Optimized Tissue Structure Modeling
Computer Integrated Surgery II
Spring 2015
Final Report
Team 10
Team: Benjamin Strober & Nate Schambach
Mentors: Prof. Kobilarov, Prof. Taylor, Preetham Chalasani

1 . Introduction

Our project is driven by the fact that there is currently no efficient way to create a digital re-modeling of both the geometry and stiffness of a surface. The practical benefit of having an efficient method to perform a re-modeling of a surface is for medical applications. It would be of great benefit to a surgeon if he or she could accurately and quickly learn both the geometry and stiffness of a tissue surface they are going to operate on. While there are current methods to develop a model of the geometry, there are no efficient and accurate methods to model the stiffness. Stiffness information on a tissue surface could provide the surgeon with knowledge on the location of biological structures of interest such as tumors or vasculature. It should be clear that the location of tumors or vasculature could not be determined from a geometric remodeling alone.

In order to address the goal of creating an efficient digital re-modeling of geometry and stiffness we started by looking at the more specific question of how to accurately reconstruct a surface without any assumptions on the underlying structure. It is essential that there are no underlying assumptions made on the structure because in a medical setting (especially if modeling a subcutaneous organ) there would be no prior information to go off of. Under the guidance of Professor Kobilarov, we choose to use force sensor mounted to a cartesian stage to measure surface height and stiffness at selected points. Furthering on this, we created a Gaussian Process (GP) algorithm to

independently model both geometry and stiffness that uses data from force sensor palpations as training points for the GP. It is noteworthy to mention that this is fundamentally an exercise in using a finite number of training points (from force sensor readings) to interpolate all other points within the tissue range. There are many mathematical methods that could be used to achieve this. However, a GP is advantageous because every predicted (interpolated) point has an associated confidence interval, which is something that a polynomial fitting could not offer. A confidence interval is clearly essential for a medical setting. To clarify how our system will work, a first point is initially palpated. Based off of height and stiffness recorded at this point, two independent GPs will be created (one for geometry and one for stiffness). The next point will then be palpated and both the GPs will be updated in light of the new training data. This iterative process will continue for all palpations.

Using GPs offers a solution to re-model both geometry and stiffness, however the questions of how to select the next point to palpate and how to have the fewest number of points to perform this reconstruction still remain. Obviously, the more points palpated, the more accurate the model will be. However, this comes at the cost of time which is critical in a medical setting. To address this problem, we developed several adaptive search algorithms to select the next point to palpate. In order to use the fewest number of points to create an accurate reconstruction, the point selected needs to be the point that provides the most information. I am being intentionally vague by saying we want to choose the point with the "most information". This is because a successful search algorithm depends on what you are searching for. Therefore, we implemented and tested various algorithms that identified their own specific features of the surface.

Section 1: Introduction

What I just described is our teams contribution to a project that is being handled by Preetham Chalasani, a Hopkins graduate student. Our team only wrote the gaussian process algorithm and the adaptive search algorithms. These algorithms were written in matlab. The interface between our algorithms and those of the cartesian stage, as well as physical control of the cartesian stage was implemented by Preetham. So with Preetham's help, we were able to test the efficacy of our algorithms using a force sensor mounted to a cartesian stage, which palpated a phantom. The phantom that we used for testing was provided by collaborators at Carnegie Mellon University.

2. <u>Gaussian Process</u>

Before we discuss our various search algorithms, we thought it would be beneficial to describe Gaussian Processes (GPs) for they are essentially the backbone of our project. Further, an understanding of GPs is beneficial to understanding the search algorithms.

A GP is entirely defined through its mean function (m(x)) and covariance function (k(x,x')) such that: $f(x) \sim GP(m(x), k(x, x'))$

The GP gives rise to an infinite dimensional function.[2] This is important because it allows the user to determine the value of f at any x. However, for the purpose of using a GP as a model, we will convert the GP to a multivariate gaussian distribution.Before we discuss specific details, we want to cover some basic notation. Training points are the (x,y) locations where palpations have already occurred. Training set values are the respective values of each (x,y) location. In the geometry GP, the training set values are heights that we have determined through palpation. In the stiffness GP, the training set values are stiffnesses that we have determined through palpation. Test set points are (x,y) locations that we wish to predict the value at. The values at the test set points are called test-set values. The multivariate distribution we will be using is defined by:

$$1) \quad \begin{bmatrix} y \\ y_* \end{bmatrix} \sim \mathcal{N} \begin{bmatrix} 0, & K & K_*^T \\ & K_* & K_{**} \end{bmatrix}$$

Where y are the values from training points. K is the training set covariance matrix. K* is the training-test set covariance matrix. K** is the training-test set covariance matrix. And y* are the test set values. y* is the only unknown in the above equation. We will soon discuss how to estimate y*. It is also interesting to note that the mean of the multivariate gaussian distribution is the zero vector. This is due to, as discussed in the introduction,

we wish to model the surface with no underlying assumptions made on the surface.[1,3] However, if the surface had a known average shape (like a ellipsoid), this could be added to the model through altering the mean of the multivariate gaussian.[4]

You may be wondering how the covariance matrices are computed. We used the squared exponential covariance matrix to compute the covariance between points. Each element is defined by:

$$\textbf{2)}\quad k(x, x') = \sigma_f^2 \exp\left[\frac{-(x - x')^2}{2l^2}\right] + \sigma_n^2 \delta(x, x')$$

where, for our purposes, x and x' are two arbitrary (x,y) positions. l is a hyperparameter that controls the effect that the distance between two points has on their covariance and thus controls the relative "smoothness" of the surface. This was estimated through visual inspection. σ_f^2 is another hyperparameter and is also the function noise. It scales the expected variance between points and can also be estimated through visual inspection. l and σ_f^2 can be optimized via maximizing the log-likelihood. However, for preliminary testing, visual inspection is sufficient. This is because, reasonable hyperparameters do drastically not affect next point selection criterium and the GP can be re-computed on existing data once hyperparameters are optimized. σ_n^2 is a hyperparameter that represents signal noise. $\delta(x, x') = 1$ if x = x', else $\delta(x, x') = 0$. Signal noise was estimated through variance present in the force sensor.

