POINT-BASED REGISTRATION OF BRACHYTHERAPY IMPLANTS

by

LAUREN ELIZABETH GORDON

A thesis submitted to the
School of Computing
in conformity with the requirements for
the degree of Master of Science

Queen’s University
Kingston, Ontario, Canada
December 2011

Copyright © Lauren Elizabeth Gordon, 2011
Abstract

Prostate brachytherapy, a treatment for prostate cancer, was a procedure that typically involved placing radioactive sources in a cancerous prostate using percutaneous needles. The placement of these sources determined the dose that the prostate and healthy tissues surrounding it received. However, because a needle could bend, tissue could deform, and a patient could move, each source may have been displaced from its planned position. This source misplacement could later cause some cancer to be spared or healthy organs to be further damaged. To better understand patterns of source misplacement, and eventually reduce the phenomenon, this work matched and registered implanted sources with their planned positions.

Each implant was registered to its plan using a sequence of four successive registrations. A rough initial registration was first found, using features known in the planned dataset and estimated from the implanted dataset. Second, subsets of sources were reconstructed in the implanted dataset. The implanted sources were next matched to the planned sources using the subsets as constraints. Finally, the optimal rigid transformation between the implants and the plan was found.

The algorithm was tested on both simulated and clinical datasets. Simulations placed limits on how properties of the subsets affected registration accuracy. When tested on 9 clinical datasets, the algorithm found 100% of correct plan-implant source
matches within seconds on commonly available computers. When the implanted strands were reconstructed as sine waves, 97% of strands had an amplitude of less than 2mm. The clinical accuracy result generally agreed with simulation: subsets with amplitudes less than 2mm were expected to produce an accuracy $\geq 90\%$. The high accuracy of the algorithm may enable its use in finding patterns of source misplacement. The fast run-time of the algorithm may additionally make it useful for use in a clinical setting.
Acknowledgments

There are so many people who have made this document possible, who have and research over this year an absolute pleasure.

I couldn’t be more grateful for the mentorship of my supervisor, Dr. Randy Ellis. Socrates said a wise man knows he knows nothing; Randy has taught me to question what I think I know best and, in doing so, discover how much more there is to learn. Through his guidance and incisive questioning, he’s pushed me to think deeper, question harder, and perform at a higher level than I ever have before.

I’m equally grateful to my other supervisor, Dr. Gabor Fichtinger. Gabor has always advocated for me, even and especially when I’ve doubted myself.

I am especially indebted to Dr. Ehsan Dehghan. His enthusiastic contribution to this work, through conversation and concept development, was indispensable to its success.

Furthermore, I’d like to thank Dr. William Morris and Dr. Mehdi Moradi, who collected and provided the clinical data for this work.

I’d like to express my appreciation for the huge amount of time and effort my committee members, Dr. Parvin Mousavi and Dr. Tamas Ungi, spent critiquing this massive work, in addition to contributing further comments at my defense. Additionally, I’d like to thank Dr. Bob Crawford for his wonderful facilitation.
I’m so appreciative of my professors, both from my undergraduate and my master’s, who helped me gain knowledge, insight, and perspective that I will carry throughout my life. Among their numbers is Dr. Alan Ableson whose joy for everything mathematical is highly contagious.

I’d like to thank Dr. John Schreiner and Dr. Greg Salomons for taking me, a non-physicist, under their very knowledgeable wings. I never thought I’d have such an enjoyable and educational time being grilled on medical physics.

I couldn’t go without thanking my former supervisor and dean, Ken Rose. He gave me my first research position, in spite of having met me at a formal reception with candy-apple red hair. I continue to value his mentorship immensely: he’s pushed me to jump outside my mental box in research, academia, and teaching and helped me choose the classes that would take me from an undergraduate degree in life science to a master’s degree in medical computing.

I’d also like to thank Lynda Yearwood, my incredible high school computer science (and classics!) teacher. She gave me a love of programming, and the cognitive framework to understand the programming languages I’ve since encountered – a programmer’s Babel fish of sorts. It’s a lasting gift I will never forget.

The great senses of humour and amazing kindness of my labmates and friends from this department and others have made this past year such a pleasure. Amy Morton, Sima Zakani, Elodie Lugez, Mark Wu, Nick Meng, Nicolle Domnik, Jessica Parker, and others whom I’ve inadvertently forgotten: you guys are the best!

To my parents: Joanne and Ian Gordon. You raised me to become the person I am today, and have supported me throughout my life and throughout this degree. Deferring medical school was a huge decision with an unclear payoff that has since
yielded fruit. Thank you for continuing to trust in and support my judgement. I love you, Mom and Dad.

Finally, to my partner in life, Raffi Rush: thank you for reading over these acknowledgements, and for being there. I couldn’t have done it without you.

I gratefully acknowledge support in part from the Natural Sciences and Engineering Research Council of Canada, and from the Ontario Graduate Scholarship program.

“Spring is like a perhaps hand
(which comes carefully
out of Nowhere)arranging
a window,into which people look(while
people stare
arranging and changing placing
carefully there a strange
thing and a known thing here)and
changing everything carefully
spring is like a perhaps
Hand in a window
(carefully to
and fro moving New and
Old things,while
people stare carefully
moving a perhaps
fraction of flower here placing
an inch of air there)and
without breaking anything.

“

- e.e. cummings
Table of Contents

Abstract i
Acknowledgments iii
Table of Contents vi
List of Tables x
List of Figures xi

Chapter 1:

Introduction ................................................. 1
1.1 Prostate Brachytherapy ................................. 2
    1.1.1 Before the procedure ............................ 4
    1.1.2 During the procedure ........................... 6
    1.1.3 After the procedure ............................ 6
1.2 Complications ...................................... 6
1.3 Merit of Treatment .................................. 8
1.4 Problem Formulation ................................. 9
1.5 Organization of This Thesis ......................... 10

Chapter 2:

Review of Relevant Literature ............................... 12
2.1 Implant Misplacement and Complications .......... 12
    2.1.1 PSA (bFFF) is an Early Marker of Recurrence .... 13
    2.1.2 Suboptimal Dose Predicts Biochemical Failure .... 13
    2.1.3 High Dose is Linked to Complications ............ 14
    2.1.4 Seed Misplacement Can Lead to Poorer Dose Coverage ... 15
    2.1.5 Misplacement in Simulation .................... 15
    2.1.6 Misplacement of Encapsulated Devices ............ 16
2.2 Quantifying Device Misplacement ................. 17
    2.2.1 Intraoperative Planning ....................... 17
2.2.2 Tissue Deformation Models ........................................ 18
2.3 Relevant Registration Methods ........................................ 19
  2.3.1 Matching Devices to Implant Plans .............................. 20
  2.3.2 Registration of Point Sets ......................................... 23
  2.3.3 Registration of Matched Points .................................. 24
  2.3.4 Computational Observations ...................................... 25
2.4 Summary ................................................................. 26

Chapter 3:

Materials and Methods ...................................................... 27
  3.1 Terminology ............................................................ 27
  3.2 Collection and Selection of Clinical Data ............................ 29
  3.3 Simulation of Implanted Devices ..................................... 31
  3.4 Testing the Simulated Data ........................................... 40
  3.5 An Algorithm for Finding Stranded Beads .......................... 41
    3.5.1 An Initial Alignment ............................................. 43
    3.5.2 Strand Reconstruction ........................................... 47
      Partial Reconstruction and Alignment ............................... 48
      Full Strand Reconstruction ......................................... 55
    3.5.3 Needle-to-Strand, Seed-to-Bead Matching ...................... 58
    3.5.4 Finding the Optimal Transformation ............................ 61
    3.5.5 Errors Introduced by Automated Image Processing .......... 62
      Missing Beads ....................................................... 63
      Extra Beads ....................................................... 64
      Spurious Beads .................................................... 65
      A Heuristic Algorithm for Managing Input Errors .............. 65

Chapter 4:

Results ................................................................. 68
  4.1 Evaluation Process ................................................... 68
  4.2 Simulation – Wavelength $\lambda$ and Amplitude $\Lambda$ .............. 70
    4.2.1 Registration Accuracy ......................................... 71
    4.2.2 Rotational Error ................................................ 73
  4.3 Simulation – Translation $\delta$ and Amplitude $\Lambda$ .............. 75
  4.4 Simulation – Translation $\delta$ and Splay $\phi$ .................... 77
  4.5 Simulation – Plane Angle $\theta$ and Splay $\phi$ .................... 77
    4.5.1 Registration Accuracy ......................................... 80
    Rotational Error ..................................................... 80
  4.6 Results with Clinical Implants ...................................... 82
# Chapter 5: Discussion and Conclusions

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1</td>
<td>Results Using Simulated Data</td>
<td>86</td>
</tr>
<tr>
<td>5.2</td>
<td>Results with Clinical Data</td>
<td>86</td>
</tr>
<tr>
<td>5.2.1</td>
<td>A Heuristic Algorithm for Input Errors</td>
<td>88</td>
</tr>
<tr>
<td>5.2.2</td>
<td>Strand Attributes</td>
<td>89</td>
</tr>
<tr>
<td>5.3</td>
<td>Contributions</td>
<td>90</td>
</tr>
<tr>
<td>5.3.1</td>
<td>Development of a brachytherapy registration algorithm that used strands as a constraint</td>
<td>90</td>
</tr>
<tr>
<td>5.3.2</td>
<td>Discerning, via simulation, how some variations in implanted seed sets might affect registration accuracy</td>
<td>90</td>
</tr>
<tr>
<td>5.3.3</td>
<td>Demonstration of an algorithm that could successfully register implanted seed sets to plans</td>
<td>91</td>
</tr>
<tr>
<td>5.3.4</td>
<td>Determination of relevant quantitative properties of implant sets</td>
<td>91</td>
</tr>
<tr>
<td>5.4</td>
<td>Future Work</td>
<td>92</td>
</tr>
<tr>
<td>5.4.1</td>
<td>Extending the scope of the algorithm to loose seeds</td>
<td>92</td>
</tr>
<tr>
<td>5.4.2</td>
<td>Creating an anatomical seed misplacement model</td>
<td>92</td>
</tr>
<tr>
<td>5.4.3</td>
<td>Using strands for ultrasound-fluoroscopy registration</td>
<td>93</td>
</tr>
<tr>
<td>5.5</td>
<td>Conclusion</td>
<td>93</td>
</tr>
</tbody>
</table>

**Bibliography**  

<table>
<thead>
<tr>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>95</td>
</tr>
</tbody>
</table>

**Appendix A:**

<table>
<thead>
<tr>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>109</td>
</tr>
<tr>
<td>A.1</td>
<td>General Notation</td>
</tr>
<tr>
<td>A.2</td>
<td>Notation for Strands, Needles, and Related Concepts</td>
</tr>
<tr>
<td>A.3</td>
<td>Notation for Simulations</td>
</tr>
</tbody>
</table>

**Appendix B:**

<table>
<thead>
<tr>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>113</td>
</tr>
<tr>
<td>B.1</td>
<td>The Assignment Problem</td>
</tr>
<tr>
<td>B.2</td>
<td>Procedure of the Hungarian Algorithm</td>
</tr>
<tr>
<td>B.3</td>
<td>Foundations of the Hungarian Algorithm</td>
</tr>
</tbody>
</table>

**Appendix C:**

<table>
<thead>
<tr>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>135</td>
</tr>
<tr>
<td>C.1</td>
<td>Distance and angular costs</td>
</tr>
<tr>
<td>C.1.1</td>
<td>Hypothesis</td>
</tr>
<tr>
<td>C.2</td>
<td>Weighting of $\alpha_1$ and $\alpha_2$</td>
</tr>
<tr>
<td>C.2.1</td>
<td>Broad survey: $0.1 &lt; \alpha_1 &lt; 0.9$</td>
</tr>
<tr>
<td>C.2.2</td>
<td>Exploring a maximum: $0.05 &lt; \alpha_1 &lt; 0.3$</td>
</tr>
</tbody>
</table>
C.2.3 The original hypothesis: $0.01 < \alpha_1 < 0.08$ ......... 142
C.3 Weighting of $\beta_1$ and $\beta_2$ .................................................. 144
C.3.1 A first local maximum: $\alpha_1 = 0.25, \alpha_1 = 0.75$ ............... 144
C.3.2 A second local maximum: $\alpha_1 = 0.15, \alpha_1 = 0.85$ ............. 146
C.3.3 The original hypothesis: $\alpha_1 = 0.0656, \alpha_1 = 0.9344$ ........ 148
C.4 Implications ................................................................. 150
### List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Some characteristics of the device sets and the prostates for which the device set was planned, sorted by prostate volume (PV) and noting prostate treatment volume (PTV). Standard deviations (σ) of the devices from the plans were calculated based on an initial alignment, described in Section 3.5.1. Data from rows in gray were arbitrarily selected as the basis for simulated data.</td>
<td>30</td>
</tr>
<tr>
<td>3.2</td>
<td>Parameters of the simulated bead set. Changing any of these parameters changes patterns in the bead set; a wide variety of conditions were used to tune and test the algorithm.</td>
<td>40</td>
</tr>
<tr>
<td>4.1</td>
<td>The value of parameters chosen for the baseline case.</td>
<td>69</td>
</tr>
<tr>
<td>4.2</td>
<td>Strand reconstruction parameters used for simulations and clinical results. s is the spacing (in cm) currently being sought in the bead set.</td>
<td>84</td>
</tr>
<tr>
<td>5.1</td>
<td>Some constraints on strand parameters that produced an accuracy above 90%.</td>
<td>87</td>
</tr>
<tr>
<td>5.2</td>
<td>Parametric bound for well-behaved strands. Bounds for individual parameters were consolidated from the results in Table 5.1.</td>
<td>87</td>
</tr>
<tr>
<td>C.1</td>
<td>Strand reconstruction parameters used for simulations and clinical results. s is the spacing (in cm) currently being sought in the bead set.</td>
<td>135</td>
</tr>
<tr>
<td>C.2</td>
<td>Distribution of costs in the bead cloud. The mean of the costs for the most common spacing is reported, as is the standard deviation of those costs.</td>
<td>136</td>
</tr>
</tbody>
</table>
List of Figures

1.1 Diagram of a prostate brachytherapy procedure. .......................... 3
1.2 The plan determining seed positions for a prostate brachytherapy pro- cedure. ................................................................. 5
1.3 Prostate brachytherapy sources appear in a fluoroscopic image. ...... 7

