Report of Project Complete the Femur in Scalismo

M. Stojchevska (marija.stojchevska@stud.unibas.ch), A. Wassmer (andreas.wassmer@stud.unibas.ch)
Dep. of Computer Science at the University of Basel
13.04.2020

In a small project, we consider 3D meshes and statistically evaluate them for predictions using the Software Library Scalismo [2] developed by members of the Graphics and Vision Research Group at the University of Basel. In more detail, we will attempt to evaluate a given set of training data, to train a statistical shape model to use this information to approximate further samples with missing pieces and complete those missing parts. We perform this task stepwise, for alignment a sample shape set and a set of partial shapes, with arbitrary positioned and oriented meshes, the creation of a Gaussian Process that is trained with the sample shape set and finally projecting complete meshes by fitting the partials using Gaussian Regression.

Introduction

In many situations like everyday life, medicine and industry, we came across the problem of elements with missing parts. In everyday life, that could mean a damaged picture or a picture with a subject covered by an obstacle. While we humans are very good in completing the missing parts, computers still have quite problems with this task. While this can be quite restrictive in algorithms like face detection, it is even more problematic for automated processes of analytics. In medicine for instance, the problem of missing information is even worse, needing this information for diagnostics or augmentations like prosthesis or reconstructions. In our project, we will primarily focus on the reconstruction and completion of given 3D shape using a statistic prediction model based on a training set of analogue subjects. For our task we only consider surface shapes and not volumetric ones. Consequently, we will use the term shape and mesh, the digital gird representation of a surface, interchangeably. With the medical area providing a major inspiration, we will use Femur bones as shape subjects for evaluation of the algorithms.

To realise this challenging endeavour, we will make use of the Scalismo Library that was developed at our university from several members of the Computer Graphics and Vision Group of Thomas Vetter [1], to mention Marcel Lüthi and Ghazi Bouabene et al. It is a Scala fork of the C++ implemented Statisimo and contains a graphical user interface, libraries for 3D modelling, image recognition, statistical learning, i.e. machine learning, collections of statistical function kernels, analysing tools and many more pre-implementations. As training data set, as much as the set partial shapes to be evaluated, we use samples from Sicas Medical Image Repository (SMIR). The sample set and the partial shape set are predefined to deliver comparability of the different solutions from people trying to solve this problem more accurate in a kind of competition.

For an easier understanding of the subject matter, we will first present key references to the theoretical background, and then discuss our implementation in a little more detail, ending with a conclusion of the project and an outlook of additional questions that we still could not answer satisfactorily. All subchapters, we present in the same order of steps as the algorithm does to give a better impression of what happens. So now let us start with the first step of our shape completion and that is alignment.

Theory

Within this chapter of the theoretical background of our project, we will present the most relevant concepts and ideas without showing the entire mathematical proof. All the mentioned concepts and solution are generally known it this field of work and were already investigated in depths by a broad spectrum of literature [3] [4] [5]

Alignment 101

The very first step in our algorithm is alignment, because any comparison and characterisation based on absolute coordinates is only applicable on aligned elements. Knowing that by shape we basically understand the geometric information reduced by translation and rotation, we will have to find a way to determine the two and get rid of them. To do so, we directly consider the surfaces, also known as contour in 2D as a set of points. But being part of a continuous space, we have more candidates that we actually need and considering a reduced set of a finite number of points is in order. Besides simply regularly distributing a larger set of points across the entire contour, we could also consider extremal points like landmarks, that also hold the most distinctive information about shape and therefore comparability. Such landmarks are typically tip points at edges and dents into the shape, and resides at the same corresponding position on all shapes to be aligned and compared. The remaining problem is then the task of mapping these set of points as closely upon each other as possible, using shifting, rotating and scaling. This task is known as Procrustes Analysis.

Procrustes Analysis

Procrustes Analysis (PA) has its name given from Greek mythology of fitting victims into beds stretching their libs or cutting them off, and therefore describes the old problem of superimposing similar types of subjects for optimal fit. To do so, we first need some measure of alignment and the

simplest way is choosing but the overall difference between the shape surfaces. Having our characteristic points, the landmarks from our previous discussion, with corresponding ones in all shapes, we have a compact form of description. Yet, this also means, we have a reduced set of statistical sample points that reduces our quality of superimposing two images. which is in essence a minimisation problem. A balance has to be found. To perform the PA, we can essentially split it into two subtasks, translation and rotation. In translation, we typically superimpose the centroids, the average of all considered points in each shape. For the rotation, we have to choose a reference to rotate the remaining shapes to it. The rest is but finding the minimal angle that produces the optimal small error. A least squared distance problem is to be solved. For a mathematical description, consider Addendum A1.