From equation 1, we can determine the distribution of predicted values at test points given the values from test points. This is exactly what we are trying to compute. It is:

$$\textbf{3)}\quad y_*|y \sim \mathcal{N}(K_* K^{-1} y, K_{**} - K_* K^{-1} K_*^T)$$

And thus we know the predicted values at the test points and the variance at those test points:

$$\textbf{4)} \quad \overline{y*} = K_* K^{-1} y \qquad \textbf{5)} \quad var(y_*) = K_{**} - K_* K^{-1} K_*^T$$

And thus we can predict the mean and variance of all points within set of test points.[1]

   Before implementing the search algorithms, we tested our GP algorithm by palpating the phantom in a uniform 18 by 18 grid that covered the entire phantom. 324 points is considerably more points to palpate than desired, but we choose to palpate so many because this was a preliminary test. This also provides us with a "near ground-truth" for stiffness. As the there exists a ground truth for the geometry of the phantom, but no ground truth for stiffness.

 As stated above, there does not exist a stiffness ground truth. However from palpating many points (as shown in the below reconstruction), we can infer the stiffness is highest along a thin wire (not shown in CAD model because it is beneath the surface). In the picture of the actual phantom shown on the left, it is the thin red line running through the middle. It can be visualized in the CAD model as the bright yellow line. All elevated surfaces show the next highest stiffness. And all non-elevated surfaces not along the wire show low stiffness.
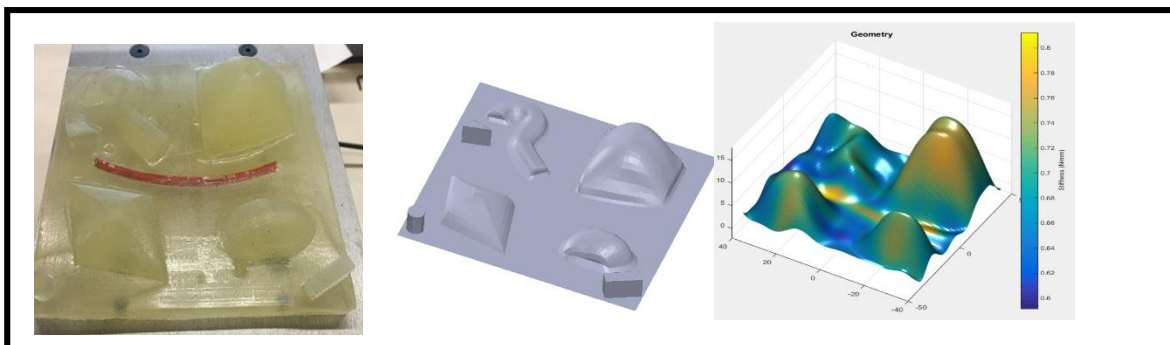
Figure 1: (from left to right). The first is an actual image of the phantom used for testing. The second picture is a CAD model of the phantom. The last image is a GP reconstruction (MSE = .75) after palpating a uniform 18x18 grid on the phantom

3. Search Algorithms/Results

3.1 Overview

When it comes to complex structures, there is no "one size fits all" search algorithm which uses the same number of sample points. One cannot both search for a maximum and hope to sample points which reconstruct the best overall model unless the structure sampled from is very simple. However, to address various needs as well as to draw comparisons we tested and implemented several different approaches. These approaches as well as their results are below.

3.1.2 Choosing the Next Point
One feature is consistent throughout all of our search algorithms. When computing the next point at which to palpate we first uniformly random sample N points within a box centered around the current point. One of these randomly selected points is then selected as the next palpation point. Random sampling is chosen over a grid selection of points due to the large amount of computation which would need to be done for the number of points necessary to cover the area with enough resolution.

3. 2 Maximum Search

3.2.1 Algorithm

This search takes the N points and predicts the expected mean and variance of each point using the GP prediction equation:

$$\max(a * predicted\ mean + b * predicted\ variance)$$

The next point chosen is then the point which satisfies the following condition:

7

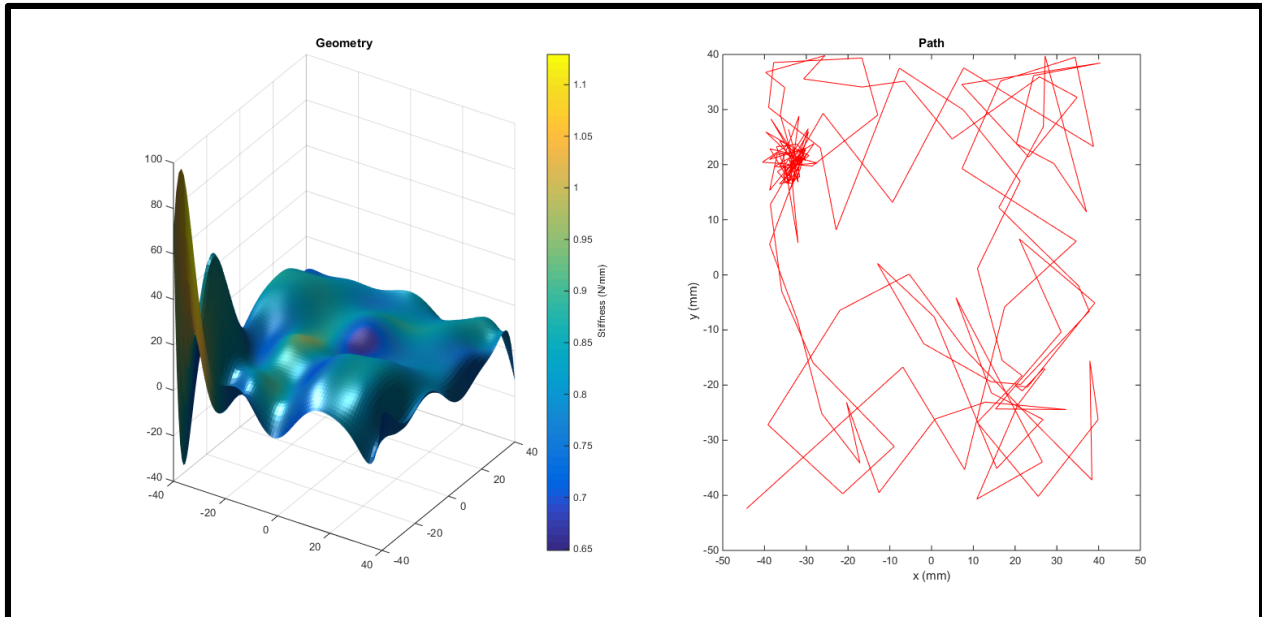Where *a* and *b* are weighting factors. While this is very efficient is has several key

problems:

1) It is difficult to combine criteria of different units such as height or stiffness

2) It leads to the discovery of local maxima, at which point the algorithm tends to

    stay indefinitely.