2.1 Reprinted with permission from Su [74] ................................. 16

3.1 An example of a clinical data set. Needles were the planned planned lines and seeds were planned along the lines. Seed sets similar to this one were deformed to create bead sets. .............................. 31
3.2 Strands aligned in the XZ plane along sine waves. Amplitude and wavelength of these sine waves were varied during simulation. .... 32
3.3 Bead sets with two different orientations. It was possible to simulate the effects of any chosen strand rotation. ......................... 35
3.4 Bead sets with two different splay angles $\phi$. Simulation could test the effect of any value of $\phi$. .............................................. 37
3.5 Bead sets with two different amounts of random error $\delta$. Simulations could test the effect of different values of $\delta$. ....................... 39
3.6 A flowchart of the registration algorithm ................................. 43
3.7 The distance between two beads, $d_{ij}$, and the angle $\phi$ between the beads and a global axis. This diagram supposes that bead $j$ was more distal than bead $i$. .................................................. 48
3.8 An example of reconstructing strands in a bead set. All five spacings are $10\text{mm}$ in length. The best connections in the bead set were found and represented in $A$. .................................................. 52
3.9 From the example in Fig. 3.8, the reconstructed strands. .............. 53
3.10 Bead 3 has been removed from the example in Figure 3.8. This graph has one spacing at $10\text{mm}$ and one spacing at $20\text{mm}$. .......... 57

4.1 A simulated (a) and clinical (b) bead set. The simulated base case was intended to mimic the clinical bead set. ........................ 69
4.2 A bead set with long wavelength and small amplitude strands (a), and a bead set with short wavelength and large amplitude strands (b). . 71
4.3 Registration accuracy results over wavelengths and amplitudes. Three simulated bead sets were used to calculate each data point. 72
4.4 Rotational error created by changes in amplitude and wavelength. Three simulated bead sets were used to calculate each data point. 74
4.5 Registration accuracy results for random translations and amplitudes. Three simulated bead sets were used to calculate each data point. 76
4.6 Registration accuracy results for random translations and splay angles. Three simulated bead sets were used to calculate each data point. 78
4.7 A bead set with a radially symmetric planar orientation and a large splay angle (a), and a bead set with a fixed planar orientation and a small splay angle (b). 79
4.8 Registration accuracy results over planar orientations and splay angles. Three simulated bead sets were used to calculate each data point. 81
4.9 The radially symmetric case as compared to various fixed planar orientations. 82
4.10 Rotational error created by changes in splay angle and planar orientation. Three simulated bead sets were used to calculate each data point. 83
4.11 The charts of a perfect matching: the needle-strand correspondence (a), the strands only (b), and the seed-bead correspondence (c) for patient 1. 85

B.1 An example of a maximal and and non-maximal assignment. 120

C.1 Registration accuracy results over different values of $\alpha_1$ ($\alpha_2 = 1 - \alpha_1$) and $\beta_1$ ($\beta_1 = \beta_2$). Accuracy results from all nine clinical bead sets were averaged to generate each data point. 139
C.2 Registration accuracy results over different values of $\alpha_1$ ($\alpha_2 = 1 - \alpha_1$) and $\beta_1$ ($\beta_1 = \beta_2$), centered around the maximum of Figure C.1. Accuracy results from all nine clinical bead sets were averaged to generate each data point. 141
C.3 Registration accuracy results over different values of $\alpha_1$ ($\alpha_2 = 1 - \alpha_1$) and $\beta_1$ ($\beta_1 = \beta_2$), covering the hypothesis value of $\alpha_1 = 0.0656$. Accuracy results from all nine clinical bead sets were averaged to generate each data point. 143
C.4 Registration accuracy results over different values of $\beta_1$ and $\beta_2$. $\alpha_1$ has been held at 0.25 and $\alpha_2$ has been held at 0.75. Accuracy results from all nine clinical bead sets were averaged to generate each data point. 145
C.5 Registration accuracy results over different values of $\beta_1$ and $\beta_2$. $\alpha_1$ has been held at 0.15 and $\alpha_2$ has been held at 0.85. Accuracy results from all nine clinical bead sets were averaged to generate each data point. 147
C.6 Registration accuracy results over different values of $\beta_1$ and $\beta_2$. $\alpha_1$ has been held at 0.0656 and $\alpha_2$ has been held at 0.9344. Accuracy results from all nine clinical bead sets were averaged to generate each datapoint.
Chapter 1

Introduction

One problem in cancer treatment that seems particularly suitable for solution using computing methods is how to determine the accuracy of implanting radioactive devices – also colloquially called sources or seeds – that are intended to kill cancer cells yet which allow surrounding tissues to recover. The placement of a radiation source inside tissue is known as brachytherapy, and in the work that led to this dissertation the type of brachytherapy of interest was the permanent implantation of low-dose-rate devices (often called ”seeds”) because of their size and shape). The organ of interest was the prostate, which is a gland in the male reproductive system that is deep in the pelvic region of the abdomen.

There are three main interventions to treat prostate cancer: prostatectomy, external beam radiotherapy, and brachytherapy. Brachytherapy, as compared to prostatectomy and radiotherapy, appears to have an equivalent or improved cause-specific survival over a 4-year term [12]. Each intervention also has a unique complication profile, having advantages and disadvantages with respect to sexual, bladder, and renal complications [66]. The patient ultimately makes the decision of treatment modality,
or a combination thereof. Clinicians may recommend a particular modality based on the centre’s experience with these modalities, and the patient’s individual biological and lifestyle factors.

Outcomes in low-dose-rate prostate brachytherapy as a treatment for prostate cancer depend on the precise placement of the radioactive sources [72, 38, 73, 46, 45, 76, 18, 65, 70, 26, 8, 76, 49, 48]. Unfortunately, in the course of the procedure, these devices are inevitably misplaced from their planned positions to a lesser or greater degree. To evaluate such misplacement, the implanted devices must be matched to their planned locations. The work presented in this dissertation was the development and evaluation of an automated method for registering actually implanted devices to the planned implantation, using the way that the devices were typically inserted into tissue as a constraint.

Prostate cancer is the most common cancer in Canadian men: in 2011, about 25,500 cases are expected to be diagnosed of which 4,100 men are projected to die of their cancers [9]. Most of these men are expected to have small and slow-growing cancers that cause few, if any, problems. Those with larger and more rapidly-growing cancers are expected to experience pain, erectile dysfunction and difficulty urinating. Treatment therefore has to treat enough of the cancer to contain it and remove symptoms while avoiding complications from damaging surrounding tissues, where the complications are not unlike those caused by the cancer itself.

1.1 Prostate Brachytherapy

In permanent prostate brachytherapy, the radioactive devices are left in the prostate and continue to irradiate it over a period of many days [43, 55, 64]. Complications are
Figure 1.1: Diagram of a prostate brachytherapy procedure. Illustration by Xiao Xiao Ma, used with permission.
caused by the misplacement of these devices, mainly by irradiating sensitive structures. Recurrence is caused by the misplacement of seeds such that they leave the cancer intact.

Although there are many clinical frameworks for the procedure, a common one can be outlined as practiced at the time of writing. Notably, this framework used *stranded* sources: the sources were connected together on an encapsulated strand.

### 1.1.1 Before the procedure

Using a preoperative computed tomography (CT) scan or ultrasound scan of the patient’s prostate, the dosimetrist generated a treatment plan with the aid of planning software. This treatment plan specified where the radioactive devices were to be placed, to ensure an optimal radiation dose. Device placement was subject to two main constraints:

1. Needles could only be inserted at points on a 0.5cm grid, corresponding to insertion points in the template grid.

2. The space between two devices needed to be an integer multiple of 5mm, corresponding to physical spacers that had a length of 5mm.

A typical plan, therefore, consisted of two diagrams: the first showed the position of each needle insertion the template grid and the second showed the spacing of sources along each needle. Examples of these planning diagrams are shown in Figure 1.2.
(a) The positions of needles on the template grid, corresponding to the transverse plane of the body. The shape represents the depth of the needle, while the number within the shape represents the number of seeds on the needle.

(b) The spacing of sources along each needle. Spacings were multiples of 5mm, and were most commonly 10mm.

Figure 1.2: The plan determining seed positions for a prostate brachytherapy procedure.
1.1.2 During the procedure

During the procedure, radioactive I-125 sources were loaded onto needles with a 1cm space between most seed pairs. Fully loaded needles were then inserted by the clinician into holes in the rectilinear template grid. A transrectal ultrasound probe was used to monitor needle insertion as it occurred. A diagram of the procedure is found in Figure 1.3.

1.1.3 After the procedure

After the procedure, X-ray fluoroscopy or CT was used to verify the position of the sources and therefore the dose coverage provided. If the sources were of uniform activity, post-operative dosimetry was relatively simple. If sources are of non-uniform activity, they needed to be registered to their planned positions before the resultant dose could be calculated.

1.2 Complications

Complications from prostate brachytherapy included urinary incontinence, irritation or obstruction occurring in 8.9% of the treatment population, bowel function impairment in 4.4% of patients, and sexual dysfunction in 16.7% of patients [66]. Patient outcomes depended on device placement: optimal placement provided a dose that destroyed the cancerous tissue while avoiding damage to surrounding structures.

However, as seen in post-operative imaging, the radioactive devices were inevitably misplaced from their planned positions by such factors as needle bending, edema, and fluid motion [56]. Achieving an optimal dose to the prostate needed to account for
Figure 1.3: Prostate brachytherapy sources appear in a fluoroscopic image. The black "lines" are the sources as they appear in the X-ray image. The white dots were added subsequently: they show the radiative centre each source. Reprinted with permission from Dehghan [16].
patterns of misplacement in preoperative and operative solutions. Characterizing patterns of misplacement required an effective plan-implant registration algorithm; this dissertation presents such an algorithm, by using the geometry of device placement as a constraint.

1.3 Merit of Treatment

Studying and reducing complications in prostate cancer research is particularly important in light of recent studies.

Recently there has been considerable controversy over the treatment of prostate cancer. Many prostate cancers are slow-growing; it is said that men more often die with prostate cancer than from it. Recent studies suggest that screening [1, 67] and aggressive treatment [30, 67] of prostate cancer may not reduce the overall population mortality from prostate cancer to a clinically significant degree. A strong case was presented for “watchful waiting”: the treatment of prostate cancer if and only if it was fast-growing and likely to metastasize [68, 27].

Patients with cancers of high and intermediate risk are often treated, as are other patients who insist on treatment over watchful waiting because they believe it would achieve peace of mind. Such patients choose, instead of a cancer with little risk, a procedure with a very real risk of complications [66]. Because of these and other factors, the mitigation of treatment risk and complications is especially important.
1.4 Problem Formulation

The planned device positions were represented as points in a three-dimensional coordinate frame. The implanted device set, reconstructed from fluoroscopy, resided in a distinct three-dimensional coordinate frame, presumably not aligned with the planned frame. Registering planned seeds to implanted seeds relied on finding a correspondence between points in the plan and implants reconstructed from fluoroscopy. This correspondence was be used to find the optimal transformation between coordinate frames.

The work had four primary objectives:

- Development of a brachytherapy registration algorithm that used the strand on which sources lay as a constraint.
- Discerning, via simulation, how some variations in implanted seed sets might affect registration accuracy.
- Demonstration that the algorithm that could successfully register implanted seed sets to plans.
- Determination of relevant quantitative properties of implant sets.

To accomplish these objectives, a four-step process was implemented to successfully register implanted devices to planned points. The steps were:

1. **Aligning implants with the plan**

   Because the implanted and planned device positions initially resided in two separate, misaligned coordinate frames, an initial rough alignment was performed
so that further registration could occur. The centroid of each point set and rough direction of needle insertion were aligned in both datasets.

2. **Reconstructing implant sets**

The intended trajectories of the planned devices were known *a priori* but the sets of associated implanted devices were unknown. A cost heuristic was used to determine if two implants were physically adjacent, where an optimal set of implants would be the global minimum of the sum of the costs between adjacent implants. A matrix method, the Hungarian Algorithm, was used to find this optimal set.

3. **Matching implant sets to planned sets**

Having sets of implanted and planned devices, implanted strands were matched to planned strands. This resulted in a correspondence between implants and planned points.

4. **Finding the optimal transformation**

The locations of the implanted devices were rotated and translated to minimize the squared distance between their matches in the planned dataset, using an algebraic method provided by Sibson [69].

### 1.5 Organization of This Thesis

**Chapter 2:** Provides an overview of the relevant literature in the area. This will establish foundations of the research as a correlation between misplacement
and dose, and dose and patient outcomes. Current methods for reducing seed misplacement are discussed in the context of the proposed method.

**Chapter 3**: Presents the methods of the research. First the algorithm is detailed; next, simulated datasets are discussed; and finally, the clinical process and method of data collection are described.

**Chapter 4**: Gives results of applying the algorithm to simulated and clinical data. Accuracy with simulated datasets provides information about the capability of the algorithm; results with clinical data provide an initial feasibility study.

**Chapter 5**: Discusses the results and summarizes the contributions.
Chapter 2

Review of Relevant Literature

During the process of inserting tiny radiological devices into a prostate, they are often displaced some amount from their planned positions. After considering some motivations for why misplacement is worth studying, current methods used to quantify and reduce the misplacement will be discussed.

2.1 Implant Misplacement and Complications

When a surgeon implants brachytherapy devices, the intention is to place them according to a treatment plan. A number of factors can conspire to displace the devices from their ideal locations [56]. Because prostatic tissue deforms around an injecting needle, the needle itself bends in response to changes in tissue properties. Fluid motion and edema can carry already-implanted devices away from their planned positions. Such tissue distortion makes surgical judgments about the true planned position of a seed more challenging.
CHAPTER 2. REVIEW OF RELEVANT LITERATURE

This movement of devices away from their planned positions can have the unfortunate potential to change the dose distribution in the prostate and thus increase negative outcomes. An appropriately high dose to the tumor is the purpose of brachytherapy treatment, so the devices must irradiate all of the cancerous prostate in order to effectively prevent recurrence. The devices must, however, not overirradiate the urethra, rectum, and penile bulb; doing so risks the postoperative quality of life of the patient.

2.1.1 PSA (bFFF) is an Early Marker of Recurrence

Recurrence can be measured in a number of different ways: as overall survival, clinical distant metastases free survival (DMFS), disease-free survival (DFS), cause-specific survival (CSS), and local control (LC) [32]. A secondary marker can also be used for early detection as well as to monitor the progression of treatment.

Developed by the ASTRO consensus panel, biochemical relapse-free survival – also called biochemical freedom from failure, bFFF – is the fraction of the treatment population who did not report, over the course of any 12 month period, 3 consecutive increases in prostate-specific antigen measurements [14], which is a nonspecific marker of prostate cancer. bFFF has been found to be correlated to all of the above criteria (DMFS, DFS, CSS, LC) [32] and to general clinical failure [40].