This is evaluated iteratively, but due to multiple degrees of freedom, several less optimal solutions, local minima, may exist. To ensure optimal convergence of our minimalisation problem, we augment it with a random variation element on the point positions, that introduces noise and ensures only global convergence. To additionally get rid of the choice of a reference point involved with rotation, we can add an additional iteration each time considering the new mean shape, as the one to align to. Including this last aspect, this is known as Generalised Procrustes Analysis (GPA).

Yet in case we don't have any landmarks, other predefined corresponding points, or we simply need more points for future comparison, we will have to sample points for a mesh and find the corresponding points within the other point sets, i.e. meshes. Because we do not have any clear one-by-one correspondence, we will have to use methods like iterative Close-Point approximation.

Iterative Close Point Approximation

Within the Iterative Close-Point (ICP) approximation, we consider the task of finding corresponding points between two distinct point sets residing in the same space. For two nearly identical and aligned shapes, we can simply consider the closest point as correspondence. But with enough variation, shapes are not locally aligned anymore we end up with wrong assignments of correspondence partners. But, keeping in mind that we observe smooth shapes, we can assume smoothness on both sides. Consequently, the point-wise transitions one shape to the other should also be smooth. Using Nearest-Neighbour Interpolation (NNI) we can ensure a smooth transition and correct correspondence assignment. Now by iteratively evaluation of those fields and reassigning correspondence, the algorithm converges very fast on a meaningful point correspondence for smooth shapes or shapes not consisting of local repetitive patterns.

The ICP can be used as several points during the project. It can be used to align the mesh data at the beginning combining the iteration with the one from PA. Although this will take significantly longer and use up more calculation power, we can sample an arbitrary number of comparison points and therefore are not bound to predefined landmarks anymore. Another important application of IPC lies finding correspondence points for additional mesh points during a later phase, when we project our trained model onto the partial shapes to create complete pendants of these meshes.

Having now aligned our set of shapes and created our set of corresponding comparable datapoints, we now want to find the characteristics of our shape set to model further potential candidates. For this, we will discuss now the creation of such a model in the next chapter

The Model Creation

In this chapter, we will discuss the method of generally describe the set of shapes based on characteristics and a generalised base. By considering application-based examples like samples of human hand, we observe that same types of subject seldomly show large variations in relative surface point positions, that allows us to describe every surface point by a probability region. Each point possesses a most likely position and a variance space around it with decreasing likelihood. A Gaussian distribution for each point can be found. With the entire shape being a linear composition of points, it is Gaussian distributed as well, described by a Multivariate distribution that is composed of most likely mean position of all points, and a variation described by the dependency on all other points. For increase comprehension, you find a general Multivariate Distribution in Addendum A2.

Yet observing explicit sample sets, we observe mainly local dependencies, represented by a sparse dependency matrix. Therefore, we get an approximate description with combination of local variance kernels across the diagonal, with each kernel described by a scaling and a variance value. Yet with a combination of such kernels describing a complete space, we can also describe more complex and wider reaching dependencies by expanding from this base, i.e. linear combine additional dependencies by their respective kernels. In fact, the multivariate distribution can be described as a linear (vector) combination of Gaussian kernels.

Now having found the mathematical description for the shape variations, we can conclude, that each shape can be represented by a mean shape and variation of the surface points within this own subspace of variations. Thus, each shape of the same characteristic can be described by the mean shape and a deformation, described by a so-called deformation field. This delivers us an interesting approach we will follow up in the next chapter.

Deformation Fields

The deformation fields are in general a description of a variation from a base state into the state to be described. Consequently, we can choose a reference shape like a mean shape and describe any shape of our set of translations of each point. Now with the different deformation fields considered as variations of each other, we remember the previous discussion and recognise this a sample states of a multivariate distribution. With the choice of a single mean mesh for all deformation fields, we get for each shape point multiple samples of its normal distribution and can therefore via a least square problem deduce the variation parameters.

This method of finding a multivariate distribution description is called regression and is used as a typical implementation of Principal Component Analysis (PCA).