3.2.1 Results

Maximum Height Search
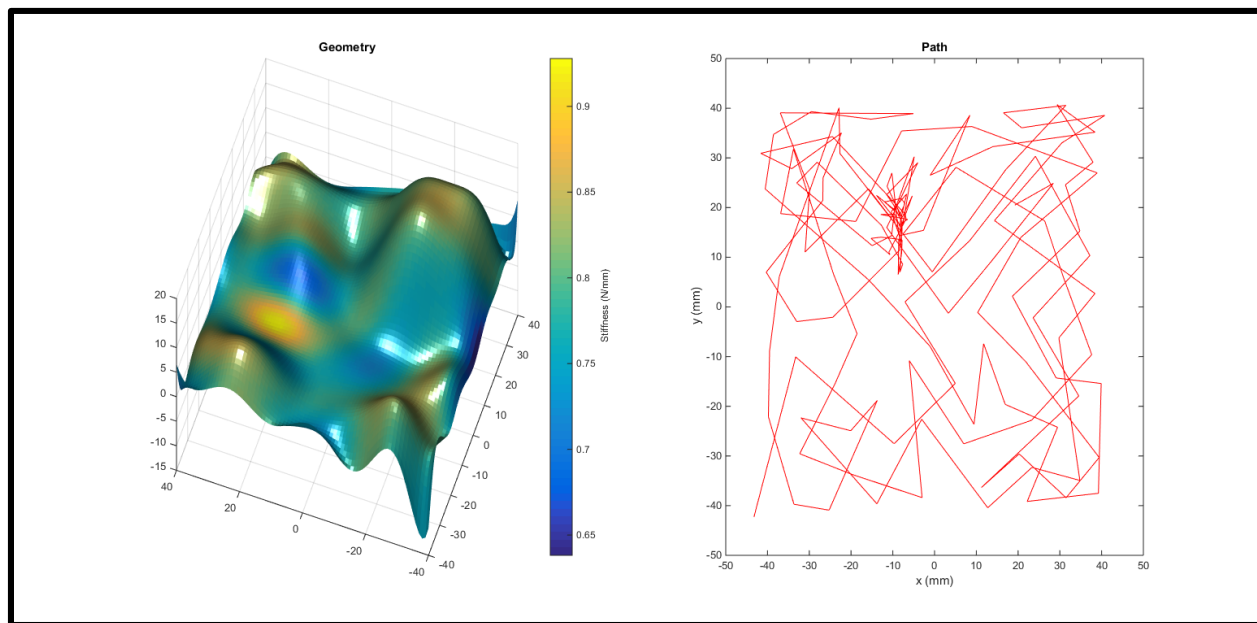

Figure 2: Results from maximum height search. On the left, an image of the reconstruction using GPs. This was done with L= 35 mm, variance weight = 20, and mean height weight = 1. The MSE was 5.046. On the right, the palpation trajectory

   With enough tuning of the box size to be sampled from and weighting of the

variance term you can get the algorithm to explore such as in this example, where the

true maxima is found and others are slightly explored. However it would still be very

easy for the algorithm to miss another maxima.

Maximum Stiffness Search

Figure 3: Results from maximum stiffness search. On the left, an image of the reconstruction using GPs. This was done with L= 35 mm, variance weight = 4.5, and mean stiffness weight = 1. The MSE was 1.521. On the right, the palpation trajectory

The results here are very interesting because the geometry reconstruction is actually quite good. The reason for this is the low stiffness of the humps and the high relatively small area of the stiff area(subcutaneous wire). The end results are almost a random grid with some favoring of the slightly stiffer humps. Eventually, when the wire is discovered the algorithm sticks to it. However because it is looking for a maximum peak it has difficulty exploring the rest of the wire. In cases where the model is similar to this one, where the interesting area is small this algorithm works relatively well.

As demonstrated, the algorithm is very efficient at finding the first maxima it encounters. A combatant for this could be a grid or randomly sampled initialization. The algorithm would initially select N points over the surface after which the Maximum search would

9

begin. For enough exploration, weighting of the variance vs mean values is highly dependent on the specific structure itself.

3.3 Dynamic Maximum Search

3.3.1 Algorithm

This basic adaptation of the Maximum Search algorithm takes the previous factors and scales them by the predicted values from the point most recently palpated. The criteria is as below:

$$\max\left(a * \frac{Predicted\ Mean}{Previous\ Predicted\ Mean} + b * \frac{Prediced\ Variance}{Previous\ Predicted\ Mean}\right)$$

The previous predicted values come from the previous iteration of the algorithm and are not recomputed at that point. They are not recomputed because of the gaussian processes nature. Freshly computed values would result in very low variances, skewing the variance term and enhancing the tendency for the algorithm to leave a newly found high point. This tendency is explained in greater detail in the behaviour below.

By scaling the means and variances by the previous predicted variances the algorithm maxes the values unitless, allowing for combination of the height and stiffness as a criterion as well as drastically reduces the likelihood of getting stuck at a local maxima.

The following table describes some of the algorithms key behavior:

Event Mean	Event Variance	Resulting Behaviour
High Predicted, Low Previous Predicted	Similar Predicted and Previous Predicted	Search approaches Maxima
Low Predicted, High Previous Predicted	Similar Predicted and Previous Predicted	Favors variance, tendency to leave maxima for more information
Similar Predicted and Previous Predicted	High Predicted, Low Previous Predicted	Favors variance, tendency to leave overly sampled area for more information
Similar Predicted and Previous Predicted	Low Predicted, High Previous Predicted	Favors means, tendency to approach maxima
Similar Predicted and Previous Predicted	Similar Predicted and Previous Predicted	Favors weighting factors, *a* and *b*, but essentially the same behaviour as a random search

This behavior is obviously very promising when the area has all been relatively well covered. But what happens when an area of interest has not been explored yet? When an area has not been explored yet, the search has the opportunity to initially miss the area of interest, hitting it once, and then likely jumping away because the mean factor is liable to be close to 1. However upon the next iteration, the mean factor will be lower and the variance term will dominate. In the ideal situation this will encourage the search algorithm to return in a similar direction.