2.1.2 Suboptimal Dose Predicts Biochemical Failure

Dose coverage of the prostate has been linked to bFFF in low-risk prostate cancer patients [4, 61]. The measurement used, the minimum dose covering 90% of the prostate (D90), was first reported by Stock et al. [72]. They found that bFFF was
correlated with a D90 up to 140Gy, which continues to be the current standard for I-125 implants. One representative study reported an 8-year bFFF of 82% for patients with optimal D90 dose coverage (I125: \( \geq 140\text{Gy} \), Pd103: \( \geq 100\text{Gy} \)) vs. 68% for non-optimal dose implants [38]. Another study reported a higher incidence of positive post-treatment biopsy in patients with lower dose coverage [73]. Dose coverage appears to be a good predictor of bFFF, likely having a causal role in preventing recurrence of cancer.

2.1.3 High Dose is Linked to Complications

Complications appear to be linked to a high dose given to healthy tissues surrounding the prostate. As a practical matter it is nearly impossible to prevent damage to proximal organs such as the urethra, rectum, and nerves. The dose measurements to each of these structures is a dose-volume histogram (DVH), reported as the volume of tissue that exceeds a given dose. Increased urethral doses have been implicated in urethral structures [46], incontinence [45], and general urethral morbidity [76, 18, 65]. Rectal complications, like bleeding and proctitis, are strongly correlated with the volume of rectum receiving a dose over 140-160Gy [70, 26, 8, 76]. Sexual dysfunction may result from radiation dose to the penile bulb [49, 48], but this is contested [71]. Sexual dysfunction is not correlated with dose to the neurovascular bundles [47, 37].

Recurrence and complications seem to be much more likely with a D90 \( \leq 140\text{Gy} \) and a high volume of sensitive tissue receiving this dose.
2.1.4 Seed Misplacement Can Lead to Poorer Dose Coverage

Dose coverage has commonly been calculated based on the placement of the devices. For any given radioactive source, the radiation dose falls off as the inverse square of distance. For multiple sources, the absorbed dose at any point is the sum of the dose contributed from each point source, which can be approximated as

\[ AD(\vec{p}) \propto \sum_{i=1}^{N} \frac{1}{||\vec{p} - \vec{d}_i||^2} \]

Brachytherapy devices, although physically similar to prolate ellipsoids, for radiotherapy purposes are well approximated by point sources or collections thereof. The variation in dose in a brachytherapy implant is thus determined exclusively by the placement of these brachytherapy devices. A key question is whether the pattern of seed motion changes the dose in a clinically relevant way.

Post-procedure device migration, measured over a 4–5 week course, has been well studied [59]. Clinical studies on intraoperative misplacement are more rare, likely because of the difficulty in accurately quantifying device misplacement.

2.1.5 Misplacement in Simulation

The effect of random device misplacement on dose has been studied extensively in simulation. An early study [15] suggested that random device misplacements of up to 2mm could cause as much as an 8% average undertose to the prostate; 4mm maximum misplacements showed a 16% average undertose. Su [74] noted that a 95% confidence interval within 5% of the planned dose was obtained with a standard deviation of misplacement of 1.5mm. A diagram of D90 vs. misplacement from the
paper is reprinted in Figure 2.1. Bues [7] did not report a clinically significant decrease in D90 with up to 4mm random displacements, but did observe an increase in urethral dose for the same displacements. As expected, rectal dose has been correlated with the proximity of posterior devices in the prostate to the rectal wall [75, 77]. No studies were located as to the effect of misplacement on dose to the penile bulb.

2.1.6 Misplacement of Encapsulated Devices

To mitigate dose coverage effect of random device migration, devices encapsulated with flexible tubes has been suggested [15]. This encapsulation appears to render the devices less susceptible to distant migration than loose devices [62]. However, recent evidence suggests that such implants may have equivalent [28] or poorer [51] dose coverage than unencapsulated implants, although a clinically significant difference in D90 is not notable in the literature searched. Encapsulated, or "stranded" devices
were used in all patients for the purposes of this work.

2.2 Quantifying Device Misplacement

A number of methods have been proposed and implemented for quantifying and/or reducing device misplacement. This discussion is limited to methods involving computational models, mainly planning the device positions and modeling device misplacements. Physical apparatus and methods of insertion are beyond the purview of this review.

Techniques used to quantify and reduce device misplacement range from intraoperative planning, inverse planning, and plan-implant based registration methods. The latter will be discussed in a separate section because of its centrality to this dissertation.

2.2.1 Intraoperative Planning

In a procedure room, a treatment plan can be modified to account for device misplacement and tissue swelling and deformation. This intraoperative planning typically involves iteratively recalculating the best treatment plan, knowing the devices already inserted and their probable needle tracts [60, 56]. Additional devices are not infrequently added to the treatment plan to account for device misplacement and edema.

Preliminary studies in intraoperative planning, with small samples, suggested that it performs comparably to preplanning alone, with comparable procedure times and
dosimetric outcomes [6]. However, upon improvements in the intraoperative planning method, it performed noticeably better than preplanning alone, achieving the prescribed dose in 91% of the prostate, vs. 86% with preplanning alone [60].

While intraoperative planning certainly appears to improve dose parameters, it also has its challenges. It has been suggested that intraoperative dosimetry in ultrasound may not accurately reflect postoperative dosimetry in CT: the current gold standard. Specifically, treatment planning in ultrasound results in a substantially underestimated dose to both the prostate and healthy tissues, especially the rectum [57]. Essentially, in an attempt to achieve optimal dose, healthy tissues may receive higher doses in the course of intraoperative planning.

Thus, intraoperative planning appears to lead to better dose coverage than preplanning alone. Dose underestimation appears to be an issue, especially as it pertains to dose in sensitive tissues like the rectum.

2.2.2 Tissue Deformation Models

If a preoperative plan is improved then less intraoperative modification may be needed. Attempts have been made at modeling prostatic tissue to optimize device placement.

Some work has been done modeling needle insertion with a non-linear finite element model. In simulation, the model has been used to optimize the needle placement to meet targets or device locations in the prostate [17, 36, 25]. One study modeled the effect of prostate edema on dosimetry, independent of other factors [10].

A limitation is that these tissue deformation models have been used almost exclusively in simulation. Because of the simplifications made in tissue and needle
properties, the accuracy of these models must be demonstrated on clinical data—which are not prominent in the literature, possibly because of the lack of robust plan-implant registration methods.

2.3 Relevant Registration Methods

In medical imaging, registration is the process of determining a mathematical relationship between two entities. Registration is most commonly between two sets of empirical data, or a set of empirical data and an abstract model of the data. This is a very large research area that has been the subject of many review articles, notably those of Maintz and Viergever [44] and Hill et al. [29].

Here, the interest is in how to register image-derived locations of imputed implants to a preoperative or intraoperative plan. This narrows the scope of registration methods to point-based ones, where "clouds" of points are to be matched.

One class of registration methods, plan-implant methods, attempt to identify the implanted devices in situ (and possibly also the needle used to insert them). Such methods have been suggested for use in identifying devices in an implant with non-standard dose coverages [2]. Knowing this detailed information, post-implant dosimetry can be performed more effectively.

A plan-implant method can also be used for quantifying misplacement. Specifically, if implanted devices are matched to their planned device positions, the plan and the implant can be registered. From this, the shape of a needle can be better quantified and the misplacement occurring in different quadrants of the prostate can be better quantified. If misplacements can be characterized, predictive planning can be performed more insightfully and effectively.
It should be noted that this class of methods cannot detect global rotations or translations of the plan with respect to the implant; it assumes a common centroid of both device sets.

### 2.3.1 Matching Devices to Implant Plans

A crucial element of registration is a *match*, which is a mapping between sets of points. Matching algorithms must deal with geometric distortion caused by tissue deformation, missing data from devices that were either not implanted or not detected, and spurious data from either additional devices that were implanted or from oddities in the detection process. Four prominent approaches to matching have been random consensus, simulated annealing, network flow, and assignment.

The RANSAC, or RANdom Sample Consensus, method [24] iteratively selected random sets of devices and, if they had a plausible needle trajectory, scored them according to their linear fit. The sets with the best scores, representing the most likely needles, were selected as the true needles. In 2009, Moradi used RANSAC to reconstruct needles amongst implanted devices reconstructed from fluoroscopy [52]. The algorithm detected 100% of needles in a brachytherapy phantom. However, when tested on four clinical datasets, the algorithm only detected 65.7% ± 12.2% of needles. This level of accuracy is not sufficient for robust plan-implant registration.

Simulated annealing, especially as the Metropolis algorithm [50] or its many variants, is a global optimization method that was inspired by a metallurgical process. Beginning with candidate distributions of states and rules for progressing between states, simulated annealing is a probabilistic way of estimating a global minimum. In 2003, Archambault proposed a method of registering implanted device positions to
planned device positions using simulated annealing on a heuristic function [2]. The heuristic used was simply the sum of the distances from the implanted device positions to their matched planned device positions. The matchings were altered until the simulated annealing algorithm found a local – ideally global – minimum in the heuristic. The process took 30-60s on a 1.3GHz processor and resulted in a registration accuracy of 89% for 58 simulated datasets. The algorithm was dependent on device density, falling to an accuracy of 80% at 2.25 devices per cm$^3$ (mean: 1.69 devices per cm$^3$). Because of its iterative optimization, the algorithm was not guaranteed to find a global minimum. Because the algorithm only used a distance heuristic to find matchings, it did not consider the continuity of strands.

Network flow is a method that uses local distance constraints to solve the problem of matching prostate implants. Chng et Al. [11] proposed a method that used a heuristic for matching an implanted device to a planned device, by including the distance between the planned and implanted device positions, as well as the likelihood that planned devices fell on the same needle. After network-flow optimization, simulated annealing was performed on the outcome of the previous step.

The algorithm resulted in an mean reconstruction accuracy 97.7± 0.4% with a runtime of 5-10s (processor not reported). Because there are thus two sets of objects – planned devices and implanted devices – and conditional heuristics defined between and within the two groups, the problem formulation had at minimum NP-complete complexity. An additional clinical difficulty was that a full set of planned positions is necessary to find the matches.

A number of sub-problems in bead-seed registration involve assigning some set of objects to another, which here means simply a mapping in the sense of elementary set
theory. Many of the subtleties arise because these mappings are not homeomorphisms, especially because they may not be single-valued maps (a member of the domain set could potentially be assigned to more than one member of the range set) or ”onto” (a member of the domain set might not be assigned). Finding an assignment is typically phrased as an optimization problem: there is a third set of parameters, and a real-valued objective function, such that some particular parameters specify an assignment that is a local minimum of the objective. One possible formulation of assignments might be to produce a 1:1 mapping with as few as possible ”missing” members of the domain, i.e., try to map as many beads to seeds as possible.

Whether it be assigning beads to seeds, strands to needles, or beads to other beads, the sub-problems considered in this dissertation can be all structured as a cost matrix or bipartite graph and solved using the same algorithm. The Hungarian Algorithm [41] was chosen as a fast, robust algorithm for solving these assignment problems. For this algorithm, the 1:1 nature of the mapping is an inviolable constraint and the objective function is a sum of positive values determined by each member of the domain and its image in the range of the mapping.

Solutions to this problem have deep roots in real analysis. It was originally called the Hungarian Method, with the intent that it would be carried out by hand, and as such it was described and popularized by Kuhn in 1955 [41]. Kuhn, however, was not the first to note this method: Ollivier and Sadik [58] note that Jacobi describes the assignment problem and a solution equivalent to the Hungarian Algorithm nearly a century before Kuhn; Jacobi’s manuscript was published posthumously in 1865 [34] (Translation: [33]).
In 1957, Munkres observed that the original formulation of the Hungarian algorithm terminated in $O(n^4)$ time [54] where $n$ was the size of the domain. A variation of the Hungarian Algorithm, presented by Kuhn in 1956 [42], was later observed to have an $O(n^3)$ time bound [21, 19]. This specific variant of the Hungarian Algorithm was used in every assignment problem whose solution is presented in this dissertation.

An excellent description of the Hungarian Algorithm and its variant have been provided by Cooper [13]. A detailed explanation of the Hungarian Algorithm is provided in Appendix material B.

### 2.3.2 Registration of Point Sets

A more general way to state the problem of matching implanted devices to a plan is as a point-based registration problem. Here, the problem reduces to finding a subset of implants and a subset of planned devices that optimize some objective function.

By far the most cited method is the Iterative Closest Point (ICP) algorithm of Besl and McKay [5] which breaks the problem into two phases: given a current registration transformation, find a match; then, from this match, re-estimate the registration. In typical implementations the empirical data (implants) are fixed and the nearest model point (plan) is taken as the match between data and model.

ICP, its variants, and related algorithms have been well discussed in the review articles cited above [44, 29] and will not be detailed here. One highly relevant algorithm that was needed for this research was a way of optimizing the registration of matched points.
2.3.3 Registration of Matched Points

It has been shown, in many different ways, that rigid matched-point registration can be reduced to finding a least-squares rotation of one zero-mean data set to another zero-mean data set. Creating zero-mean data is easily done for each set, by subtracting the arithmetic mean of all the vectors from each vector. The problem is then one of finding an optimal spherical displacement of one set to the other set.

A common representation of a set of 3D vectors is as a matrix, with each column being the position of a point. Here, if there were \( m \) beads (detected implants) that were matched to \( m \) seeds (planned device locations), the zero-mean set of seeds could be written as the \( 3 \times m \) matrix \( Q \). The matching beads could be placed in a matrix in order corresponding to the seeds, written as the \( 3 \times m \) matrix \( P \):

\[
\bar{I}_q = \frac{\sum_{k=1}^{m} I_{\vec{s}_{i_k}}}{m} \quad \bar{F}_p = \frac{\sum_{k=1}^{m} F_{\vec{b}_{j_k}}}{m} \\
\bar{I}_Q = [I_{\vec{s}_{i_1}}, I_{\vec{s}_{i_2}}, \ldots, I_{\vec{s}_{i_m}}] - \bar{I}_q \quad \bar{F}_P = [F_{\vec{b}_{j_1}}, F_{\vec{b}_{j_2}}, \ldots, F_{\vec{b}_{j_m}}] - \bar{F}_p
\]

\( \vec{i} \) are the indices of the seeds matched to the beads represented by indices \( \vec{j} \). Thus, seed \( \vec{i}_k \) was matched to bead \( \vec{i}_k \).

The optimal transformation is the \( 3 \times 3 \) matrix \( R \) such that

\[
\bar{I}_Q \approx R_2 \bar{F}_P 
\]
This problem is variously known as the Orthogonal Procrustes problem [69, 20], the least-squares rotation problem [3], and the absolute-orientation problem [31]. Sibson (cited within the better known work of Dorst) used the singular-value decomposition (SVD) in his derivation and then showed how to compute the optimum using only square roots of a symmetric matrix; Arun et al. used the SVD in the calculation; and Horn used the largest positive eigenvalue of a quaternion correlation matrix which, having a characteristic polynomial of fourth degree, can be solved in closed form.

