mathbf{s}_0$ is called wrapping (backward).
- We wrap the image appearances onto the base mesh $\mathbf{s}_0$ and

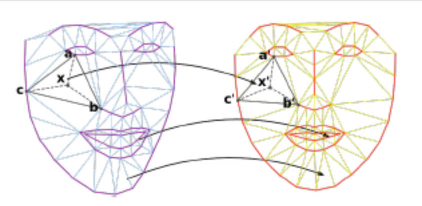


Figure: Wrapping appearance from $\mathbf{s}$ to $\mathbf{s}_0$

AAM: Representation

- We wrap the images onto the base mesh $\mathbf{s}_0$ and
- now we have a data set of appearance all on the base mesh.
- The images $A_i(x)$ are computed by applying PCA.

AAM: base image and appearances

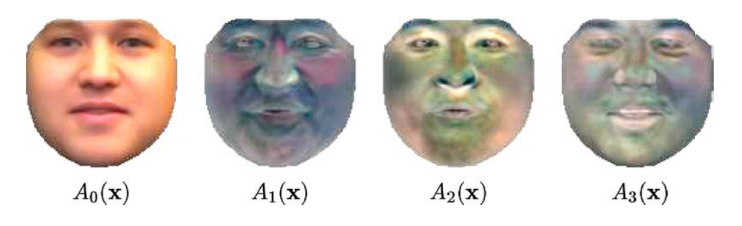


Figure: Base image $A_0(x)$ and appearances $A_i(x)$

AAM: Appearance reconstruction

Figure: Reconstruction of an observed shape **s**

Wrapping

- The process of mapping the colors on a shape $\mathbf{s}$ to the shape $\mathbf{s}_0$ is called backward wrapping.
- The process of mapping the colors on a base shape $\mathbf{s}_0$ to any shape $\mathbf{s}$ is called forward wrapping.
- Given a triangle formed by 3 identifiable vertices, any point in the triangle can be identified using the distance to the 3 vertices.
- Usually via a linear combination of the vertices.

Wrapping

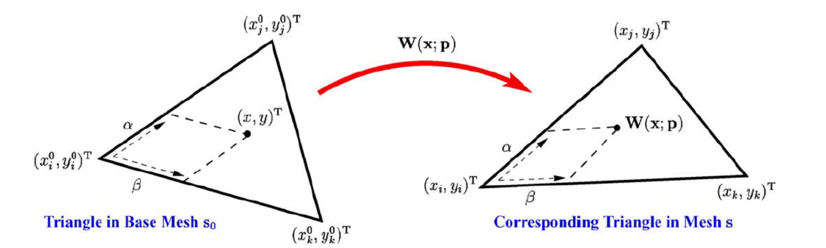
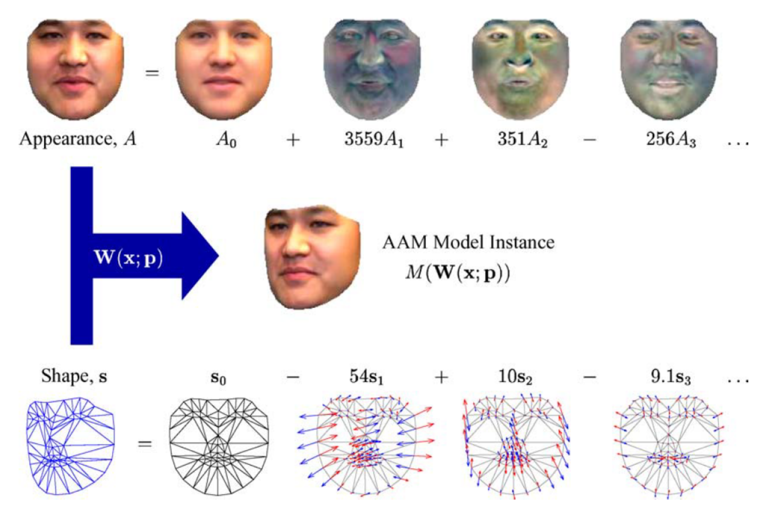


Figure: Wrapping

Reconstruction of a colored shape.



Why ?

- I suppose from a generative perspective that is enough ?
- We can adjust p_i and λ_i to create new faces.
- Though face generation is not the main purpose of these models.
- It is usually facial recognition or landmark identification (shape registration)

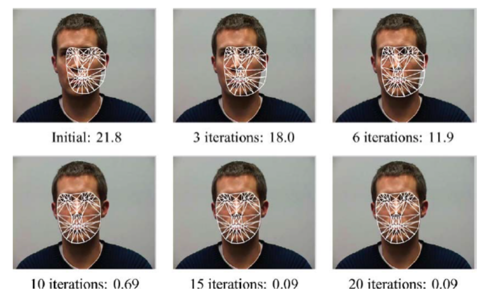
Fitting an AAM

- To fit an AAM is to find the optimal shape parameters p_is and appearance parameters λ_is for a new image.
- Given an image I we can backward wrap it onto $\mathbf{s}_0$: $I(W(x,p))$ (function of the shape parameters)
- We can try to reconstruct the look of this image with our appearance model $A_0(x) + \sum_{i=1}^{m} \lambda_i A_i(x)$
- We want to identify p_is and λ_is that minimizes:

$$\sum_{x \in \mathbf{s}_0} \left[A_0(x) + \sum_{i=1}^{m} \lambda_i A_i(x) - I(W(x,p)) \right] \tag{4}$$