Now we combine the facts that the shapes can be described via normally distributed variations of a deformation field, and the fact that multivariate distributions can be described by linear combinations of Gaussian Kernels. Consequently, we can describe all shape samples by a reference shape and a linear combination of gaussian kernels. With the kernel weights as parameters in a probability distribution, they represent random variables. This leads to a description of all shapes as sets of random variables and a reference shape. With a set of random variables describing a joint distribution and a transition from an initial state, we have a Gaussian Process like description of our shape set

Gaussian Process

As we described before, we have arrived at a point, that our set of shapes can now be described by a Gaussian like process. A Gaussian Process (GP) itself is a stochastic process that describes a joint distribution of normally distributed random variables. It therefore allows to characterise a set of variable tuples and predict additional candidates for that set by sampling a choice for each random variable. Consequently, a GP produces valid samples for all combinations of valid random variable choices, i.e. the entire definition space. Hence, we need do interpolate our data to get a description as a proper Gaussian Process. Because we assume shapes with smooth transitions in all aspects, i.e. properties described by Gaussian base functions, a common Nearest Neighbour Interpolation (NNI) is applicable.

Now with an effective GP description of our shape set, we can consider additional properties that follow. With every combination of random variables deliver a shape and each random variable describing different characteristics of that shape, there is a bijective map between shapes and random variables. In combination with a GP describing continuous transitions and Gaussian base function are compact over real space $\mathbb{R}$, GP can be found to describe any kind of shape and shape transition with a even exact representation. Yet with our shapes representing approximations, scans with measuring errors and such alike, and further contain additional uncertainty due to approximations in alignment and digitalisation, a discrete description with a limited number of base functions suffices. A common analogon to this would be using a Taylor expansion to find a polynomial representation of an arbitrary function and choosing a meaningful upper boundary of a polynomial degree fitting the target. As it is for Taylor expansions, we can use a flexible enough GP to find a suitable description for any shape too. For GP this is known as inference. Furthermore, a GP is in essence a statistic distribution, what means that evaluating an applicable distribution to a given set of weights, i.e. sample, can also be understood as a posteriori evaluation of the probability distribution. Mathematically, these conditioning on a target sample means also finding a stricter, i.e. closer, form of the underlying variation matrix, i.e. covariance matrix. This results in a so-called Closed Form Mean and Covariance. In conclusion, these GP properties delivers us two additional points of view and methods to generate, i.e. train, the model description of the given shape set.

With the acquired understanding of the GP, we now can find a suitable GP shape description using two additional approaches. Using posteriori conditioning, we can choose a flexible enough basic prior distribution and prime it by iteratively fitting, i.e. convolutional filtering, the given sample shapes onto it, with each sample narrowing and specifying the flexibility to the characteristics of the shape set. Hereby additional information over observed symmetries, smoothness degree and dependency range can be used to optimise the prior and, in the end, optimise the model. This is called Gaussian Regression or conditioning. Another approach would be using Principal Component Analysis (PCA) that uses convolutional base function description as a eigenvalue decomposition to find characteristics. And because all of these and previous methods deliver equivalent descriptions, also combinations of these approaches can be used.

Here, we would like to mention too that inferring on samples with own uncertainty, as we can resume for the given sample sets, can also deliver less optimal results due to conflicting variation properties. In other words, a measurement error can conflict with a previously trained characteristic for instance. The result can be, that the priming iteration will get suck in a local minimum. Similar to our solution in alignment to prevent getting stuck in local minima, adding an additional noise variation to the shape to be regressed ensures only global convergence and optimal fitting.

Now with multiple valid ways to construct our generator of characterised shapes given by the sample set, we can now use an additional advantage of having a Gaussian Process. One of these advantages lies in Gaussian inference and its flexibility on the number of observables. Although the sensor model, i.e. the mapping of the GP state to an observable sample state has a fixed rank, this is only enforced as an upper limit in inference. Consequently, GP allows straight out of the box for partial shape evaluation and even learning. Later on, we will use this fact to infer on partial meshes, i.e. condition the model onto the partial, and find all suitable complete shapes, the most likely one respectively.

Another property of GP lies in description of the effective state using an initial prior one. Using a chosen reference mesh as a prior, we now can directly sample shapes instead of changes, i.e. transitions. This description brings us to the next chapter, the Statistical Shape Model.