Dorst [20] showed that Sibson’s and Arun’s methods are mathematically equivalent, and an empirical study [23] has shown all three to be computationally equivalent. For simplicity and ease of code portability, Sibson’s method is particularly attractive because, from the cross-correlation matrix $H$, the optimal registration of Equation 2.1 can be computed as

$$H = [P]Q^T$$

$$R_2 = (H[H^TH]^{-1/2})^T$$

### 2.3.4 Computational Observations

Naive implementations of conventional methods for point-based registration do not appear prominently in the literature, likely in part because of the fine structure of brachytherapy plans. A brute-force approach is exponential in the number of implants because, in the worst case, all subsets of a plan must be potentially matched to all subsets of implants (where the number of planned and implanted devices are
comparable in magnitude). This worst case arises from (a) planned devices not being implanted, (b) extra devices being implanted, and (c) substantial deformation of the plan by physical processes that include tissue deformation.

2.4 Summary

A brief survey of relevant literature shows that misplacement of prostate brachytherapy implants is a problem with significant clinical consequences. Existing registration methods show promise, both in providing an optimization framework and as a source of technical solutions to coordinate transformations.

One open research question is how to efficiently perform partial matching of a plan for device placement with the actual implants detected postoperatively. The remainder of this dissertation provides a method, presents results on both real and simulated data, then discusses these results to find strengths and weaknesses of the algorithms that were implemented as part of this research.
Chapter 3

Materials and Methods

This research required two forms of data as its materials, clinical data from implant procedures on live patients and simulated data derived from clinical data; the latter had controllable "errors" artificially introduced that exercised the matching algorithms in clinically relevant ways.

This chapter will first clarify the terminology, then describe the clinical data and the production of artificial data. The remainder of the chapter is devoted to the description of the methods used to match data to idealized models. All simulations and algorithms, with the exception of the Hungarian Algorithm, were programmed in MATLAB by this author.

3.1 Terminology

The terms planned seed positions and implanted seed positions have so far been used, somewhat casually, to describe the entities to be registered. Although these terms suffice for general discussion, better terms can encompass the type of data being
CHAPTER 3. MATERIALS AND METHODS

manipulated and the coordinate frame in which it is manipulated.

The first term to clarify is that for post-implant data, which heretofore have been referred to as implanted seed positions because, in a clinical setting, the small radioactive devices that are implanted into the prostate are commonly referred to as “seeds”. These devices can be imaged using fluoroscopy; from multiple fluoroscopic images, these real-world devices can be represented as geometric points. Here, such geometric points will be called beads.

The radioactive devices were implanted into living tissue using needles. Each of these needles bored a needle tract into the prostate. In a procedure room, the needle defined the connections of and distances between the devices it was used to implant. However, fluoroscopy imaged neither the needle nor the tract it made. Here, the term strand is defined as a representation of the relevant features of a needle tract: the connections and spacings of the beads are, ideally, those of the devices left by the needle in the tract.

Moving on to planned device positions, the needles themselves were loaded with radioactive devices placed at defined spacings along the lengths of the needles. Such a real needle can be simplified to be an ideal needle – perfectly straight, with seeds placed along its length. An ideal needle is thus represented as being placed precisely where it was planned to go, and with its seeds ending up in their planned positions. Here, this ideal situation will be described with the colloquial terminology: seeds are the objects that lie on needles, whose paths are defined exactly. We will represent these seeds and needles in a fixed coordinate frame. The beads and strands can then be rotated and translated as needed.

This work will use two principal coordinate frames: the ideal coordinate frame
housing seeds on needles, and the reconstructed coordinate frame housing beads on strands. Notations for the coordinate frames, and all related topics, is given in Section A.

Additional notation will be introduced and explained as needed.

3.2 Collection and Selection of Clinical Data

The clinical data were collected by Drs. William J. Morris and Mehdi Moradi at the Vancouver Cancer Center in Vancouver, British Columbia. Patients without complicating factors, such as heart disease, were selected to participate. In total, 9 provided sufficient data for consideration in this research.

The patients were implanted with radioactive $^{125}$I devices according to a conventional preoperative plan with intraoperative modifications. The procedure was performed under TRUS guidance. After the devices were implanted, they were imaged using a motorized C-arm fluoroscope (OEC9800, General Electric, Schenectady US). For each patient, five images were taken and used to reconstruct the geometric beads as a three-dimensional point set. First, implanted devices were segmented manually in each of the 50 patient images: 5 images were used for each of 10 patients. Next, 3D reconstructions were created from these manually segmented 2D images using the MARSHAL [35] algorithm with motion correction [16], using groups of three images to reconstruct the implanted device locations. Mean reconstruction error was less than 0.7mm, with an average of 0.5mm.

Some properties of the bead sets are shown in Table 3.1. The prostates varied by nearly a factor of 2 in volume and by over 30% in the number of devices used to treat the cancer. Three sets were arbitrarily chosen as being representative of the number
CHAPTER 3. MATERIALS AND METHODS

of needles and seeds used across this small patient population.

Table 3.1: Some characteristics of the device sets and the prostates for which the de-
vice set was planned, sorted by prostate volume (PV) and noting prostate treatment volume (PTV). Standard deviations (σ) of the devices from the plans were calculated based on an initial alignment, described in Section 3.5.1. Data from rows in gray were arbitrarily selected as the basis for simulated data.

<table>
<thead>
<tr>
<th>Patient ID</th>
<th>PV (mL)</th>
<th>PTV (mL)</th>
<th>No. of Devices</th>
<th>No. of Needles</th>
<th>σ_x (mm)</th>
<th>σ_y (mm)</th>
<th>σ_z (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>28</td>
<td>47</td>
<td>102</td>
<td>25</td>
<td>15.5</td>
<td>14.8</td>
<td>7.8</td>
</tr>
<tr>
<td>2</td>
<td>30</td>
<td>49</td>
<td>100</td>
<td>26</td>
<td>11.8</td>
<td>12.2</td>
<td>9.9</td>
</tr>
<tr>
<td>3</td>
<td>31</td>
<td>48</td>
<td>105</td>
<td>25</td>
<td>14.4</td>
<td>15.5</td>
<td>10.0</td>
</tr>
<tr>
<td>4</td>
<td>31</td>
<td>52</td>
<td>113</td>
<td>28</td>
<td>15.4</td>
<td>16.3</td>
<td>8.6</td>
</tr>
<tr>
<td>5</td>
<td>35</td>
<td>51</td>
<td>104</td>
<td>25</td>
<td>14.9</td>
<td>15.7</td>
<td>9.5</td>
</tr>
<tr>
<td>6</td>
<td>36</td>
<td>46</td>
<td>105</td>
<td>20</td>
<td>17.3</td>
<td>15.8</td>
<td>9.5</td>
</tr>
<tr>
<td>7</td>
<td>43</td>
<td>61</td>
<td>115</td>
<td>24</td>
<td>15.5</td>
<td>15.5</td>
<td>10.6</td>
</tr>
<tr>
<td>8</td>
<td>45</td>
<td>65</td>
<td>122</td>
<td>22</td>
<td>18.5</td>
<td>15.2</td>
<td>9.6</td>
</tr>
<tr>
<td>9</td>
<td>54</td>
<td>71</td>
<td>135</td>
<td>30</td>
<td>16.2</td>
<td>16.1</td>
<td>11.2</td>
</tr>
</tbody>
</table>

Colloquially, clinicians and other experts observed four major variations from ideal placements. These variations, used to produce simulated data, were:

**Waviness:** Data were perceived to have some degree of periodic variation from mean lines, which could be simulated by sinusoids.

**Orientation:** Oscillating data did not consistently vary along anatomical planes, which could be simulated by rotating planar sinusoids.

**Splay:** Data appeared to be spread so that devices distal from the insertion guide were relatively more distant than proximal devices, which could be simulated by radially varying the axis of planar sinusoids.

**Other:** Data had a smaller uncorrelated appearance, which could be simulated by random translations from sinusoidal patterns.
An example of a clinical plan, for the patient with numerical identifier 8, is shown in Figure 3.1.

Figure 3.1: An example of a clinical data set. Needles were the planned lines and seeds were planned along the lines. Seed sets similar to this one were deformed to create bead sets.

### 3.3 Simulation of Implanted Devices

In addition to the clinical bead sets, simulated bead sets were used to provide tests over a wider set of conditions. It was recognized, at the outset of research, that the potential variations from the ideal seed set were infinite in scope. These variations were chosen to capture the bulk of what clinicians and researchers – speaking informally in unsupervised sessions – identified as the most likely variations. The primary intent of testing on simulated data was to demonstrate strengths and limitations of proposed
methods. Results could then be interpreted with a better improved understanding of when these methods could reasonably be expected to succeed or to fail.

Bead sets were created by taking the seed sets and deforming each strand in four distinct ways. Here, the simpler deformations are represented with a notation that is consistent with more elaborate deformations, at the risk of seeming to be more complex than strictly needed individually. The deformations were:

**Oscillation:** A needle was deformed to a strand by a sine sine wave that to mimicked local variations in the stiffness of prostate tissue.

![Figure 3.2: Strands aligned in the XZ plane along sine waves. Amplitude and wavelength of these sine waves were varied during simulation.](image)

Each strand was a sine wave in the XZ plane. The origin of the strand, $\mathbf{r}_{ij}(0)$, was simply the origin of the corresponding needle, $\mathbf{r}_{ij}(0)$, accounting for any rigid transformation between the coordinate frames of the set of strands and...
the set of needles.

For the synthetic data, the coordinate frames \( \{ \mathcal{F} \} \) and \( \{ \mathcal{I} \} \) were initially aligned so they had no translation or rotation. The rotation matrix between coordinate frames was therefore the \( 3 \times 3 \) identity matrix and the translation between the two was simply the zero vector, so

\[
\begin{pmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1
\end{pmatrix}
\]

\[
\begin{pmatrix}
0 \\
0 \\
0
\end{pmatrix}
\]

\[
\mathcal{F}_j t_j(0) = (\mathcal{F}_i R)^T \mathcal{n}_j(0) + \mathcal{F}_i T
\]

The strand was then parameterized from the local origin as a sine function with an amplitude \( \Lambda \) and a wavelength \( \lambda \):

\[
\mathcal{F}_j t_j(\mu) = \mathcal{F}_j t_j(0) + \begin{bmatrix}
\mu, 0, \Lambda \sin \left( \frac{2\pi}{\lambda}\mu \right)
\end{bmatrix}^T
\]  

(3.1)

Next, beads were placed along this strand with spacings specified by the spacing of seeds on the corresponding needle. A vector between adjacent seeds \( i \) and \( i - 1 \) on a needle \( j \), \( \mathcal{F}_j d_i \), was first transformed from \( \{ \mathcal{I} \} \) to \( \{ \mathcal{F} \} \) using an identity transformation to preserve numerical coordinates. The resulting vector \( \mathcal{F}_j d_i \) was used to define the distance between the corresponding beads. Using \( \mathcal{F}_j d_i \), the cumulative distance, \( j m_i \), between the strand \( j \)'s origin and the bead \( i \) was
calculated as

\[ \tau_{ji} = \tau_{si} - \tau_{si-1} \]
\[ \vec{\mathbf{d}}_i = (I_3) \vec{\mathbf{d}}_i + [0]_{3 \times 1} \]
\[ \kappa_j = \omega_j \]
\[ m_i = \sum_{k=2}^{i} || \vec{\mathbf{d}}_k || \]
\[ \vec{m}_j = [j m_1, j m_2, ..., j m_{\kappa_j}]^T \]

By construction, the distance \( j m_1 \) from the strand origin to the first bead was 0, meaning the first bead on the strand was at the strand origin.

The distance \( j m_i \), however, was not the straight-line distance between beads; it was the distance between the beads along the sinusoidal strand. To place beads at the right locations along a strand, the distance \( \tau_{ji}(\Gamma) \) from the origin to a point parameterized by \( \Gamma \) along a strand \( j \) was calculated as

\[ \tau_{ji}(\Gamma) = \int_{0}^{\Gamma} || \frac{d \vec{\mathbf{d}}_{ji}(\rho)}{d \rho} || \, d \rho \] (3.2)

Thus, whenever \( \tau_{ji}(\Gamma) = j m_i \) there was a bead. For \( n_j \) beads on a strand, their locations were calculated as

\[ \vec{\mathbf{b}}_i = \{ \vec{\mathbf{d}}_{ji}(\Gamma) \mid \tau_{ji}(\Gamma) \in \vec{m}_j , j \in [1, 2, ..., \eta_t] \} , i \in [1, 2, ..., \kappa_j]^T \]
\[ \bar{\mathbf{B}}_j = [\vec{\mathbf{b}}_1, \vec{\mathbf{b}}_2, ..., \vec{\mathbf{b}}_{\kappa_j}]^T \]
\[ \bar{\mathbf{B}}_0 = [\bar{\mathbf{B}}_1, \bar{\mathbf{B}}_2, ..., \bar{\mathbf{B}}_{\eta_t}]^T \]
Thus, strands were formed as sine waves in the XZ plane and the location of every simulated bead was along some such strand. The amplitude Λ and wavelength λ of the sine function could be varied as needed.

**Orientation:** Each strand was rotated about its local X axis to vary the orientation of the planes of the sine wave. This mimicked a commonly observed variation in prostate tissue stiffness.

Figure 3.3: Bead sets with two different orientations. It was possible to simulate the effects of any chosen strand rotation.

The beads on each strand were coplanar, so the orientation of the plane of the strand could be varied. Specifically, each strand plane was rotated about the local X axis. There were two ways in which strands were rotated: by a fixed angle about \( \vec{x} \), or in a radially varying pattern.