A quadseek algorithm (see 3.4.3) could be argued to be implemented for improving overall results. However as the results by Seo et al. show, ALC is only better than selecting the points with high predicted variance when there are few numbers in the training set. For the advantage of simplicity and because areas which are of interest will

naturally have more points on them in this algorithm, the advantages of ALC were deemed not worth it.   Combination of criteria for stiffness and geometry is done as follows:

$$Geometry\ Term = \ a * \frac{Predicted\ Mean\ Height}{Previous\ Predicted\ Mean\ Height} + b * \frac{Predicted\ Variance\ in\ Height}{Previous\ Predicted\ Mean\ in\ Height}$$

$$Stiffness\ Term = \ c * \frac{Predicted\ Mean\ Stiffness}{Previous\ Predicted\ Mean\ Height} + d * \frac{Predicted\ Variance\ in\ Stiffness}{Previous\ Predicted\ Mean\ in\ Sitffness}$$

$$\max\big(G_{weight} * GeometryTerm + K_{weight} * StiffnessTerm\big)$$

3.3.2 Results
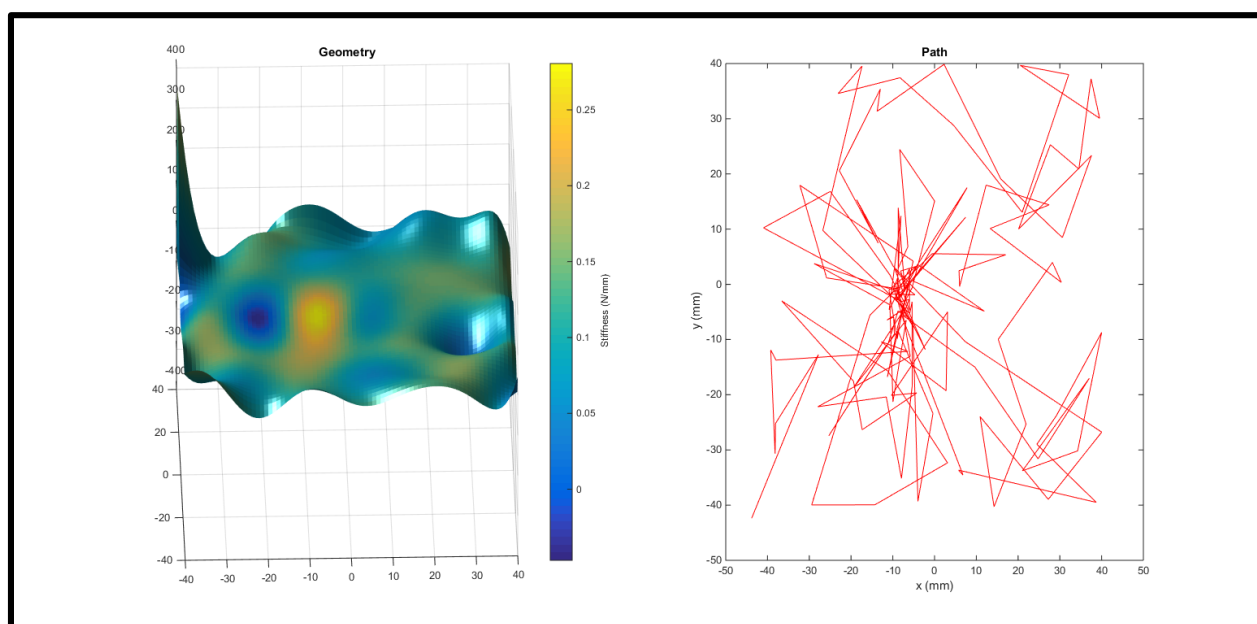**100% Weighting on Geometry, 0% Weighting on Stiffness**
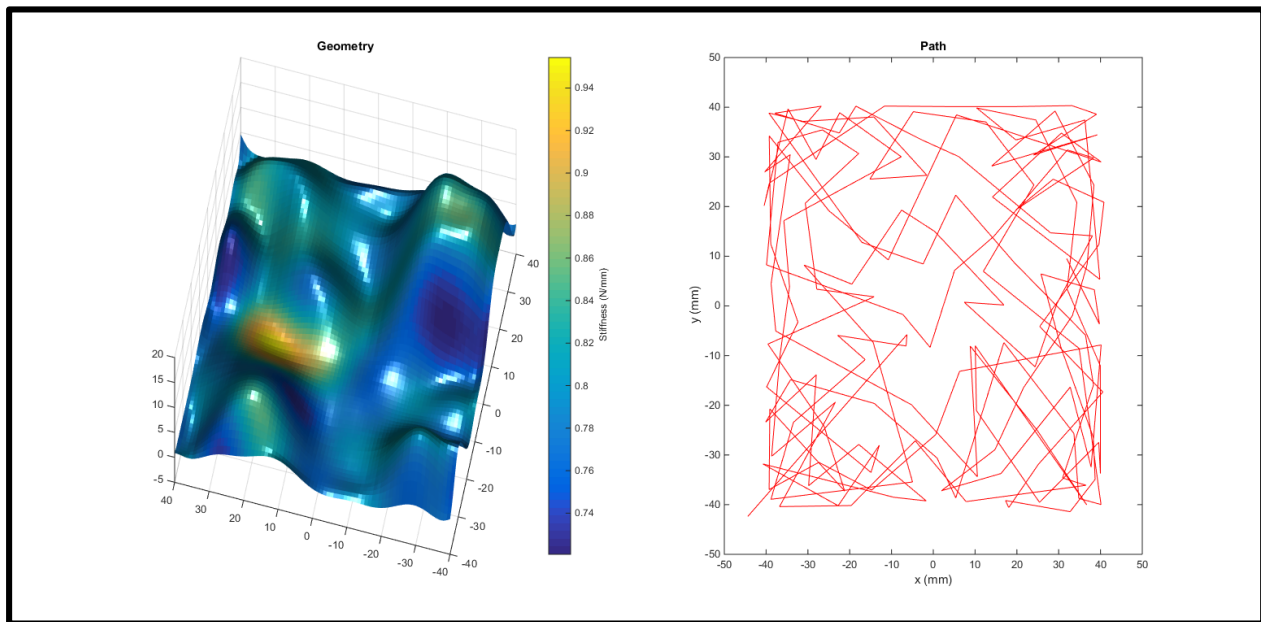
Figure 4: 100% weighting on geometry, 0 % on stiffnessResults from dynamic maximum geometry search. On the left, an image of the reconstruction using GPs. This was done with L= 35 mm, variance weight = 1, and mean geometry weight = 1. The MSE was 1.1719. On the right, the palpation trajectory
*Some of the geometry is off due to a shift in the phantom during measurement