The Statistical Mesh Model

As we have seen in the previous chapters, we can describe the set of sample states via a GP and using a reference mesh as prior initial state. The resulting Statistical Mesh Model (SMM) can be seen as a direct generator of shapes having a specific characteristic described by a GP. The GP characteristic on the other hand is defined by the initial data set used to create and train the GP. With the SMM being base on a Gaussian, we also get a description of the likelihood space of any predicted, i.e. sampled shape.

For our task of finding completed shapes base on partial ones, we can use the partial inference property of the GP, as described in the previous chapter, and generate a posterior model describing all suitable shapes fitting the given partial. Using now the provided information of the corresponding likelihood space by the GP, we can deduce the most likely shape quite quickly. With the GP being based on normal distribution, symmetric around a mean with decreasing likelihood, the mean of the posteriori model directly describes our target.

Now having created a model generating the set of suitable candidates fitting our partial shape, we now have to deal with identifying the missing information and combining it with the original partial shape, i.e. mesh.

The Missing Pieces – a not so Trivial Task

Within the previous chapters, we discussed the task of shape completion up to the point of isolating the missing information and recombining of the shape, which we are going to do here now. The task of identifying the missing information seams rather simple, but we have to understand that our sampled complete shape is not a perfect match. Therefore, the partial and complete shape each possesses an own pointset different from one another. Consequently, we cannot check for the existence of mesh points in the partial mesh, and have to consider the distance to the corresponding points. Because the mesh alignment does not involve reordering the mesh construction, we cannot assume identical point ordering or point ID naming and have to resort to closest point approximations. Using this approximation, we will assign to each mesh point of the complete estimated shape a mesh point of the original partial shape. Using now the distance between each point and its assigned counterpart as a metric, we can choose a threshold upon which mesh-points are considered part of the missing information of partial meshes. To deduce a threshold, we cannot hardcode it allowing for arbitrarily selected edge-sizes, with the edges representing the space in-between the vertices, i.e. mesh points. Because these edge-sizes can even vary within a shape, a combination of relative differences of distances and sizes have to be considered to evaluate potential candidates for missing information. Here, we would also like to mention, that an iterative version of the closest point approximation, i.e. ICP approximation, is not necessary. The complete mesh is aligned an fitted onto the given partial, and consequently corresponding points should be found within a single iteration. Of course, generally it is possible to apply ICP at this point.

Generally, for the task we consider here, both closed and open meshes are possible. Consequently, even complete meshes can possess borders, as well as loop around and have none. This delivers different cases to be handled. In a first step, we consider the closest matching point of the original partial mesh and based on its number of connected neighbours we deduce whether or not it is indeed a border point. With this first criterion fulfilled, we have to check the mesh point in the completed shape for its neighbours. If it is not a border point, we can deduce that this is in fact a mesh point describing missing information. But if it is a border point itself, we have to evaluate further. A distance to the corresponding point that is a lot bigger than to its own mesh-neighbours, displays also a clear indicator for being a mesh point of the gaps. For the remaining points, we could not find a general working rule to assign points and implemented it with hard coded parameters for relative distances. For a discussion of their selection, please read on in the Implementation part of Extracting and Gluing.

Now having identified the mesh parts of missing information in the original partial shape models, we can now isolate the shape parts by reducing the complete mesh or marginalise the generating model onto this missing shape parts. The latter one would also allow us to sample afterwards new potential candidates beyond the most likely one. Now having all the theoretical parts discussed, we will continue with the discussion about the explicit implementation of the task.

Implementation

The Alignment

For the alignment part, we used the predefined method of Procrustes Analysis and rigid alignment from the Scalismo library [2]. With the full Procrustes implementation, we could directly link an entire folder of meshes and produce somehow aligned set with a mean, yet have no control over the process. We mainly used the rigid alignment on manually loaded meshes and landmarks implementing our own iterative process of centroid and landmark based alignment. In this iterative process, we also calculate a mean shape with mesh and landmark set. For already well aligned and close resembling sets, we also have a non-iterative alignment that uses the first mesh as mean reference to allow faster computation. The mentioned centroid alignment is an additional speedup process that helps with faster convergence in the alignment process. To manipulate the meshes and landmarks, we used the Scalismo internal transformations from relative translations or returned from rigid alignments. In Addendum B1, you find a plot of all samples shapes prealigned, centroid aligned and Procrustes Aligned to get an idea of the generality of the given sample space.