If each strand was to be rotated about the origin, the strand’s location would

\[
\begin{align*}
\theta &= 0 \\
\theta &\text{ varied for each strand, producing a radially symmetric set of strands.}
\end{align*}
\]
change (possibly be a great amount). Instead, a remote rotation was implemented: each strand was translated to the origin of the strand set, rotated, then translated back to its original strand location. For a given axis of rotation \( \vec{u} \), a rotation angle \( \theta \), and a remote rotation center \( \vec{r} \), the remote rotation was

\[
j R(\vec{r}, \vec{u}, \theta)(j \vec{b}_i) = j R(\vec{r}, \vec{u}, \theta)(j \vec{b}_i - \vec{r}_j) + \vec{r}_j
\]

(3.3)

After each strand was rotated, \( \vec{r}_j \) was the origin of strand \( j \): \( \vec{r}_j(0) \). The vector \( \vec{u} \) was simply the X axis, \([1, 0, 0] \). The angle \( \theta \) was a parameter that could be varied in simulation and was tested using two patterns:

1. **Fixed value of \( \theta \).** With a fixed value of \( \theta \), each strand of the original bead set, \( F_1B \), was rotated to form a new bead set, \( F_2B \) as

\[
F_jB = j R(\vec{r}_j, \vec{u}, \theta)(F_1B_j)
\]

(3.4)

2. **Radially varying \( \theta \).** For a radially symmetric plane pattern, \( \theta \) was the signed angle \( \in [-\pi, \pi] \) between the strand origin and the Z axis. This rotated the planes of the strands were radially symmetric, using

\[
\vec{x} = [1, 0, 0]^T, \quad \vec{y} = [0, 1, 0]^T, \quad \vec{z} = [0, 0, 1]^T
\]

\[
\vec{r}_j = \vec{r}_j(0)
\]

\[
yz\vec{r}_j = \vec{r}_j - (\vec{r}_j \cdot \vec{x})\vec{x}
\]

\[
\theta(\vec{r}_j) = \frac{yz\vec{r}_j \cdot \vec{y}}{|yz\vec{r}_j \cdot \vec{y}|} \cos^{-1} \left( \frac{yz\vec{r}_j \cdot \vec{z}}{|yz\vec{r}_j|} \right)
\]
to form a new bead set $F_{2}B$ as

$$F_{2}B_{j} = jR(\vec{r}_{j}, \bar{x}, \theta(\vec{r}_{j}))(F_{1}B_{j})$$  \hspace{1cm} (3.5)$$

Simulations included various fixed values for $\theta$, as well as a radially symmetric pattern.

**Splay:** Strands could be oriented so that they radiated outwards from the central X axis. This was intended to mimic swelling, or edema, of prostate tissue.

![Figure 3.4: Bead sets with two different splay angles $\phi$. Simulation could test the effect of any value of $\phi$.](image)

To do this the strands were angled away from the set’s X axis by an angle $\phi$. To avoid strands from being carried away from one another, the rotation was performed at the centroid $\vec{c}_{j}$ of the strand as

$$\vec{c}_{j} = \left( \frac{\sum_{i=0}^{n_j} j\vec{b}_{i}}{n_j} \right)$$  \hspace{1cm} (3.6)$$
CHAPTER 3. MATERIALS AND METHODS

The rotation axis $\vec{u}_j$ for a strand was constructed perpendicular to the X axis and the strand origin $\vec{r}_j$. Rotating about this axis angled the strand in the plane formed by the X axis and the $\vec{r}_j$ to be

$$\vec{u}_j = \frac{\vec{x} \times \vec{r}_j}{||\vec{r}_j||}$$

$$[u_j]_\times = \begin{bmatrix} 0 & -z\vec{u}_j & y\vec{u}_j \\ z\vec{u}_j & 0 & -x\vec{u}_j \\ -y\vec{u}_j & x\vec{u}_j & 0 \end{bmatrix}$$

where

$$jR(\vec{u}_j, \phi) = I_3 + (\sin(\phi)) [u_j]_\times + (1 - \cos(\phi)) [u_j]^2$$

$$\tilde{\vec{F}}_B = (jR(\vec{c}_j, \vec{a}, \phi)) \tilde{\vec{F}}_B$$

The overall appearance of the bead set was that of 'splayed" strands, closer to each other at the proximal end of the bead set and farther from each other at the distal end.

Other: Each bead was randomly translated along the three axis to capture other, less-structured effects.

To add random error to the bead set, many error vectors were constructed. Each element of a $3 \times 1$ error vector was randomly selected using a uniform distribution in the range $[-\delta, \delta]$. The bound on the random error, $\delta > 0$, was a
(a) $\delta = 1\text{mm}$

(b) $\delta = 3\text{mm}$

Figure 3.5: Bead sets with two different amounts of random error $\delta$. Simulations could test the effect of different values of $\delta$.

Because $\delta$ represented a maximum shift in one direction, the maximum radial distance that the bead could move was $\sqrt{3(\delta)^2}$.

In summary, a bead set $\mathcal{F}_4 B_j$ was deformed in a combination of four ways: with a sine wave strand shape, with different planar orientations, with different splay angles, and with random error. The parameters controlling each deformation are summarized in Table 3.2 and were varied against one another to obtain many different bead sets.
Table 3.2: Parameters of the simulated bead set. Changing any of these parameters changes patterns in the bead set; a wide variety of conditions were be used to tune and test the algorithm.

<table>
<thead>
<tr>
<th>Deformation</th>
<th>Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strand shape</td>
<td>$A$ : Amplitude of the sine wave</td>
</tr>
<tr>
<td></td>
<td>$W$ : Wavelength of the sine wave</td>
</tr>
<tr>
<td>Planar orientation</td>
<td>$\theta$ : Angle of remote rotation about $\vec{x}$ to change the orientation of the strand-plane</td>
</tr>
<tr>
<td>Splay</td>
<td>$\phi$ : Angle of remote rotation about $\vec{x} \times \vec{r}_j$ to produce a splay effect</td>
</tr>
<tr>
<td>Random translation</td>
<td>$\delta$ : Bound on random error along a single axis</td>
</tr>
</tbody>
</table>

### 3.4 Testing the Simulated Data

Simulated beads had a known correspondence (or ground truth) and an unlimited, flexible opportunity for testing. Because correspondence between seeds and beads was known, registration errors could easily be found. Registration success could then be scored as the percentage of correct assignments found by the algorithm as

$$\text{success rate} = \frac{\text{number of correct bead-seed assignments}}{\text{total number of bead-seed assignments}}$$  \hspace{1cm} (3.7)

Transformation accuracy was easy to measure because the simulated bead sets were neither translated nor rotated during their creation. If the bead set had to be rotated or translated to minimize the distance between bead-seed matches, an error must have been introduced by bead-seed mismatches, and/or by deformation of the bead set.

Translational error was defined as the distance from the centroid of the seed set to the centroid of the bead set. Both centroids remained roughly constant through the simulation process and so translational error was not a measure of interest.
Rotational error was defined as the angle of rotation carrying the bead set to the seed set. This was extracted from the compounded rotation matrix by inverting the Rodrigues rotation formula to find the associated skew-symmetric matrix

\[
\begin{bmatrix}
0 & -z\vec{u} & y\vec{u} \\
-z\vec{u} & 0 & -x\vec{u} \\
-y\vec{u} & z\vec{u} & 0
\end{bmatrix}
\]

From the skew-symmetric matrix, the Rodrigues vector \( \vec{u} \) could be assembled and the net angle \( \psi \) computed as

\[
\psi = \sin^{-1}\left(\frac{||\vec{u}||}{2}\right)
\]  

(3.8)

3.5 An Algorithm for Finding Stranded Beads

In this section, the algorithm used to match implanted seeds with their planned positions is described. This algorithm was designed to be used in a clinical setting. Because of this, we set out three goals for the success of our algorithm that were kept in mind during the entire design process:

**Accuracy.** *Ad hoc* solutions are generally inappropriate for applications involving medical treatment of human subjects. Because this algorithm may be used to assess and plan interventions, special care was taken to ensure accuracy. Algorithmically, solutions that found global maxima were used wherever possible.

**Performance.** Intraoperative applications require fast, quasi-realtime algorithms. When NP-complete problems arose, these were usually reduced to a simpler one that could be solved in polynomial time.
Independence. For speed and convenience, the amount of information needed for the algorithm was to be minimal. This way, the algorithm could more easily resolve partial datasets that in the future may occur in both intraoperative settings and postoperative analysis.

The matching-and-alignment problem was solved by breaking it down into smaller steps, then analyzing the inputs and desired output. Inputs were the spatial relationships of the planned seeds and implanted beads, plus the needle assignments; the output was a match between planned seeds and synthetic or detected beads.

Finding a matching using with spatial relationship alone could potentially lead to nonsensical needle paths, because this problem would be under-constrained. Connectivity was therefore used as an additional constraint, so that seeds could be grouped by the needle with which they were implanted. This led to a four-part algorithm:

Initial Alignment: The overall axis of the needle ensemble was estimated using linear regression on a projection of the bead set. The bead set was then rotated to align this first needle axis estimate with the X axis.

First Reconstruction and Alignment: • Partial strands were reconstructed from the aligned beads. These reconstructed strands were used to better estimate the overall needle axis. The bead set was again rotated to align this better needle axis estimate to the X axis.

• Using this well-aligned bead set, full strands were reconstructed.

Bead-Seed Registration: Full strands were matched to needles. Beads were then matched to seeds, using the strand-needle matching as a constraint on the matchings.
**Optimal Transformation:** The optimal transformation between the bead cloud and seed cloud was then found.

The process is summarized in Algorithm 3.1, represented visually in Figure 3.6 and detailed in the following subsections.

![Flowchart of the registration algorithm](image)

- **1.** Align plan and implant
- **2.** Reconstruct strands
- **3.** Match strands, then beads
- **4.** Find the optimal transformation

**Figure 3.6:** A flowchart of the registration algorithm

### 3.5.1 An Initial Alignment

The overall X axes were the average direction of the needles in the ideal frame \(\{I\}\) and of the strands in the fluoroscopic frame \(\{F\}\). The first step of the algorithm was to find this initial alignment, which was represented as a rotation about the ideal Y axis of the ideal frame to the fluoroscopic frame. The obvious method, which would be to use a global fit of seeds to a line, was impractical for prostate data because such a fit tended to be along a diagonal. Instead, because needles were exactly planned
Algorithm 3.1 Registers beads to seeds using strands as a constraint on the registration.

{1. Initial Alignment}
\[ \text{estimate}1 \leftarrow \text{estimateNeedleAxis}(\text{beads}) \]
\[ \text{alignedBeads}1 \leftarrow \text{alignWithX}(\text{beads}, \text{estimate}1) \]

{2. Strand Reconstruction} {First Reconstruction and Alignment} {The frequency of occurrence of some spacing, spacings[i], was recorded in frequencies[i]. Frequencies are in descending order.}
\[ \text{spacings}, \text{frequencies} \leftarrow \text{getSpacings}(\text{needles}) \]
\[ s \leftarrow \text{spacings}[1] \text{ } \{ \text{most common spacing} \} \]
\[ \text{freq} \leftarrow \text{frequencies}[1] \]
\[ \text{strandEstim} \leftarrow \text{strandReconstruction}(\text{alignedBeads}1, \text{alignedBeads}1, s, \text{freq}) \]
\[ \text{estimate}2 \leftarrow \text{estimateNeedleAxis}(\text{beads}, \text{strandEstim}) \]
\[ \text{alignedBeads}2 \leftarrow \text{alignWithX}(\text{alignedBeads}2, \text{estimate}2) \]

{Full-Strand Reconstruction}
\[ \text{for } i = 1 \rightarrow \text{length} (\text{spacings}) \text{ do} \]
\[ s \leftarrow \text{spacings}[i] \]
\[ \text{freq} \leftarrow \text{frequencies}[i] \]
\[ \text{strands} \leftarrow \{ \} \]
\[ \text{[beginnings}, \text{ends}] \leftarrow \text{getBeginningsAndEnds} (\text{strands}, \text{alignedBeads}2) \]
\[ \text{proximal} \leftarrow \text{ends} \]
\[ \text{distal} \leftarrow \text{beginnings} \]
\[ \text{partialStrands} \leftarrow \text{strandReconstruction}(\text{proximal}, \text{distal}, s, \text{freq}) \text{ } \{ \text{ends of partial strands assigned to others’ beginnings to make a longer strand} \} \]
\[ \text{strands} \leftarrow \text{connect}(\text{strands}, \text{partialStrands}) \]
\[ \text{end for} \]

{3. Strand-To-Needle, Bead-To-Seed Registration}
\[ \text{strandsToNeedles} \leftarrow \text{matchStrandsToNeedles}(\text{strands}, \text{needles}) \]
\[ \text{beadsToSeeds} \leftarrow \text{matchBeadsToSeeds}(\text{beads}, \text{seeds}, \text{strandsToNeedles}) \]

{4. Optimal Transformation}
\[ \text{R} \leftarrow \text{getTransformation}(\text{beads}, \text{seeds}, \text{beadsToSeeds}) \]

\[ \text{return } \{\text{strands}, \text{beadsToSeeds}, \text{R}\} \]
along the X axis – and strands presumably had a similar overall alignment – the bead
set was first projected onto a plane by using additional information.

The coordinate axes of the fluoroscope were, from the imaging setup in the pre-
procedure room, roughly aligned to the three anatomical axes: anteroposterior (AP)
corresponded with the X axis in \( \{ \mathcal{F} \} \), mediolateral (ML) with the Y axis, and super-
oinferior (SI) with the Z axis. Needles were approximately inserted along the X axis
and angled towards the +Z axis (ventrally). Although the strands covered a large
spread in Y positions, each individual strand did not appear to stray far from the XZ
plane. The bead set was accordingly projected to the XZ plane as

\[
\vec{x}_{xz} \vec{b}_i = [x_{bi}, z_{bi}]^T
\]

\[
\vec{x}_{xz} B_0 = [\vec{x}_{xz} \vec{b}_1, \vec{x}_{xz} \vec{b}_2, \ldots, \vec{x}_{xz} \vec{b}_n]
\]

A line in this plane was fit, in the least-squares sense, using an ordinary numerical
method [78]. For a given data pair, the linear model and residual errors could be
written as

\[
z_{b_i} = p_0 + p_1 x_{b_i}
\]

\[
z_{b_i} = y_{b_i} + \epsilon_i
\]
or, gathering the data into matrix form,

\[
\mathbf{F} \mathbf{z} B^T_0 = \begin{bmatrix}
  p_0 + p_1 x b_1 \\
  p_0 + p_1 x b_2 \\
  \vdots \\
  p_0 + p_1 x b_{\eta_b}
\end{bmatrix} + \begin{bmatrix}
  \epsilon_1 \\
  \epsilon_2 \\
  \vdots \\
  \epsilon_{\eta_b}
\end{bmatrix}
\]

\[
\mathbf{F} \mathbf{z} B^T = \begin{bmatrix}
  1 \\
  1 \\
  \vdots \\
  1
\end{bmatrix} \begin{bmatrix}
  \epsilon_1 \\
  \epsilon_2 \\
  \vdots \\
  \epsilon_{\eta_b}
\end{bmatrix}
\]

Abbreviating \( \mathbf{F} \mathbf{z} B^T_0 \) as \( \mathbf{Z} \) and \( [1 \ F^T_0] \bar{\mathbf{p}} \) as \( \mathbf{X} \), the sum of squared errors could be expressed as

\[
\mathbf{Z} = \mathbf{X} \bar{\mathbf{p}} + \mathbf{E}
\]

\[
\Rightarrow \mathbf{E} = \mathbf{Z} - \mathbf{X} \bar{\mathbf{p}}
\]

\[
\Rightarrow \mathbf{E}^T \mathbf{E} = (\mathbf{Z} - \mathbf{X} \bar{\mathbf{p}})^T (\mathbf{Z} - \mathbf{X} \bar{\mathbf{p}})
\]

Setting the partial derivatives to zero,

\[
\frac{\partial \mathbf{E}^2}{\partial \bar{\mathbf{p}}} = -2 \mathbf{X}^T (\mathbf{Z} - \mathbf{X} \bar{\mathbf{p}}) = 0
\]

\[
\Rightarrow \mathbf{X}^T \mathbf{X} \bar{\mathbf{p}} = \mathbf{X}^T \mathbf{Z}
\]

\[
\Rightarrow \bar{\mathbf{p}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{Z}
\]
which, in terms of the point-based matrices, was

$$
\vec{p} = ([1 \ T \ x \ B_0^T] [1 \ T \ x \ B_0^T])^{-1} ([1 \ T \ x \ B_0^T])^T (x \ B_0^T)
$$

The terms on the right-hand side were known bead coordinates, so $\vec{p}$ was easily calculated. The direction of the needle axis $F \vec{a}$ was the unit vector

$$
F \vec{a} = \frac{[1 \ p_0]^T}{||[1 \ p_0]^T||} = \frac{[1 \ p_0]^T}{\sqrt{1 + p_0^2}}
$$

The rotation of this axis to the global X axis was the matrix

$$
\theta = \cos^{-1} (F \vec{a} \cdot \vec{x})
$$

$$
R_0 = \begin{bmatrix}
\cos(\theta) & 0 & \sin(\theta) \\
0 & 1 & 0 \\
-\sin(\theta) & 0 & \cos(\theta)
\end{bmatrix}
$$

which was used to produce a transformed bead set that was roughly aligned to the fluoroscopic X axis as

$$
F \vec{B}_1 = R_0 (F \vec{B}_0)
$$

This matrix $R_0$ was recorded for later use.