Fitting an AAM

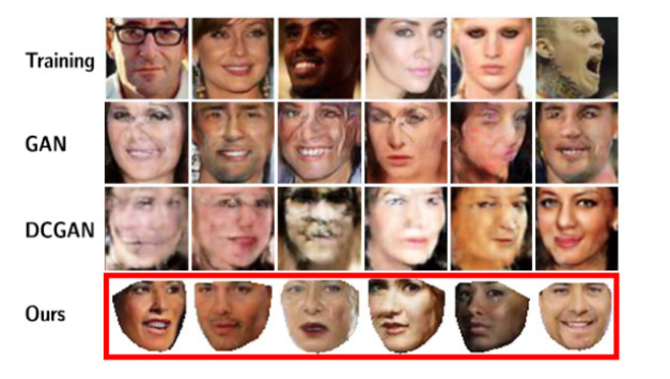
- This is doable but difficult.
- Tons of gradient-based model with multiple tricks.
- We can talk about it later!



Geometry Aware GAN (GAGANs)

Geometry Aware Generative Adversarial Network

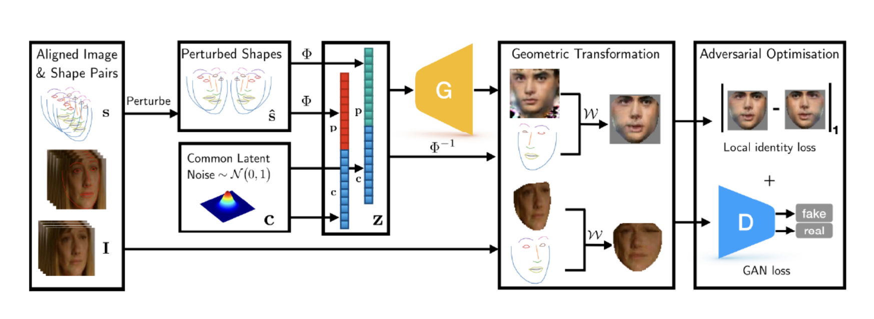
- First attempt that we know of to combine the deep learning adversarial aspect of GANs with the shape analysis aspect of AAMs.



GAGAN: Concept

- Uses ASM for the geometry of the image, but uses GAN concepts in place of the appearance model.
- Multiple images (different random shapes) of the same appearance are created.
- Fake and real images are fed to the discriminant D after wrapping on the base shape $\mathbf{s}_0$.

GAGAN: Concept



GAGAN: Shape model

- Once again, suppose we have v vertices, the shape is represented by a vector of size $2v$: $\mathbf{s} = (x_1, y_1, x_2, y_2, ..., x_v, y_v)$
- Translation, rotation and scaling are removed using Procrustes Analysis.
- Then, we extract the mean shape $\mathbf{s}_0$ and PCA is applied.
- We keep the $k-4$ shape vectors $\mathbf{s}_i$ associated with the top $k-4$ eigenvalues $\lambda_i, ...\lambda_{k-4}$ (paper says keep till λ_k).

GAGAN: Shape model

- To allow for the generator to also affect scale, translation and rotation,
- they build 4 additional components, for a total of k shape parameters.
- $\mathbf{s} = \mathbf{s}_0 + \mathbf{S}p = \mathbf{s}_0 + \sum_{i=1}^{k} p_i \mathbf{s}_i$

GAGAN: Shape model

- Claim: consider p_i's to be independent Gaussian variable with mean zero and variance λ_i.
- True for the $k-4$ first one but what about translation, rotation and scaling ? (paper uses $\lambda_{k-3}...\lambda_k$ from PCA).

GAGAN: Shape model

- Claim: Normalizing the parameters $\frac{p_1}{\sqrt{\lambda_1}}, \ldots, \frac{p_k}{\sqrt{\lambda_k}}$ enforce independence.
- Gives a criteria of how realistic is the shape:
- $\sum_{i=1}^{k} \frac{p_i}{\sqrt{\lambda_i}} \sim \chi^2$
- They cite a book on that one, but no pages nor sections.

GAGAN: Formal Definitions

- Given n images $I \in \mathbb{R}^{n \times h \times w}$, where h is height and w width.
- and their shapes $\mathbf{s} \in \mathbb{N}^{n \times v \times 2}$ (the vertices).
- For each shape $\mathbf{s}^j$ (observation), they generate L perturbed shapes $\hat{\mathbf{s}}^j = (\hat{\mathbf{s}_1}^j, ..., \hat{\mathbf{s}_L}^j)$.
- Denote $\hat{p}^j = (\hat{p_1}^j, ..., \hat{p_L}^j)$ the shape parameters associated with the perturbed shapes.