The Model Creation

For the implementation of the statistical model, we made use of the several Scalismo internal definitions implementing the different methods to generate and train a statistical model. We started with hard defined flexible Gaussian distribution based on sparse composition of default kernels to be posteriori trained. After variating the kernel parameters and compositions, we also investigated GP generation using PCA. With shapes allowing for a wide range of variation, we also investigated the impact of ICP and the number of iterations. At the end, we were using a combination of the lot to ensure big enough flexibility, smooth transitions and capturing of the shape characteristics. The most successful construction uses a scaling and changepoint Gaussian Kernel composition with a Nystrom 2000 point low rank approximation and limitation to 100 base functions as a high flexible base to use ICP. For scaling we have scales between 20 and 60, and with variances we used 30 to 45. With changepoint kernels we insured different scaling in different subdomains. With 10 iterations, we found correspondences for all 5000 sample points per shape now

using the Gaussian regressed posteriori mean as representative for each mesh. Due to the high flexibility yet correspondence to the reference mesh, we were able to ensure much smoother surfaces than using the original sample meshes. These new found representatives and correspondences are now feed into a PCA to generate our effective model of the training data. Experiments testing the inclusion of the provided (distinct) landmarks as additional correspondence points showed only little effect. Alternatively, using PCA with correspondences from the original sample meshes also showed adequate results using posteriori smoothing, yet greatly increased computation requirements in comparison.

In addendum B2 you find a collection of the partial Meshes and the corresponding complete meshes that resulted from the implemented calculation. As it is clearly visible, our final choice of kernel did not produce optimal projections for all partial meshes. Although not getting nice results for shape 3 and 9 with these parameters, an earlier combination did in fact deliver nice results for these shapes and worse for others. Finding a single configuration for all meshes turned out to be a very tricky task.

Extracting and Gluing

Now in this last part, we implemented the automated handling of shape completion having a partial and a completed mesh available. As we have described earlier, the difficulty lies in finding the correct corresponding partners in the other mesh, and identifying the relative position of the mesh points. Using the pre-implemented algorithm from Scalismo of find the closest point within a given mesh to a given position, the difficulty lies in finding appropriate thresholds and rules to assign points to be part of the one-sided missing information or not. Our choice of solution then fell to a rule-based filtering, in which every rule captures a specific part of missing information.

As we described within the theory section, the distinction between points inside a surface and at a border has to be established. Because we have use triangular meshes, each point has 6 neighbours, except it is a at a border, there the number of neighbours can vary from 5, an inwards pointing corner, to 2, an outward pointing tip. Keeping in mind our target of combining our estimation of the missing information with the original mesh, a slight overlap with point-duplication at the boundary is intended. Thus, no further implementation is necessary to construct the first criterion.

For the further rules that established assignment for correspondence between border points in both meshes, a complete generic implementation is not possible any more. By comparing the distances of corresponding points to grid space dimensions, the thresholds have to be pre-decided. While theoretically an additional learning process with a neural network could be trained to find threshold candidates, we did not had access to a set of related training data and used a hard-coded estimate. Choosing that the distance has to be bigger than the biggest local mesh space as threshold, we found a working solution for our task here.

To smoothen further the transition of a new border from our estimation of the missing piece joining up with the border of the partial mesh, we added an additional criterion. We further assume, that all border points of the complete shape that have a neighbour, which is a surface inside point and corresponds to a boundary point of the partial mesh, are also part of the estimated piece of missing mesh points.

This found piece can now be combined with the partial by merging overlapping mesh points using default pre-implemented subroutine for mesh generation. This last step remains not yet implemented though. Although the implementation works for our set of samples, broader testing would be necessary to properly validate our choice of thresholds. For the most part of our testing, we skipped this extraction step all to gather for better validation of the matching algorithm and higher performance.

Conclusions

The Scalismo library [2] delivers a well working base that helped us achieving a modest working solution to estimating potential completions for partial mesh shapes. Yet there are still limitations in our solutions

For once, we still are using a fixed number of samples in posteriori fitting and a fixed number of iterations for ICP. Potentially finding a suitable general metric to estimate the iteration quality would allow for a more dynamic approach and increase quality and reduce running speed.

Although we found a good balance between precision and smoothness, being bound to a fixed chosen kernel as first high flexible approximation can introduce artifacts and has to be manually evaluated for other shape types much further. Here, a more general dynamic solution perhaps involving Neural Networks would be interesting.