### 3.5.2 Strand Reconstruction

The next step of the algorithm was to find strands, which were sets of detected beads that corresponded to ideal seeds on needles. Some previous work on strand
reconstruction assumed that strands formed linear or quadratic trajectories [53] but, in practice, the shape of strands is insufficiently consistent and regular for these previous methods to work robustly.

Here, a local non-parametric approach was used that connected one bead to another bead down a strand. This method assumed that there was some rough amount of organization to strands, which seemed justified from clinical practice. Strands were reconstructed twice: partial strands were reconstructed to find and align the true needle axis, then full strands were reconstructed from the aligned bead set.

**Partial Reconstruction and Alignment**

The basic idea was that beads on a given strand, if perfectly placed and detected, were on a line parallel to the axis $\vec{a}$ of the bead set. For a pair of beads $b_i$ and $b_j$, where $i$ is more proximal to the insertion jig than is $j$, the distance $d_{ij}$ is well defined. The angle $\phi$ between the vector from bead $i$ to bead $j$ and the axis $\vec{a}$ is also well defined, as shown in Figure 3.7.

The the angle of the variation from the planned direction could be represented
as a cost, i.e., as a distance metric. Likewise, absolute variation from the planned inter-seed distance could be represented as a second cost. If a strand was perfect then these costs would be zero, but as imperfections arose these costs would increase. The structure of the clinical data sets led to the supposition that minimizing combined costs would place beads along strands, a supposition that was successfully tested and is reported below in the chapter on Results. Calculation of the angular cost was straightforward, whereas the distance cost required a deeper appreciation of the data and their properties.

To calculate a distance cost, there must be some definite notion of the expected distance between beads. This notion could come from many sources: it might be the most common planned inter-seed distance, or might be the inter-bead distance that occurred most often, or from elsewhere. The reference length was denoted as a “tuning” parameter \( l_k \) that could be varied by the user or by another algorithm.

The distance cost, \( c_d(i, j) \), and the angular cost, \( c_\phi(i, j) \), were defined between any two beads \( b_i \) and \( b_j \) as

\[
\vec{d}_{ij} = \vec{r}_j - \vec{r}_i \\
d_{ij} = ||\vec{d}_{ij}|| \\
c_d(i, j) = |d_{ij} - l_k| \\
c_\phi(i, j) = \cos^{-1}\left( \frac{\vec{a} \cdot (\vec{d}_{ij})}{d_{ij}} \right)
\]

A comprehensive cost function \( C \), representing the cost of having seed \( j \) just distal
to seed $i$ was the simple polynomial

$$C(i, j) = \alpha_1 \cdot (c_d)^{\beta_1} + \alpha_2 \cdot (c_\phi)^{\beta_2} \quad (3.10)$$

The parameters for Equation 3.10 were manually selected for the dataset, with the values and their sensitivities described in Section C.

The assignment of $j$ as just distal to $i$ was represented as a binary assignment function: $A(i, j) = 1$ meant that this assignment was performed, and $A(i, j) = 0$ meant that the assignment was false (so some other assignment was available). The cost of placing $j$ as just distal to $i$ was simply the product

$$C(i, j)A(i, j)$$

so, summing over all possible values of $i$ and $j$,

$$C_S(A) = \sum_{i=1}^{\eta_b+\eta_r} \sum_{j=1}^{\eta_b+\eta_r} C(i, j)A(i, j)$$

The bead-stranding problem was thus reducible to the mathematical problem of finding the assignments $A$ that minimized the objective function $C_S$. The Hungarian Algorithm [41], rather than methods such as exhaustive search or linear programming methods, was selected to solve this minimization problem. This algorithm is described in detail in the Appendix material; here, crucial details of using it in a graph-theoretic sense need elucidation.

The assignment function $A$ could be thought of as a matrix and equivalently as a description of a directed graph, so $CA$ was a weighted graph where a node was a
bead and an arc in the graph was weighted as the combined cost between nodes. As described in Chapter B, the Hungarian Algorithm ensured a complete assignment: every seed had exactly one proximal and one distal connection. However, by insisting that every seed have a proximal and distal connection, the algorithm would find the best *cycles* in the graph rather than the best *strands*. Also, connections between beads had a practical constraint: for a given spacing $l_k$, only $f_k$ connections should be found amongst $n_b$ beads because the plan dictated how many seeds were supposed to have been implanted for each inter-seed distance $l_k$.

These could be resolved in many ways, with a simple and elegant one being the addition of additional nodes to capture unassigned beads. At a spacing, $l_k$, if $f_k$ connections were expected to be found in $\eta_b$ beads, then $\mu_k$ “remainder”’ beads should have been left unassigned proximally and $\mu_k$ beads should have been left unassigned distally. Some of these $\mu_k$ beads would be assigned at other spacing lengths, and others would be designated as the ends of strands. With the assignment of beads at a given spacing length, there were $f_k$ assignments that created $\mu_k$ partial strands in the bead set.

$$\mu_k = \eta_b - f_k = (\eta_t) + (\sum_{j=0}^{\eta} f_j - f_k)$$

These beads left unassigned were either ends of strands, numbering $\eta_t$, or were to be assigned later at a different spacing, numbering $\sum_{j=0}^{\eta} f_j - f_k$.

An additional $\mu_k$ nodes with fixed costs to all beads (arbitrarily set to 0) were
(a) Graph of connections between beads, beginning, and ends nodes. Only one beginning and one end are shown to retain clarity. Two beginnings and two ends are present in this graph.

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>end'r_1</th>
<th>end'r_2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>∞</td>
<td>30</td>
<td>4</td>
<td>41</td>
<td>25</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>200</td>
<td>∞</td>
<td>27</td>
<td>3</td>
<td>37</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>260</td>
<td>240</td>
<td>∞</td>
<td>43</td>
<td>8</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>570</td>
<td>265</td>
<td>220</td>
<td>∞</td>
<td>34</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>780</td>
<td>540</td>
<td>255</td>
<td>235</td>
<td>∞</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

(b) Matrix of assignment costs, $C$, between beads, beginnings and ends. Connection costs are conceptual only.

\[
A = \begin{bmatrix}
0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 \\
\end{bmatrix}
\]

(c) Matrix $A$ of optimal assignments.

\[
\vec{b}_1 \rightarrow \vec{b}_3 \\
\vec{b}_2 \rightarrow \vec{b}_1 \\
\vec{b}_3 \rightarrow \vec{b}_5
\]

$$C_\Sigma = 8 + 3 + 4 = 16$$

(d) Connections between beads and total cost.

Figure 3.8: An example of reconstructing strands in a bead set. All five spacings are 10 mm in length. The best connections in the bead set were found and represented in $A$. 
added proximally to capture the beginnings of strands and, to capture the ends of strands, $\mu_k$ nodes were added distally. Because a beginning node constituted a proximal bead with a distal connection, it was represented as an extra row of the cost matrix. Because an end constituted a distal bead with a proximal connection, it was represented as an extra column of the cost matrix. Instead of beads with high assignment costs being forced into an assignment, the beads were assigned to end nodes. Similarly, beginning nodes were able to capture seeds without a plausible proximal neighbor. This is shown in the example in Figure 3.8.

The final dimensions of the cost matrix were consequently $\eta_b + \mu_k$ by $\eta_b + \mu_k$ and takes the following form:

$$C(i, j) = \begin{bmatrix}
\infty & C(1, 2) & \ldots & C(1, \eta_b) & [0]_{1 \times (\mu_k)} \\
C(2, 1) & \infty & \ldots & C(2, \eta_b) & [0]_{1 \times (\mu_k)} \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
C(\eta_b, 1) & C(\eta_b, 2) & \ldots & C(\eta_b, \eta_b) & [0]_{1 \times (\mu_k)} \\
[0]_{(\mu_k) \times 1} & [0]_{(\mu_k) \times 1} & \ldots & [0]_{(\mu_k) \times 1} & [\infty]_{(\mu_k) \times (\mu_k)}
\end{bmatrix}$$
Beginning-of-strand nodes were prevented from being connected to end-of-strand nodes by a connection cost of $\infty$. All additional nodes were therefore assigned to beads, enforcing $\eta_t$ strands and $\eta - \eta_t - f_k$ strand gaps in the bead set. Similarly, seeds were prevented from being assigned to themselves by a connection cost of $\infty$ down the diagonal of the matrix.

This Hungarian Algorithm was then used, on this adjusted square cost matrix, to find an $A(i, j)$ that represented the optimal complete graph assignment. Using the bead set $\mathcal{F} B_1$, the most common spacing $l_0$, and its number of occurrences $f_0$, $\eta_b - f_0$ partial strands were found in the bead set.

These connections were next used to calculate a more accurate strand axis. Originally the overall strand axis was found via linear regression. With a more accurate strand axis, $c_\phi$ would more accurately reflect the true angular error of any potential connection. Thus, with a more accurate strand axis, strand reconstruction becomes more accurate. Such a local strand axis was calculated as the average of the normalized direction vectors between assigned beads:

\[
D_{i*} = [\vec{d}_{i1}, \vec{d}_{i2}, \ldots, \vec{d}_{i\eta_b}]
\]

\[
A_{i*} = [A_{i1}, A_{i2}, \ldots, A_{i\eta_b}]
\]

\[
\vec{a} = \frac{\sum_{i=0}^{n} D_{i*} A_{i*}^T}{\eta_b - \eta_r}
\]
To rotate the new $\vec{a}$ to the X axis, an angle and axis of rotation were found as

$$\psi = \cos^{-1}(\vec{a} \cdot [1, 0, 0])$$
$$\vec{u} = \vec{a} \times [1, 0, 0]$$

If $|\psi| < 90^\circ$, the dot product of $\vec{a}$ with the X axis was positive. Operating on this value, the inverse cosine produced only positive values for $\psi$. Symmetric positions of $\vec{a}$ thus resulted in the same rotation angle. The cross product, however, could distinguish between symmetric pairs; if the axis of rotation was negated in one of these symmetric pairs, a rotation in the opposite direction was calculated.

This axis and angle were used to construct a rotation matrix using the Rodrigues' rotation formula [63] to align the bead set:

$$R_1 = I_3 + (\sin(\psi))[u]_\times + (1 - \cos(\psi))[u]_\times^2$$

$$F B_2 = R_1(F B_1)$$

The matrix $R_1$ was retained for later use.

**Full Strand Reconstruction**

Knowing partial strands, it was possible to estimate full strands by iterating over different spacing lengths so that every possible connection was found. Strands were traced between these connections. This algorithm is restated in Algorithm 3.2.
Algorithm 3.2 Reconstruction of full strands for a given strand axis and spacings.

\begin{algorithm}
\begin{algorithmic}
\FOR{$i = 1 \rightarrow \text{length}(\text{spacings})$}
    \STATE{$s \leftarrow \text{spacings}[i]$}
    \STATE{$freq \leftarrow \text{occurrences}[i]$}
    \STATE{$\text{strands} \leftarrow \{\}$}
    \STATE{$[\text{beginnings}, \text{ends}] \leftarrow \text{getBeginningsAndEnds}(\text{strands}, \text{alignedBeads2})$}
    \STATE{$\text{proximal} \leftarrow \text{ends}$}
    \STATE{$\text{distal} \leftarrow \text{beginnings}$}
    \STATE{$\text{partialStrands} \leftarrow \text{strandReconstruction}(\text{proximal}, \text{distal}, s, freq)$} \COMMENT{ends of partial strands assigned to others’ beginnings to make a longer strand}
    \STATE{$\text{strands} \leftarrow \text{connect}(\text{strands}, \text{partialStrands})$}
\ENDFOR
\end{algorithmic}
\end{algorithm}

To capture bead spacings of different lengths, the algorithm for finding partial strands was performed for each bead length with values of $l_k$ and $f_k$ changed accordingly. In the previous section, the costs between beads formed a $\eta_b \times \eta_b$ cost matrix, not including additional beginning and end nodes. Each bead could therefore be assigned to a more proximal bead, or have a more distal bead assigned to it. Once a set of connections was made, however, the difference between most-proximal and most-distal beads became more relevant.

Rows of the cost matrix represented more proximal beads and columns of the cost matrix represented more distal beads. Each bead could have at most one proximal connection and one distal connection. Thus, when a bead $i$ and a more distal bead $j$ were assigned, bead $i$ could not have another distal connection, even in a later iteration of the algorithm with a different spacing. Bead $j$ was likewise prevented from being assigned to any other more-proximal beads.