We had our most successful geometry reconstruction using this algorithm with a MSE of 1.1719. Further, as can be observed, the path was constrained to all four corners of the grid. This is exactly what we wanted to occur because the corners are the only locations that have height above ground level. What is so great about this algorithm is that, unlike the prior searching for max height algorithm, the path did not get stuck on one hump. It continuously explored. We were very pleased with this result.
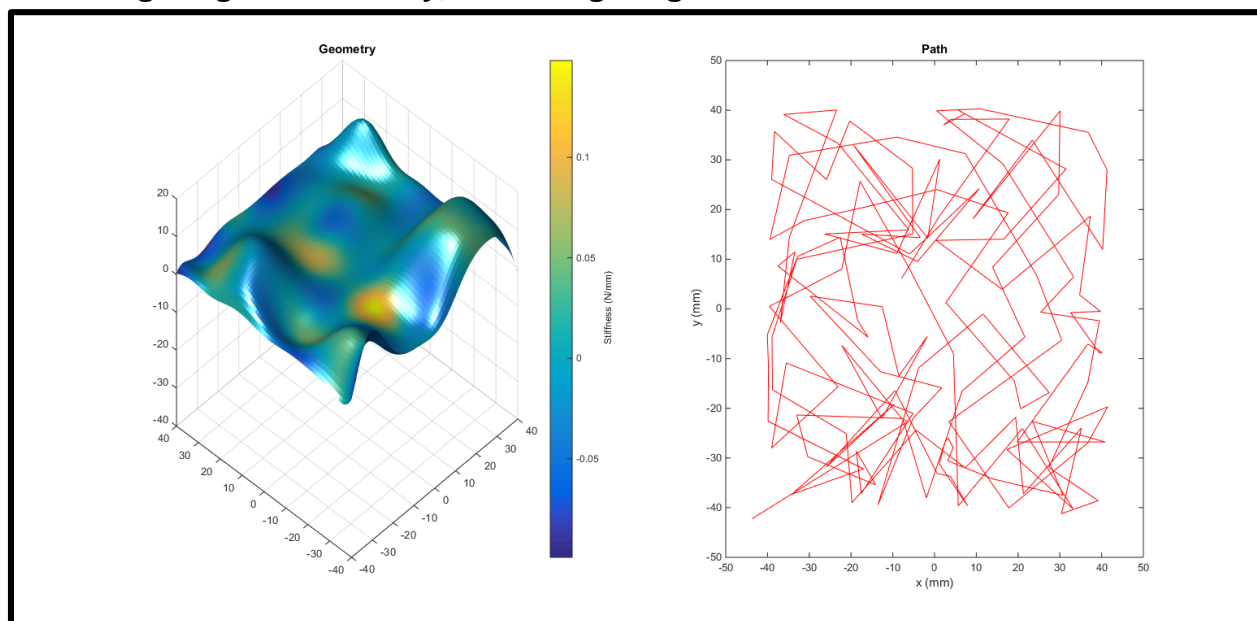
0% Weighting on Geometry, 100% Weighting on Stiffness

Figure 5: 0% weighting on geometry, 100% on stiffness .Results from dynamic maximum stiffness search. On the left, an image of the reconstruction using GPs. This was done with L= 35 mm, variance weight = 5, and mean stiffness weight = 1. The MSE was 17.84. On the right, the palpation trajectory

The result from this algorithm is also very interesting. The MSE is quite large, but don't let that detour you. The MSE is from the geometry reconstruction relative to the ground truth, and this algorithm focuses on only stiffness. As the stiffest point is along the subcutaneous wire (which has no geometry), the path taken was focused around the

13

subcutaneous wire (as can be observed by the high density in the middle). The

algorithm is flawed however in that it did not cover points along the entire length of the

wire. It only covered a region of the wire that was about a quarter of its entire length and

did not explore any further regions. This is in part due to the fact that the wire was so

thin that it was not probable the force sensor would land on it.

50% Weighting on Geometry, 50% Weighting on Stiffness

Figure 6: 50% weighting on geometry, 50% on stiffness .Results from dynamic maximum
combination search. On the left, an image of the reconstruction using GPs. This was done with
L= 35 mm, geometry mean weight = 1, geometry variance weight = 1, stiffness variance weight
= 5, and mean stiffness weight = 1. The MSE was 2.5412. On the right, the palpation trajectory

Of interesting value this dual weighting hit perhaps the most of the wire out of all our

algorithms. We believe this is because the geometry weighting helped the algorithm

search away from a local stiffness high point.

The resulting algorithm, retains virtually all of the advantages of max search, mainly

being ease of implementation and efficiency, while providing the ability to weigh multiple

characteristics in the search at once. As can be observed, the path still explored the

four corners (that contain interesting geometries), and also explored the center of the

grid (that had interesting stiffness). We consider this result to be a great success in that

it was able to seek two different features of the surface.

3.4 Maximum Change in Variance Search

Our most complex searching algorithm. The idea and core algorithm is primarily based

on the Active Learning Cohn or ALC method as described by Seo, Wallat, Graepel and

Obermayer.[6] In selecting the next palpation point from the N randomly sampled points

we compute the expected change in variance should we add that point to the training

set. In an attempt to encourage selection of palpation points in areas of interest we also

implemented a dynamic radius, and a an algorithm which we refer to as quad seek

which is described below.

.

3.4.1 Dynamic Radius

To encourage point selection in an area of high interest, we implemented a dynamically

changing box size in which to sample points to choose from. The side length is

calculated as follows:

$$Interest\ Factor = \frac{|Boring - Previous|}{Max\ Difference} + Correction\ Factor$$

$$L = \frac{Minimum\ L}{Interest\ Factor}$$

The behavior is such that if the last point was very similar to the *Boring* then the *Interest*

Factor will be much smaller than 1 and the box will grow. When a point is very different,

then the *Interest Factor* will be close to 1 and the box size will be close to its minimum value.

Boring is the level of the measured value which is the baseline or not of interest to the user. In our testing case where the phantom has a flat base, *Boring* is set to 0 when basing interest of the height. *Previous* is the height of the last sampled value and *Max Difference* is the largest expected difference between *Boring* and an area of interest, or in the case of geometry importance on our model, it would be the highest peak. *Correction Factor* is a small fraction used for more control the box size. *Minimum L* is the minimum side length of the box to sampled in. This equation for governing the box size was chosen because it ensures the box size cannot shrink too small, grows and shrinks predictably without the need for more model specific parameters to be chosen and is robust. The key thing to note is that the *Max Difference* should be estimated erring on the side of being overly large so that the box does not shrink when a very interesting area is discovered.