Further, we are not entirely sure about the applicability of our algorithm finding the missing mesh pieces in unclear situations for all potential types of subjects. Here further testing would be necessary.

Outlook

Due to the hard to predict character of the involved numerical elements, there is still a lot potential in finding alternative well working configurations. A general solution, although highly desirable is still beyond the grasp of this and many other related projects for the time being. For now, optimising ICP usage or finding an alternative, optimising PCA and increasing performance to allow higher ranking GPs are all meaningful reachable tasks. Furthermore, algorithms to automatically find distinct landmarks and metric evaluation of the iteration steps deliver also prominent candidates for follow up projects.

References

1. Prof. Dr. T. Vetter, Dr. M. Lüthi, , Dr. A. Forster, D. Rahbani, D. Madsen, et al. , https://gravis.dmi.unibas.ch , University of Basel, Department of Computer Science

2. G. Bouabene, T. Gerig, M. Luethi, A. Forster, D. Madsen, D. Rahbani, P. Kahr, https://scalismo.org , Project of Graphics and Vision Research Group at the University of Basel in Collaboration with Shapemeans GmbH

3. https://www.wikipedia.org

4. With Scalismo associated MooG https://www.futurelearn.com/courses/statistical-shape-modelling

5. G. Zheng, S. Li, G. Székely , "Statistical Shape and Deformation Analysis" , V1 *Elservice Ltd* DOi : 10.1016/C2015-0-06799-5 , ISBN : 978-0-12-810493-4 , https://www.sciencedirect.com/book/9780128104934/statistical-shape-and-deformation-analysis

Appendix

A1 : Math of Procrustes Analysis

Scaling Centreoid : with k the number of mesh points of a shape and $\vec{x}_i$ the position vector for each point :

$$\overline{x} := \frac{1}{k} \cdot \sum_{i=1}^{k} (\vec{x}_i)$$

Rotation : Choosing a mesh as a reference with its shape points $\{\vec{x}_{r,i}\}_{i=1}^{k}$, we have to find for each of the other shapes a rotational matrix $R(\phi, \theta)$, so that their coordinate points $\{\vec{x}_i\}_{i=1}^{k}$ fulfil :

$$\vec{x}_{r,i} = R(\phi + \Delta\phi_i, \theta + \Delta\theta_i) \cdot \vec{x}_i \ \forall i$$

with $R(\phi, \theta)$ a minimal rotational matrix :

- $\phi, \theta \in [0, 2\,\pi)$ minimal
- $\sum_i (\Delta\phi_i)$ and $\sum_i (\Delta\theta_i)$ minimal

Further, we know $R(\phi, \theta)^T \cdot R(\phi, \theta) = I_{3\times 3}$ the idendity matrix and $\det(R(\phi, \theta)) = 1$.

We use here an approximation, because no the shapes also vary and it is not possible to

A solution for this delivers the Singular Value Decomposition that lets you rewrite the matrix $\vec{x}_{r,i} \odot \vec{x}_i^T = M_{\text{leftside}} \cdot \Sigma \cdot M_{\text{rightside}}$ with $\odot$ the matrix multiplication and Σ an unary matrix. The Rotation matrix can then be evaluated as $R = M_{\text{leftside}} \cdot M_{\text{rightside}}$

A2 : Multivariate (Normal) Distribution

Let$\vec{x}_i$ describe the points in a shape for instance and $\vec{\mu}_i$ the mean position and $\sigma_{i,j}$ the covariance matrix between point i and j :

$$\begin{pmatrix} \vec{x}_1 \\ \vdots \\ \vec{x}_k \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \vec{x}_1 \\ \vdots \\ \vec{x}_k \end{pmatrix}, \begin{pmatrix} \sigma_{1,1} & \dots & \sigma_{1,k} \\ \vdots & \ddots & \vdots \\ \sigma_{k,1} & \dots & \sigma_{k,k} \end{pmatrix} \right)$$

with each covariance matrix itself a 3×3 matrix. As a covariance matrix, is is symmetric and positive definite

In general, x_i only represent an general random variable of arbitrary dimension. By training our data and using low rank approximations, we can describe our shape by lower dimensional matrices, in which each random variable describes one type of characteristic of the shape.