GAGAN: Generator G

- Given a noise vector c_i^j (for $i = 1, .., L$) (noise for GAN appearance)
- denote $\hat{z}^j$ the variable concatenating the shape parameters and the noise: $\hat{z}_i^j = (\hat{p}_i^j, c_i^j)$ (in paper $\hat{z}^j = (\hat{p}^j, c^j)$ but $\hat{p}^j$ is supposed to be a vector).
- This $\hat{z}$ is fed to the NN generator G who then produce an image.
- Thus this step handles the appearance.

GAGAN: Adversarial training

- For the adversarial training, we wrap fake and real images to their base shape. (As in AAM)

$$\min_G \max_D \mathbf{E}_{I,\mathbf{s}\sim p_{\text{data}}}[\log D(W(I,\mathbf{s}))] + \mathbf{E}_{z\sim N(0,1)}[\log (1 - D(W(G(z),\hat{s})))]$$

- Still a bit unclear if we sample $\hat{p}$ as well from a Normal(0,1).

GAGAN: Appearance preservation

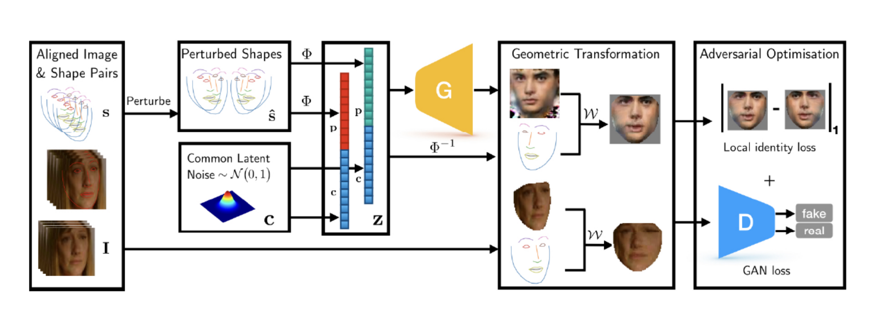
- Finally, differences in head pose should ideally not affect appearance.
- All shapes $\mathbf{s}^j$ and shape parameters p^j are mirrored $(\mathbf{s}^j_M, p^j_M)$.
- Given $m()$ a function that *flips* images.
- We minimize (w.r.t. G) the distance (after wrapping on the base shape $\mathbf{s}_0$):

$$LAP = |W(G(z), s) - W(m(G(z_M)), m(s_M))|$$

GAGAN: Objective function

$$
\begin{aligned}
\min_G \max_D \; & \mathbf{E}_{I,\mathbf{s}\sim p_{\text{data}}}[\log D(W(I,\mathbf{s})] \\
& + \mathbf{E}_{z\sim N(0,1)}[\log(1 - D(W(G(z),\hat{s})))] \\
& + \alpha \cdot LAP
\end{aligned}
$$

GAGAN: Model


Aligned Image
& Shape Pairs
s
I
Perturbe
Perturbed Shapes
ŝ
Φ
Φ
Common Latent
Noise ~ 𝒩(0, 1)
c
p
c
p
c
z
G
Φ⁻¹
Geometric Transformation
𝒲
𝒲
Adversarial Optimisation
Local identity loss
+
D
fake
real
GAN loss

GAGAN: Results

GAGAN: Control

GAGAN: Comparative Results

(a) Varying the shape parameters

(b) Varying the appearance parameters

Future Work

AAM: Possible improvement

- Can we analyse the shapes **s** using functional tools ?
- Consider 2d vertices and using FPCA ?
- Can we also improve on the appearance model ?
- Should we/can we include pieces of the appearance model even with adversarial training ?

GAGAN

- Need clarification regarding training.
- Can we improve LAP: We should make sure that the different images (different poses) of the same appearance (same person) look alike when projected on $\mathbf{s}_0$.
- Can we control the shape parameters ?
- Get an interpretable set of shape parameters and then we can fix them to our hearts desire.
- Can we get sparse shape parameters, allowing us to control facial features individually.
- Can we better control appearance ?

I would love to answer your questions.

Goodfellow, Ian, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. "Generative adversarial nets." Advances in neural information processing systems 27 (2014).

Srivastava, Anuj, and Eric P. Klassen. "Functional and shape data analysis. Vol. 1." New York: Springer (2016).

Matthews, Iain, and Simon Baker. "Active appearance models revisited." International journal of computer vision 60, no. 2 (2004): 135-164.

Kossaifi, Jean, Linh Tran, Yannis Panagakis, and Maja Pantic. "Gagan: Geometry-aware generative adversarial networks." In Proceedings of the IEEE conference on computer vision and pattern recognition (2018): 878-887.

Davies, Rhodri, Carole Twining, and Chris Taylor. "Statistical models of shape: Optimisation and evaluation." Springer Science & Business Media (2008).