Through this indirect way of encouraging palpation in areas of interest, we open up a way for weighting and combining the search for multiple criteria of interest through a second *Interest Factor*. Namely, stiffness and height as before.

A major drawback of this method is that these parameters require prior knowledge of the data set, either through initialization from the user or extracted through an initial grid sample.

3.4.2 Maximum Change in Variance

The value of this search algorithm is in producing reconstructions which have low variance in the predicted values. By computing the expected change in variance of the potential palpation points we can choose the point which offers the most information to the model. The calculations are as follows:

$$\Delta\sigma^2_{\widetilde{y(\xi)}}(\tilde{x}) = \frac{(K_N K^{-1} K_* - Cov(\tilde{x}, \xi))^2}{(Cov(\tilde{x}, \tilde{x}) - K_*^T K^{-1} K)}$$

$$K_N = [C(x_1, \xi), \quad ...., \quad C(x_N, \xi)]$$

Where xi is a reference point and x tilde is the potential point to add to the training set. To utilize this we predict the change in variance at each potential point should each potential point be palpated and choose the point which results in the highest change in variance over all the sampled points.

Some pseudocode should help make this more clear:

```
PotentialPalpations = random(N);
for i in PotentialPalpations
{
        K_Ns(i) = K_N(PreviousItTrainedPoints, RefPt);
}
for i in PotentialPalpations
{
        PotentialPoint = i;

        for j in Potential Palpations
        {
                RefK_N = K_Ns(j);
                ChangeInVariances(j) = ChangeInVar(RefPt,PotentialPoint, K_N);
        }
        OverallChange = mean(ChangeInVariances);
}

nextPoint = max(OverallChange);
```

3.4.3 Quadseek

Our next algorithm we created for this method is the Quadseek. The idea behind this

algorithm is that while the Maximum Change in variance may choose the point which

offers the most information, this point may be an anomaly in an otherwise well covered

area. To counter this, we divide the box of potential points into four quadrants. The

quadrant with the highest average expected change in variance is then chosen from to
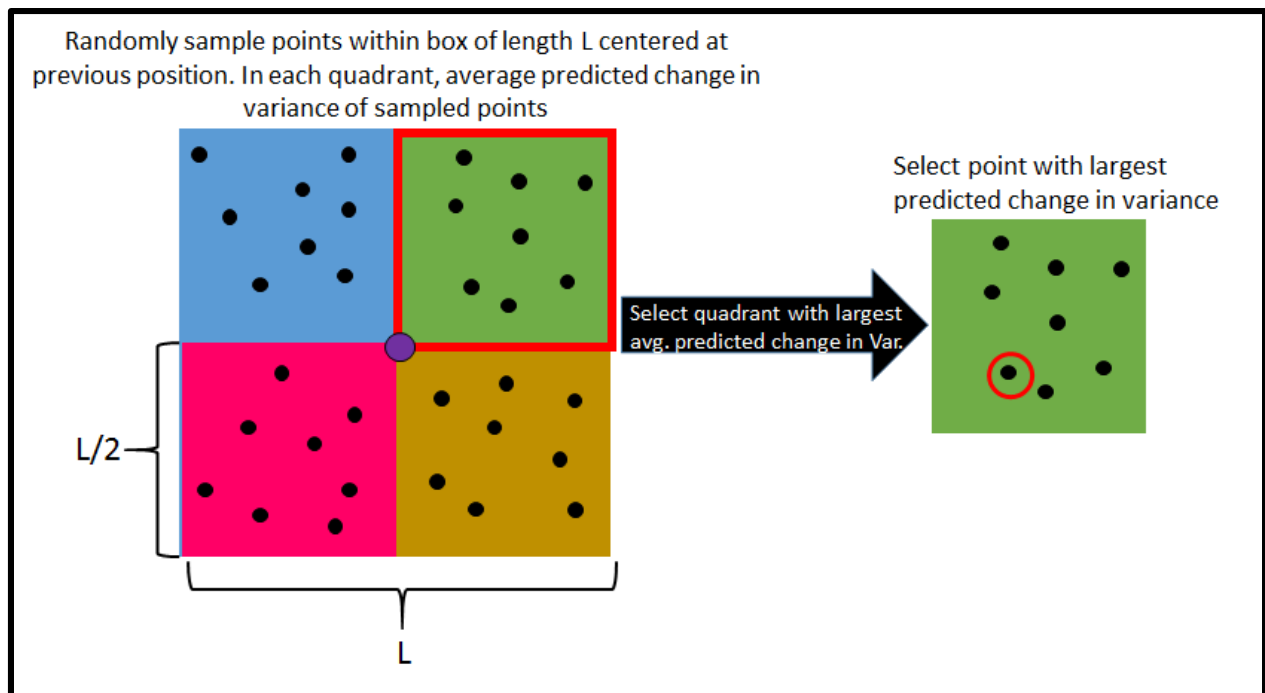
pick the point with the maximum change in variance.

Figure 7: A teaching schematic outlining out "quadseek algorithm"

3.4.4 Results

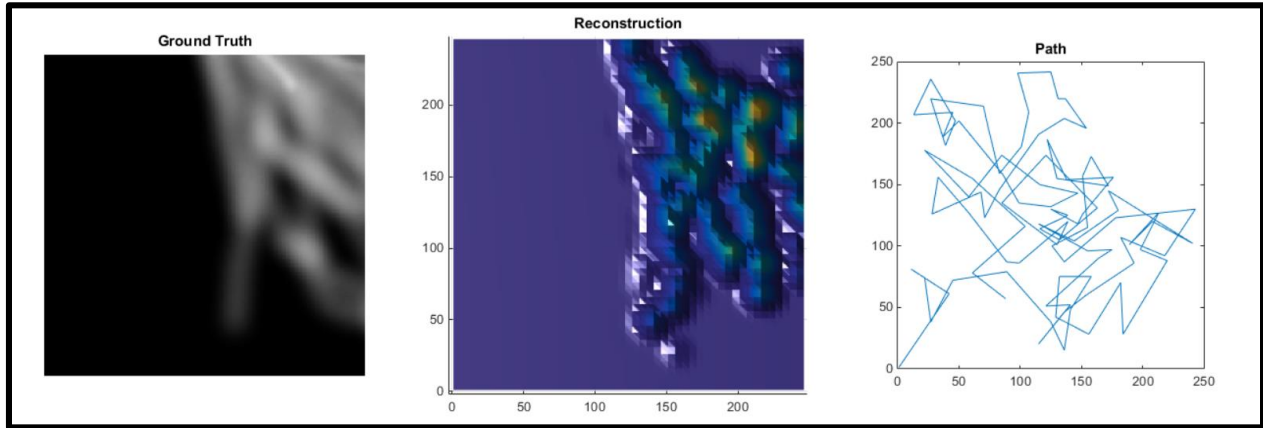## Initial Testing on an Image

Figure 8: Here, we show images of our initial testing (prior to receiving a phantom). We performed a GP with the value being pixel density. On the left, the ground truth (image used). In the middle, we have the reconstructed image using a GP. On the right, we see the path taken along the image (for choosing the next point