As an intersting side node, with all these characteristcs, i.e. dependencies described as distributions, we will end up with the Gaussian kernel representation, that was mentioned in our discussion. Each describing a base function that will map the characteristic back into a $\mathbb{R}^3$ centered around origin. Considering this a vector space, i.e. deformation of a reference

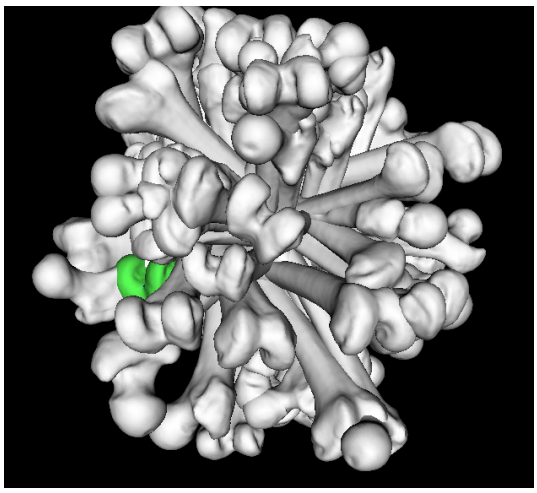
[!H] 

Figure 1 Unaligned

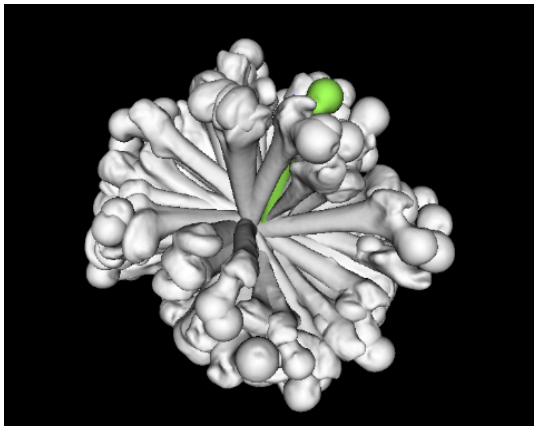
[!H] 

Figure 2 centroid aligned

object in $\mathbb{R}^3$, we end up again with a shape described by its surface points.

B1 : Alignment Pictures :

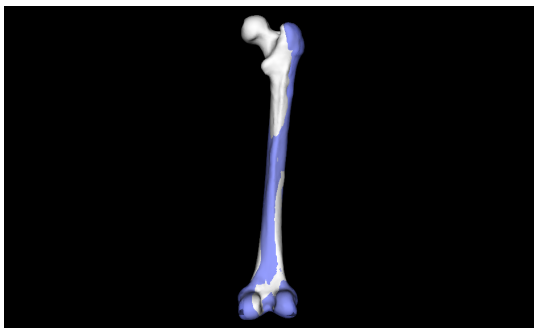
[!H] 

Figure 3 partial 1

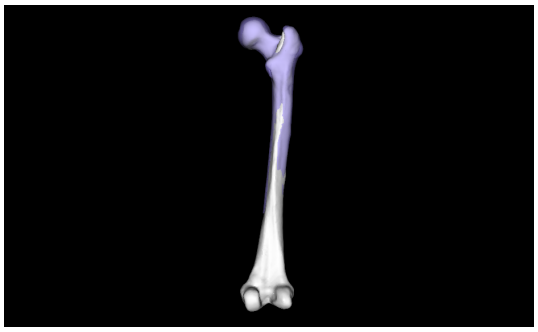
[!H] 

Figure 4 partial 2

B2 : Projected Partials :

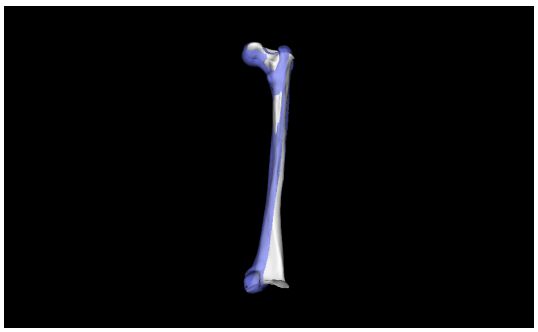
[!H] 