In the first iteration of the algorithm, at the most common spacing $l_0$, beads had the potential to form both proximal and distal connections.
Figure 3.10: Bead 3 has been removed from the example in Figure 3.8. This graph has one spacing at 10\(mm\) and one spacing at 20\(mm\).

\[
\text{proximal}_0 \leftarrow \text{alignedBeads2} \\
\text{distal}_0 \leftarrow \text{alignedBeads2}
\]

The cost matrix was then defined between the proximal beads in rows and the distal beads in columns. Using the example graph in Figure 3.10, the first iteration of the algorithm would assign bead 2 to bead 4, leaving beads 1 and 5 as single-seed partial strands.

In the second and subsequent iterations of the algorithm, some connections have already been made because partial strands were present amongst the beads. To avoid any possibility of a bead getting two proximal or two distal connections, proximal and distal beads were reassigned. In each iteration, proximal beads were the ends of strands formed in the previous iteration: those beads without a distal connection. The seeds without a proximal connection – the beginnings of strands – became the
CHAPTER 3. MATERIALS AND METHODS

distal beads for assignment.

\[
\text{proximal}_k \leftarrow \text{ends}_{k-1}
\]

\[
\text{distal}_k \leftarrow \text{beginnings}_{k-1}
\]

The cost matrix was again formed between proximal beads – ends – and distal beads – beginnings. The best \( f_k \) connections at a given spacing \( l_k \) were then found in the bead set. The resulting connections were between ends of partial strands and beginnings of others. These two partial strands were then merged into a single strand.

In the example above, beads 1 and 5 would be connected with a spacing of 20\,mm. The proximal connections of bead 1 would be connected through bead 1 to bead 5, and indirectly to any distal connections that bead 5 might have. They now form part of the same strand.

Strand reconstruction was completed when connections at every seed spacings present in the plan were found, i.e., every partial strand was stitched into some full strand. The full strands were used for seed-to-bead matching and for shape analysis.

3.5.3 Needle-to-Strand, Seed-to-Bead Matching

The method presented so far could find strands effectively and efficiently in the bead set; these strands were then used to constrain the match between seeds and beads. To get the correct correspondence between seeds and beads, strands were first matched to needles. Next, beads on each strand were matched to seeds on the corresponding needle.
A needle in the plan was heuristically deemed likely to correspond to a strand in the implant if the two mathematical objects were (1) close to one another and (2) contained the same number of seeds. The first heuristic was useful because of a good approximate registration of strands to needles, and the second because of the accuracy of the needle reconstruction algorithm.

These heuristics could be employed in many ways, e.g., they could be used as objective in a minimization process or as constraints in a search process. Here, they were instantiated as cost functions, $c_v$ and $c_\kappa$, which were combined into a total strand cost function $C_\Omega$. The cost functions were then used to form a graph matrix that could, once again, be solved using the Hungarian Algorithm.

The distance between a strand and a needle was defined as the distance between their centroids. To calculate this distance, the needle’s centroid in frame $\{I\}$ needed to be brought into frame $\{F\}$. Because the bead set in $\{F\}$ was already well aligned to $\{I\}$ – their centroids and needle axes roughly aligned – it was assumed, for this distance measure, that no translation or rotation existed between the two frames.

\[
\bar{\mathbf{b}}_j = \sum_{i=1}^{\kappa_j} \frac{\mathbf{b}_i}{\kappa_j}
\]
\[
\bar{\mathbf{s}}_k = \sum_{i=1}^{\kappa_k} \frac{\mathbf{s}_i}{\kappa_k}
\]
\[
\bar{\mathbf{s}}_k = I_3 \bar{\mathbf{s}}_k + [0]_{3 \times 1}
\]

so the distance cost function was simply the distance

\[
\bar{\mathbf{v}}_{jk} = \bar{\mathbf{b}}_j - \bar{\mathbf{s}}_k
\]
\[
c_v(j, k) = ||\bar{\mathbf{v}}_{jk}||
\]
Next, a correspondence was deemed less likely if the numbers of seeds and beads was vastly different. If $w_k$ was the number of seeds on needle $k$, and $z_j$ was the number of beads on strand $j$, then the cost resulting from differences in the numbers was $c_{x}(j, k) = |z_j - w_k|.$

The total cost was defined as the sum of the two cost functions, so

$$C_{\Omega}(j, k) = c_{\phi}(j, k) + c_{x}(j, k)$$

This cost function was used to produce a cost graph between needles and strands, which could not be searched exhaustively because there were too many needles in a typical set of data (18-30). Rather than using a local search method, the needle-strand correspondence problem was solved using a further application of the Hungarian Algorithm to find a matching of strands to needles.

In the final step of the sub-process, seeds were matched to beads down the length of the strand-needle pairs. For a strand $j$, matched to a needle $k$, a cost function was defined using distance between seeds and beads only to be

$$c_{\sim}(h, i) = ||\vec{F}_j \vec{b}_h - \vec{F}_k \vec{s}_i||$$

so that seed-to-bead correspondence could be computed. For problems of this size (2-8 beads per strand), exhaustive search was certainly possible; in this instance, the Hungarian Algorithm was again used although the performance improvement was at best negligible.

By matching needles to strands first, the seed matching problem was reduced from
an $n^3$ matching problem to a $(n/6)^3 + (n/6) \cdot (6)^3$ matching problem, for an average needle length of 6 seeds. For the large values of $n$ in the data considered in this research, this reduced the complexity of the restated problem to $1/216$ of that of the original problem.

### 3.5.4 Finding the Optimal Transformation

Once a match was found, the optimal transformation between seeds and beads could be calculated. Because the previous algorithms provided an exact correspondence between data in two sets, any of a variety of paired-point registration methods could be used. In the case of seeds mapped to beads, where the deformation in the map is non-zero but small relative to the diameter of the set of seeds, a rigid registration can be used.

It has been shown, in many different ways, that rigid paired-point registration can be reduced to finding a least-squares rotation of one zero-mean data set to another zero-mean data set. Creating zero-mean data is easily done for each set, by subtracting the arithmetic mean of all the vectors from each vector. The problem is then one of finding an optimal spherical displacement of one set to the other set.

A common representation of a set of 3D vectors is as a matrix, with each column being the position of a point. Here, if there were $m$ beads/seeds matched, the zero-mean set of seeds was written as the $3 \times m$ matrix $Q$. The matching beads were placed in a matrix in order corresponding to the seeds, written as the $3 \times m$ matrix
\[ P. \]

\[ \bar{q} = \frac{\sum_{k=1}^{m} I_{\vec{s}_{i_k}}}{m} \]

\[ \bar{p} = \frac{\sum_{k=1}^{m} F_{\vec{b}_{j_k}}}{m} \]

\[ I_\mathcal{Q} = [I_{\vec{s}_{i_1}}, I_{\vec{s}_{i_2}}, \ldots, I_{\vec{s}_{i_m}}] - \bar{q} \]

\[ \mathcal{F}P = [\mathcal{F}\vec{b}_{j_1}, \mathcal{F}\vec{b}_{j_2}, \ldots, \mathcal{F}\vec{b}_{j_m}] - \bar{p} \]

\( \vec{i} \) were the indices of the seeds matched to the beads represented by indices \( \vec{j} \). Thus, seed \( \vec{i}_k \) was matched to bead \( \vec{i}_k \).

The optimal transformation was the \( 3 \times 3 \) matrix \( R \) of Equation 2.1, solved using Sibson’s method previously described in Section 2.3.3.

Combining this optimal rotation with the previous rotations found, the optimally aligned bead set, \( \mathcal{I}B_2 \), becomes

\[ \mathcal{I}B_2 = R_2(R_1R_0 \mathcal{F}B_0 - \mathcal{F}\bar{p}) + \bar{q} \]

Transforming the bead set in this way aligns the beads optimally with their corresponding seeds, in the ideal coordinate frame, \( \{\mathcal{I}\} \). From this, discrepancies between the plan and the implant can be measured.

### 3.5.5 Errors Introduced by Automated Image Processing

Because of its potential applications in assessing radiation dose and guiding invasive procedures, strand reconstruction must be accurate. Here, the plan was taken to be perfect data: straight needles with fixed spacing between seeds, no seeds missing, no
extra seeds present, and no seeds misplaced from their planned locations.

However, as applied on a patient and deduced from post-procedural medical imaging, there might have been discrepancies between the planned seeds and the beads, which were detections of the implants. For example, the physician might have omitted seeds, added seeds, or occasionally placed seeds in different areas of the prostate than either the pre-operative plans or surgical notes reported. Reconstruction of the bead set from fluoroscopy could also have caused missing, extra or spurious beads (a notorious source being from back-projection errors from two X-ray images).

These missing, extra, and spurious seeds needed to be handled by the strand reconstruction and IF-RF registration algorithms. After considering the challenges these errors pose and how they might be overcome manually, a modification to the previous algorithm will be presented. This final algorithm was an attempt to automate some of this manual error compensation.

Missing Beads

A bead may be missing from the beginning, end, or middle of a strand. If a bead is missing from the beginning or end of a strand, a shorter strand results. No special modification to the algorithm was needed here: the number of strands remained the same, as did the remaining relationships between beads.

A missing bead produced a more complicated situation when it was missing from the middle of a strand, forming a gap. There were two ways to manually correct the problem. One was to increase the number of strands in the bead set, which reconstructed one strand as two strands that were separated by the gap where the missing bead would have been. The second way was to simply add an additional "odd
spacing” at twice the most common spacing distance; by doing this, the gap was filled and the beads on either end of the missing bead were appropriately connected.

**Extra Beads**

Additional beads are a problem for any algorithm, with two reasonable solutions: either the number of strands is constant or the number of strands increases. If the number of strands remains the same, then additional beads will have to be added to existing strands; this may produce incorrect connections of beads, such as changing the assignment of another bead to a distinct strand. Otherwise, if the number of strands is increased (e.g., by adding one strand) then one or more additional beads can be placed on the new strand own.

If the deformations of needle tracts are not large relative to the diameter of the seed set, then it is possible to detect gross mis-assignments of beads to strands. This is most evident when a “zig-zag” strand shape appears because such a shape is physiologically implausible. Although it is possible to manually check each dataset and adjust the number of needles accordingly, such a process takes time and is prone to error. Automation of the shape recognition is possible, such as by fitting a cubic spline to each strand and testing for high curvatures, but it is far from clear how to adjust such an algorithm for all reasonable clinical datasets that might be processed.

Instead, the number of strands can be used as a search parameter in the bead-seed assignment process. The idea is straightforward: for a given number of strands, the overall quality of the bead/seed assignment can be measured by the connection cost; if a given number of strands has a lower cost than another number of strands, the number with the lowest overall cost was selected as the better assignment.
Here, the overall cost was chosen to be the greatest connection cost (GCC) of the set. It is possible to use other measures, such as the mean cost, but the maximum cost proved useful in a small empirical trial of actual implants in prostate-cancer patients. Also, an exhaustive search was used because the number of strands was small and the algorithm was sufficiently fast to make such a search feasible. The algorithm will be presented in detail after considering the third and final major source of input errors.

**Spurious Beads**

Spurious beads, which are defined here as beads in unexpected places, have aspects of both missing and extra beads: they are missing from where they should be, yet appear elsewhere. An elegant solution to a spurious bead is to treat it them as one missing bead plus one extra bead. By correcting for a missing bead, either a strand can be added or an extra spacing can be introduced. By correcting for an extra bead, a strand can be added.

In doing so either two strands are added, or one strand and one odd spacing are added, to capture both the extra bead on its own strand and the possible gap between beads.

**A Heuristic Algorithm for Managing Input Errors**

After adding strands and spaces, the seed/bead assignment can be repeated by varying the number of strands from 1 to some maximum value $\text{numStrands}$. For a given number of strands, each connection between beads has an assignment cost associated with it, defined by Equation 3.10. When the connections have been found by the
Hungarian Algorithm, these assignment costs become connection costs. The connections with the smallest costs are, by design, the most likely to be correct. Conversely, the connections with the largest costs are more likely to be incorrect. By looking at the greatest connection cost (GCC) of the strand set, we get a measure of the worst connection in that strand set.

If the GCC of a strand set with an incorrect connection is always higher than the GCC of a strand set with only correct connections, it is prudent to make a decision based on a decision threshold. The number of strands can then be adjusted to be the smallest number of strands with a GCC that is below the decision threshold. An algorithm for finding the true number of strands, and their configuration, is shown in Algorithm 3.3.
Algorithm 3.3 An algorithm for finding the number of strands in a bead set. Instead of iterating over numStrands, a search algorithm could be employed.

\[
\text{outputStrands} \leftarrow -1
\]

\[
\text{while outputStrands} == -1 \text{ do}
\]

\[
\text{[strands, connectionCosts]} \leftarrow \text{reconstructStrands(beads, numStrands)}
\]

\[
\text{GCC} \leftarrow \max(\text{connectionCosts})
\]

\{If GCC has crossed the threshold, the appropriate output value is assigned\}

\[
\text{if } \text{GCC} \geq \text{threshold AND prevGCC} < \text{threshold then}
\]

\[
\text{outputStrands} \leftarrow \text{prevStrands}
\]

\text{else if } \text{GCC} \leq \text{threshold AND prevGCC} > \text{threshold then}

\[
\text{outputStrands} \leftarrow \text{strands}
\]

\text{end if}

\{Change the number of strands\}

\[
\text{if } \text{GCC} \geq \text{threshold then}
\]

\[
\text{numStrands} \leftarrow \text{numStrands} + 1
\]

\text{else}

\[
\text{numStrands} \leftarrow \text{numStrands} - 1
\]

\text{end if}

\[
\text{prevGCC} \leftarrow \text{GCC}
\]

\[
\text{prevStrands} \leftarrow \text{strands}
\]

\text{end while}

\text{return } \text{outputStrands}
Chapter 4

Results

Registration of beads to seeds was tested on both simulated and clinical data, with the former derived from clinical data by introducing variations with clinical relevance. Because there were 5 constructed ways to vary the data, and a 5-parameter space was too large to fully explore in the time available for conducting this research, instead a carefully selected subset of pairwise comparisons was conducted.