Here we performed some initial testing on an image. The image, is of concentration gradients within a lake, as you can see the results are semi accurate however they varied drastically every time the algorithm was run. Due to the decent performance on some iterations, hoping for better, we proceeded to test it on data that was more realistic to our real life model.

Toy Data Set: For baseline validation of results of our algorithms we tested on a toy

data set which we created. The toy dataset, seen below, was built to resemble the

phantom in hump size and spacing but with simple normal distributions. This allows the

gp to very easily recompute the results and for us to iterate quickly, rather than wait an

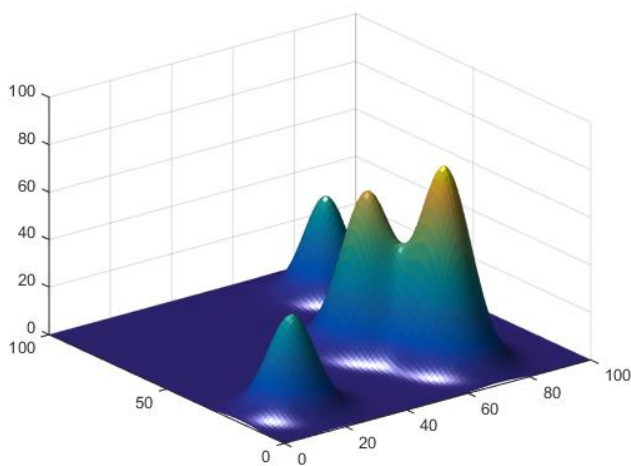hour to sample a sufficient number of point.

Figure 9: Toy data set plotted

We tested our algorithm on this data set with a significant number of variations in radiuses, weighting, etc but could never get it to behave the way we wanted. Below you can see a representative path of 150 points. Compare this to the path of our adaptive max search algorithm on the same data set, also of 150 points, where you can see just how much more efficient the adaptive max search is. It is shown below:
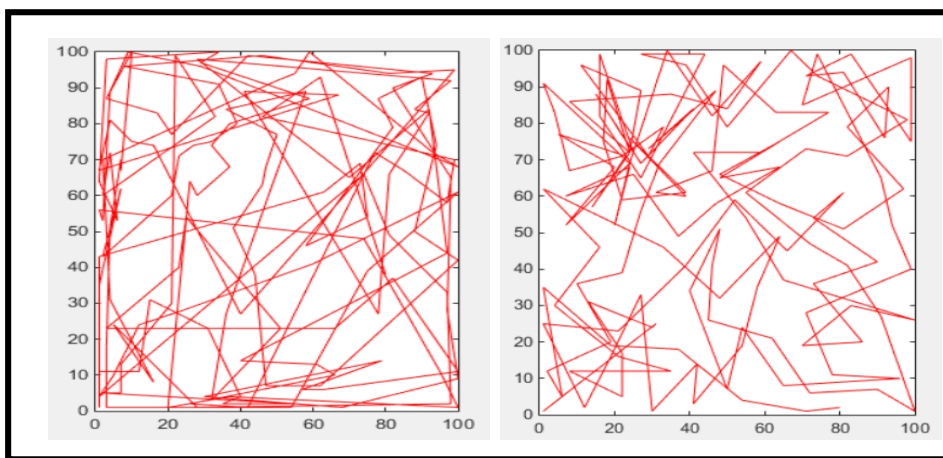
Figure 10: Palpation trajectories. On the left, the palpation trajectory of the representative path of 150 points. On the right, our adaptive max search algorithm

Similar results were recreated on the phantom, with the Change in Variance Search proving little better than a random path search. An additional disadvantage of the change in variance search is the very large variations in results from one test to another and high sensitivity to hyperparameters.

4. Discussion

4.1 Personal Lessons

It is important, and exciting for us to note that this project was largely a success. As will

be noted on in the below management summary, we achieved all of our deliverables

that we set in our initial project plan presentation (with some minor alterations). Further,

our team certainly learned a lot. When we started a the beginning of the semester,

neither of us knew anything about Gaussian Processes. We also had minimal robotics

experience(CIS1). Now, to discuss the results of our paper.

4.2 Algorithm Choice Recommendations/Future Algorithmic Work

As noted previously there is no one size fits all algorithm for an optimal search

trajectory for all models. This is particularly true when the user wishes to search for

multiple characteristics such as stiffness and height. When the structure is very basic or

initialization can be done nearby the area of interest a simple max search is the best

option. However if this is not possible the adaptive max search is the best performing

algorithm. The change in variance algorithm, is only useful in highly complex structures.

If this were the case though, we would recommend removing the dynamic radius

because of the required initialization.

An additional recommendation we have is to perform a sparse palpation grid

initialization of the structure before running any of our algorithms. We expect this to

drastically improve the results of all our algorithms although our previous

recommendations on choice still hold.

4.3 Limits of the Approach

4.3.1 Gaussian Process

Apart from the search algorithms limits there is also inherent limits within the gaussian process. Because it models structures as gaussians, it does a very poor job modeling a structure such as a thin wire without sampling almost all of the wire. This is a problem when searching because if the wire is missed by even a small amount the gp will predict that there is no structure in the near vicinity. Of course, you could change the length scale to be smaller but then when the wire is palpated, future predictions will still predict that there is no nearby structure anyway. Additionally, this would not work for varied structures within a tissue. Again, one size fits all, is a very difficult task.

The only way to get around such problems is by limiting the box from which the palpated points are chosen to a very small area. This though will take a very long time to sample and effectively turns the sampling into a grid search in the number of points taken or could, worse miss wide swaths of data because of its limited view.