Figure 5 partial 3

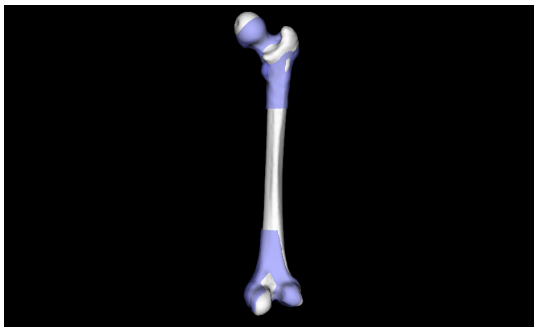
[!H] 

Figure 6 partial 4

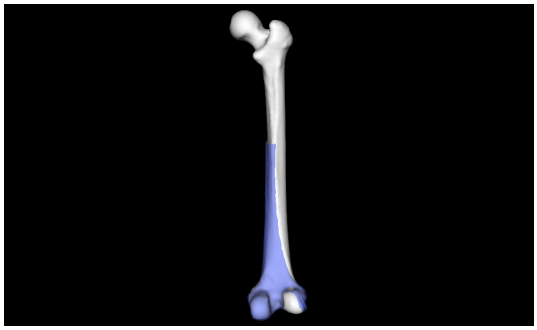
[!H] 

Figure 7 partial 5

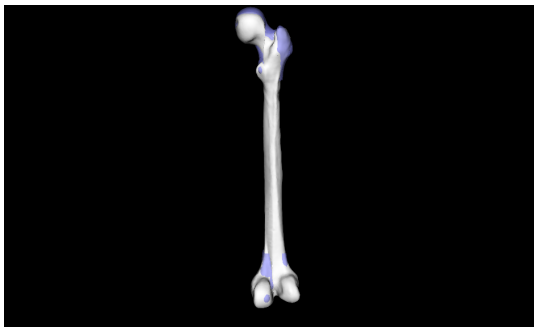
[!H] 

Figure 8 partial 6

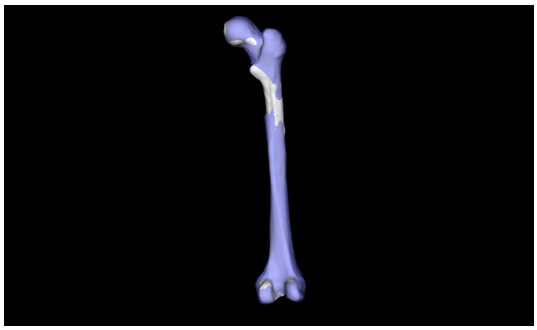
[!H] 

Figure 9 partial 7

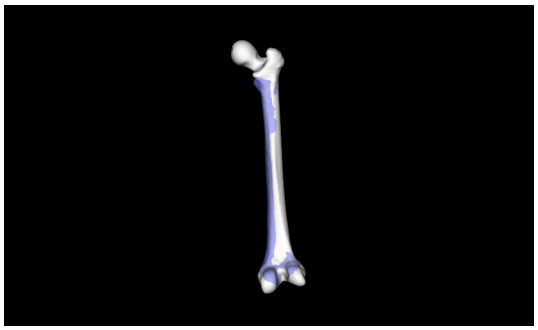
[!H] 

Figure 10 partial 8

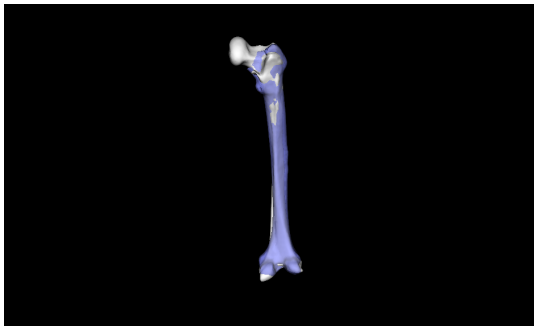
[!H] 

Figure 11 partial 9

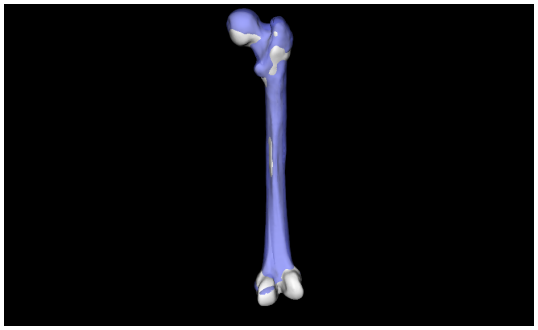
[!H] 

Figure 12 partial 10