4.3.2 Stiffness and Surface Calculations

The approach is also limited in the assumption that the stiffness is linear. A better approach would not have this approximation.

There is also a limit in where the stiffness and surface geometry are computed as a depth of the true surface. For example, a tumor might reside 3mm below the surface, here the stiffness would be much higher than the stiffness above it. Combining underlying structures and stiffness would need a very advanced modeling technique but averaging stiffness without incorporating this could result in misclassifying stiffnesses when their average stiffness as a function of depth is similar.

5. Management Summary

At the beginning of the semester, we defined our deliverables as follows. Our minimum deliverable with a geometry reconstruction of a sample image using Gaussian process and cross entropy optimization (to be used to select the next point).[5] Our expected deliverable was a geometry reconstruction of sample tissue with gaussian process and cross entropy optimization. And our maximum deliverable was to use the geometry reconstruction of the sample tissue, create a model of stiffness within the tissue. The main difference between these preliminary deliverables and what we actually achieved was we decided not to use cross entropy optimization to select the next point. This was under the guidance of our mentor Dr. Kobilarov. Instead, as we have shown, we have implemented various other ways to select the next point. Further, we also differed from our initial maximum deliverable slightly. While we did accomplish a model of stiffness, the stiffness model was independent of geometry. We choose to do this because we made the assumption that stiffness and geometry are, in fact, independent of one another (which is a reasonable assumption to make if the surface is human tissue).

There are several things that can hopefully be accomplished in the future. The first would be to test our current system on other stiffness distributions. Currently, the phantom we used had its main stiffness above the subcutaneous wire. It would be interesting if the stiffness was more circular in shape (as opposed to a line). We believe our search algorithms would perform much better in this case. Further, a circular stiff region has several implications in biology such as the shape of a tumor. Another aspect to look at in the future would be to assume stiffness to be a non-linear model. In our model, we assumed stiffness obeyed Hooke's law (ie stiffness was proportional to

depth). However, this is not necessarily the case. The last thing to look at in the future would be to implement the GP with co-dependent outputs. So no longer making the assumption that stiffness and geometry are independent.

We certainly learned a lot. Firstly, while GPs are versatile, they do have their limits in terms of what they can model. They work much better for circular shapes, as opposed to lines (like the subcutaneous wire). Secondly, adaptive searching is no small feat whatsoever. Initially, our search algorithms did not work as we planned whatsoever. Also, as can be observed by our maximum change in variance search algorithm, algorithms that work well on simulated data, may not perform perfectly in practice.

As we were only a group of two, we work together for the most part. Major contributions by Ben were Quad-Seek (3.4.3), the max search (for height and stiffness). Major contributions by Nate were maximum change in variance (3.4.2) and the adaptive maximum search.

6. Technical Appendices

     All of our code can be found in a zip file on our wiki page. Feel free to look!
We will now present a brief user's manual to all of the code. As we did not construct the codes that controlled the robot (those were written by Preetham), we will not inlcude the documentation. However, if one desires to access these codes speak with Preetham. They are well documented on GitLab.

The gp object used throughout the code consists of the following.

gp.sf - function noise
gp.sn - signal noise
gp.l - length scale
gp.xs - independent points for training set
gp.sf - dependent points for training set
gp.a - inverse of training set covariance matrix, computed and saved for efficiency

gp_test - A driver script for testing the algorithms on imported data, not for use with the cartesian stage

gp_online_train: Driver for performing the search and creating the gaussian process.
Inputs: gp, gpk, and Data. gp and gpk should take the form of gp.sf, gp.sn and gp.L.
Output: gp, gpk

gp_train: Trains the gaussian process on the data within it. Inputs: gp, Output: gp

gp_train_update: For retraining the data set when one point has been added to the gp since it was last trained. Input: gp, Output: gp

gp_pickNext_max_combo: This is used to implement the dynamic max search (section 3.3)
Inputs: The gaussian process for geometry (gp) and the gaussian process object for stiffness (gpk)
Outputs: The selected point (point)

gp_pickNextALC_Single.m: This is used to implement the search algorithm described in section 3.4.
Inputs: The gaussian process for geometry (gp) and the gaussian process object for stiffness (gpk)
Outputs: The selected point (point)

ComputeK.m: Computes the stiffness of a given point

26

Inputs: Force (f), depth (z), and surface height (z0)
Outputs: Stiffness

gp_sqexp.m: Is used to compute an element of the the covariance matrix.
Inputs: The gp object that contains the covariance matrix(gp), the two points that the covariance is being computed between (xa and xb)
Outputs: The computed covariance element (f)

createStiffnessModel.m: This creates a sample stiffness model that can be used for simulation testing purposes
Outputs: After this script has run, it saves the stiffness model in the object SIM_data.mat

PlotPhysicalResultsColorMap.m: This function allows you to plot your GPs for geometry and stiffness in one graph for user-specified test points
Inputs: The geometry GP (gp), the stiffness GP (gpk), and the number of points you desire to plot (resolution)
Output: The image (called reconstruction)

We also included some sample data:
GridData.mat: Data recorded from an experiment that sampled from a uniformly sized 18X18
gridSample.mat: This is the same data from GridData.mat, yet it includes every single force value that was recorded during the experiment

ReTune - A script for importing data from the robot, adjusting hyperparameters and running PlotPhysicalResultsColorMap

Citations

1. C. E. Rasmussen & C. K. I. Williams, Gaussian Processes for Machine Learning, the MIT Press, 2006, ISBN 026218253X. c 2006 Massachusetts Institute of Technology.
2. Ebden, Mark. *Gaussian Process for Regression: A Quick Introduction*. N.p., Aug. 2008. Web.
3. Neal, R.M.: Regression and classification using Gaussian process priors (with discussion). In Bernardo, J.M., et al., eds.: Bayesian statistics 6. Oxford University Press (1998) 475–501
4. Williams, C.K.I.: Prediction with Gaussian processes: From linear regression to linear prediction and beyond. In Jordan, M.I., ed.: Learning in Graphical Models. Kluwer Academic (1998) 599–621
5. Kroese, D. "The Cross-Entropy Method." *A Unified Approach to Combinatorial Optimtimization, Monte-Carlo Simulation and Machine Learning*. By R. Rubinstein. N.p.: n.p., n.d. N. pag. Print.
6. Seo, S., Wallat, M., Graepel, T., Obermayer, K., *Gaussian Process Regression: Active Data Selection and Test Point Rejection.* Department of Computer Science, Technical University of Berlin, 2000.