4.1 Evaluation Process

Recalling the variational parameters, they were:

**Amplitude** $\Lambda$: nominal amplitude of a planar sinusoid

**Wavelength** $\lambda$: spatial wavelength of a planar sinusoid

**Plane Angle** $\theta$: orientation of all sinusoid planes from the ideal orientation

**Splay Angle** $\phi$: angle of each nominal sinusoid axis from the ideal axis

**Translation** $\delta$: displacement of each bead from its nominal location
These parameters were used to modify a seed set – which was the ideal plan – to a simulated baseline bead set. This baseline case was chosen based on rough estimates of properties of the clinical implants. In this baseline case, the beads could be registered to the seeds with 100% accuracy. Table 4.1 provides numerical values of these parameters and Figure 4.1 provides a visual comparison of the baseline bead set to an actual clinical bead set. The baseline values were:

Table 4.1: The value of parameters chosen for the baseline case.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amplitude</td>
<td>Λ: 2 mm</td>
</tr>
<tr>
<td>Wavelength</td>
<td>λ: 60 mm</td>
</tr>
<tr>
<td>Plane Angle</td>
<td>θ: 0°</td>
</tr>
<tr>
<td>Splay Angle</td>
<td>φ: 5°</td>
</tr>
<tr>
<td>Translation</td>
<td>δ: 0 mm</td>
</tr>
</tbody>
</table>

Figure 4.1: A simulated (a) and clinical (b) bead set. The simulated base case was intended to mimic the clinical bead set.

Of the 10 possible pairs of parameters, 4 were selected as reflective of observed
patterns in the clinical data:

<table>
<thead>
<tr>
<th>Pair</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\lambda \leftrightarrow \Lambda$</td>
<td>Sinusoid wavelength vs. amplitude</td>
</tr>
<tr>
<td>$\delta \leftrightarrow \Lambda$</td>
<td>Translation vs. amplitude</td>
</tr>
<tr>
<td>$\delta \leftrightarrow \phi$</td>
<td>Translation vs. splay</td>
</tr>
<tr>
<td>$\theta \leftrightarrow \phi$</td>
<td>Orientation vs. splay</td>
</tr>
</tbody>
</table>

Other pairs or groups of parameters could be investigated for a more thorough characterization.

Further variations from this baseline were then registered and analyzed as

**Accuracy Total:** number of beads correctly assigned;

**Accuracy Deviation:** standard deviation of the number of correct assignments;

**Rotational Error:** difference between the registered and true overall bead direction;

**Rotational Deviation:** standard deviation of the rotational error.

Significance of error was analyzed using Student’s paired 2-tailed t-test. It should be noted that this is bivariate analysis of multivariate data: it is expected that 5% of the statistically significant results below are due to chance alone. However, individual statistical differences are less important in this case than broader statistical trends.

### 4.2 Simulation – Wavelength $\lambda$ and Amplitude $\Lambda$

Wavelength $\lambda$ was varied from 30\text{mm} to 150\text{mm} by steps of 30\text{mm} while amplitude $\Lambda$ was varied from 0\text{mm} to 7\text{mm} by steps of 1\text{mm}. All other parameters were held
at baseline values. Figure 4.2 shows examples of the bead sets with their strands reconstructed, and Figure 4.3 plots the registration accuracies.

Figure 4.2: A bead set with long wavelength and small amplitude strands (a), and a bead set with short wavelength and large amplitude strands (b).

4.2.1 Registration Accuracy

Accuracy decreased and variability increased when the wavelength of strand deformation was short and the amplitude was large. If amplitude was small, then a short wavelength had little effect on accuracy. Similarly, if the wavelength was long, a large amplitude had little effect on accuracy.

Significance of changing wavelength was assessed by using a two-tailed paired t-test. Wavelengths from 90 mm to 150 mm showed no significant differences them. There was a significant difference between $\lambda = 30 mm$ and all other wavelengths
(a) Registration accuracy over a range of wavelengths ($30mm < W < 150mm$) and amplitudes ($0mm < A < 7mm$).

(b) Standard deviation of the sample for the registration accuracies in (a).

Figure 4.3: Registration accuracy results over wavelengths and amplitudes. Three simulated bead sets were used to calculate each data point.
(\(p < 0.005\)), between \(\lambda = 60\text{mm}\) and \(\lambda = 120\text{mm}\) \((p < 0.05\)), and between \(\lambda = 60\text{mm}\) and \(\lambda = 150\text{mm}\) \((p < 0.05\)).

Amplitudes were similarly assessed. Amplitudes from 0\(\text{mm}\) to 5\(\text{mm}\) were not significantly different. There was a difference between \(\Lambda = 6\text{mm}\) and \(\Lambda \leq 5\text{mm}\) \((p < 0.05\)), and between \(\Lambda = 7\text{mm}\) and \(\Lambda \leq 5\text{mm}\) \((p < 0.05\)). The difference between \(\Lambda = 6\text{mm}\) and \(\Lambda = 7\text{mm}\) was just below significance \((p < 0.05)\).

It was inferred that, to maintain a registration accuracy above 90\%, the algorithm should be restricted to cases where the wavelength was at least 60\(\text{mm}\) or the amplitude was below 4\(\text{mm}\). Either condition sufficed to keep registration accuracy high.

### 4.2.2 Rotational Error

Effects of amplitude and wavelength on rotational error are reported in Figure 4.4. Being smaller and less variable, rotational error at \(\lambda = 150\text{mm}\) was significantly different from that at all other wavelengths \((p < 0.05)\). Similarly, \(\lambda = 120\text{mm}\) was significantly different than \(\lambda = 60\text{mm}\) and \(\lambda = 90\text{mm}\) \((p < 0.005)\), but not \(\lambda = 30\text{mm}\) \((p > 0.06)\).

Unlike wavelength, amplitude nearly always produced a significant results from one measurement to the next \((p < 0.05)\). The only two insignificant transitions were from \(\Lambda = 4\) to \(\Lambda = 5\) \((p > 0.06)\) and from \(\Lambda = 6\) to \(\Lambda = 7\) \((p > 0.05)\).

Inconsistent and variable changes over wavelength suggested that wavelength was not a consistent predictor of rotational error. Rotational accuracy appeared to be more consistently sensitive to changes in amplitude than to changes in wavelength.
Figure 4.4: Rotational error created by changes in amplitude and wavelength. Three simulated bead sets were used to calculate each data point.
4.3 Simulation – Translation $\delta$ and Amplitude $\Lambda$

The bound on random translation, $\delta$, was varied from 0\text{mm} to 3\text{mm} in steps of 0.5\text{mm} while amplitude was varied from 1\text{mm} to 7\text{mm} in steps of 1\text{mm}. Because of the random translation, each of the three seed sets was used to generate five bead sets. The 15 resulting bead sets were used to generate accuracy and standard deviation results, as shown in Figure 4.5.

The trend in the figure is clear: as amplitude increased and as random translation increased, registration accuracy decreased with variability generally increasing in the same pattern. Above an amplitude of 4\text{mm}, every amplitude was significantly different from every other ($p < 0.05$). Every additional 0.5\text{mm} in randomness produced a significant change ($p < 0.005$).

To achieve a 90% accuracy rate, amplitude should stay $\Lambda \leq 4\text{mm}$ and the bound on random translation should stay $\delta \leq 1.5\text{mm}$. If random translation was $\delta \leq 1.0\text{mm}$, amplitude could increase to as much as 6\text{mm} without accuracy dropping below 90%. This leeway afforded to amplitude is lacking for random translation. No matter the amplitude, a random-translation bound above 1.5\text{mm} produced a registration accuracy below 90%.
(a) Registration accuracy over a range of random translations ($0 \text{mm} < \delta < 3 \text{mm}$) and amplitudes ($0 \text{mm} < \Lambda < 7 \text{mm}$).

(b) Standard deviation of the sample for the registration accuracies in (a).

Figure 4.5: Registration accuracy results for random translations and amplitudes. Three simulated bead sets were used to calculate each data point.
4.4 Simulation – Translation $\delta$ and Splay $\phi$

The bound on random translation, $\delta$, was varied from 0 mm to 3 mm in steps of 0.5 mm while splay angle was varied from 0° to 30° in steps of 5°. Because of the random translation, each of the three seed sets was used to generate five bead sets. The 15 resulting bead sets were used to generate accuracy and standard deviation results. These results are shown in Figure 4.6.

A similar pattern to Figure 4.5 can be observed in Figure 4.6. As randomness increased and splay angle increased, accuracy decreased. In fact, every splay angle increase produces a significant change in accuracy ($p < 0.05$). Similarly, nearly every increase in the random-translation bound produced a highly significant change ($p < 0.005$). The two transitions that remained insignificant were from $\delta = 0 mm$ to $\delta = 0.5 mm$ ($p > 0.1$) and from $\delta = 1.0 mm$ to $\delta = 1.5 mm$ ($p > 0.05$).

The results suggested that, to ensure the registration accuracy remains above 90%, the random-translation bound should remain under 1.5 mm while the splay angle remains under 15°. Once again, if one of these is decreased, the other could increase while maintaining better than 90% accuracy.

4.5 Simulation – Plane Angle $\theta$ and Splay $\phi$

Planar orientation was varied from 0° to 180° in steps of 20° while splay angle was varied from 0° to 30° in steps of 5°. Figure 4.7 shows Examples of bead sets for representative variations from the base case.
(a) Registration accuracy over a range of randomness bounds ($0 \text{mm} < \delta < 3 \text{mm}$) and splay angles ($0^\circ < \phi < 30^\circ$).

(b) Standard deviation of the sample for the registration accuracies in (a).

Figure 4.6: Registration accuracy results for random translations and splay angles. Three simulated bead sets were used to calculate each data point.
Figure 4.7: A bead set with a radially symmetric planar orientation and a large splay angle (a), and a bead set with a fixed planar orientation and a small splay angle (b).
4.5.1 Registration Accuracy

Figure 4.8 shows registration accuracy, a cross-section of which is given in Figure 4.9. On visual inspection, accuracy appeared to decrease with larger splay angles. Variability also increased with splay angle. Planar orientation did not appear to have an effect on accuracy.

Statistical analysis confirmed this interpretation. A two-tailed paired t-test found no significant difference between any given planar orientation and any other orientation ($p > 0.05$). Rotationally symmetric planar orientation, shown in Figure 4.9, was not significantly different than any fixed planar orientation ($p > 0.1$).

Splay angles $\phi \leq 15^\circ$ showed no significant difference between them: every dataset was reconstructed with 100% accuracy. There was a significant difference between $\phi = 20^\circ$ and all values $\phi \leq 15^\circ$ ($p < 0.01$), between $\phi = 25^\circ$ and all values $\phi \leq 20^\circ$ ($p < 0.0001$), and between $\phi = 30^\circ$ and all values $\phi \leq 25^\circ$ ($p < 0.00001$).

The registration accuracy remained above 90% in every case where the splay angle was $\phi \leq 25^\circ$.

Rotational Error

Unlike registration accuracy, rotational error appeared to vary with the planar orientation of the strand, as Figure 4.10 shows.

As expected, rotational error appeared to increase as splay angle increased. A significant difference was reached between $\phi = 0^\circ$ and $\phi = 5^\circ$ ($p < 0.01$) and between $\phi = 0^\circ$ and $\phi = 10^\circ$ ($p = 0.028$). At larger splay angles, $\phi = 25^\circ$ was significantly different from splay angles that had $\phi = 10^\circ$ to $\phi = 20^\circ$ ($p < 0.01$). Splay $\phi = 30^\circ$ was significantly different from all other splay angles ($p < 0.05$).
(a) Registration accuracy over a range of planar orientations ($0^\circ < \theta < 160^\circ$) and splay angles ($0^\circ < \phi < 30^\circ$).

(b) Standard deviation of the sample for the registration accuracies in (a).

Figure 4.8: Registration accuracy results over planar orientations and splay angles. Three simulated bead sets were used to calculate each data point.
The shape of this graph is quite clear. Planar orientation appeared to affect accuracy directly at small splay angles. This effect did not appear at large splay angles, but planar orientation did seem to affect the variance of the sample at higher splay angles.

4.6 Results with Clinical Implants

The registration accuracy of the algorithm was evaluated both manually and statistically for nine clinical bead sets. The parameter set was selected manually to maximize accuracy. These parameters are reported in Table 4.2. With these parameters, the algorithm was 100% accurate for all nine clinical datasets. Figure 4.11 shows an example of the strands (interpolated by splines for better illustration) and the bead-seed
(a) Absolute rotational error over splay angles and planar orientations

(b) Standard deviation of the sample for the registration accuracies in (a).

Figure 4.10: Rotational error created by changes in splay angle and planar orientation. Three simulated bead sets were used to calculate each data point.
\[ \alpha_1 = 0.0521 \quad \text{Weight for distance cost function} \]
\[ \beta_1 = 1.5 \quad \text{Exponent for distance cost function} \]
\[ \alpha_2 = 0.9479 \cdot s \quad \text{Weight for angular cost function} \]
\[ \beta_2 = 4 \quad \text{Exponent for angular cost function} \]

Table 4.2: Strand reconstruction parameters used for simulations and clinical results. 

\( s \) is the spacing (in cm) currently being sought in the bead set.

matches for a representative data from a patient.

The algorithm reliably terminated in under 5.9 seconds (mean 3.49s) with a 2.67GHz processor and 6GB of RAM. These results suggest that the algorithm can be both effective and efficient enough for clinical application.

As reported in the Appendix, selection of the cost weighting parameters did not appear to be highly sensitive. However, testing on a larger dataset would be prudent before applying this registration method in a large clinical study.
(a) Needle-strand correspondence for patient 1. No errors were found.

(b) Strands only, interpolated by splines.

c) Seed-bead correspondence.

Figure 4.11: The charts of a perfect matching: the needle-strand correspondence (a), the strands only (b), and the seed-bead correspondence (c) for patient 1.
Chapter 5

Discussion and Conclusions

The primary goal of this work was to register beads to seeds using strands as constraints, with a secondary goal of characterizing the registration algorithm’s performance on representative data.

The results of the algorithm on simulated data will be discussed, followed by an examination of parametric bounds for a well-behaved strand. Next, the performance of the algorithm on clinical data will be discussed. Clinical results included the accuracy of the algorithm for clinical data and whether the algorithm could correctly predict the number of strands in the dataset. The percentage of clinical strands that were well-behaved will be briefly considered.

5.1 Results Using Simulated Data

In simulation, datasets consisting of strands with particular wavelengths, amplitudes, planar orientations, splay angles, and random bounds were tested. Some of these
parameters resulted in more accurate registrations than others. A successful registration was defined as one that captured \(\geq 90\%\) of correct seed-bead matchings. A well-behaved strand was defined as one with parameters that produced a successful registration. In general, one could expect that if a clinical strand is well-behaved, then it would be more likely to be correctly detected in the dataset. It is thus useful to determine some parametric bounds on well-behaved strands.

Table 5.1: Some constraints on strand parameters that produced an accuracy > 90%.

<table>
<thead>
<tr>
<th>Constraints</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wavelength and Amplitude</td>
<td>(\lambda \geq 60\ mm) and (\Lambda \leq 4\ mm)</td>
</tr>
<tr>
<td>Amplitude and Random Translation</td>
<td>(\Lambda \leq 4\ mm) and (\delta \leq 1.5\ mm)</td>
</tr>
<tr>
<td>Random Translation and Splay</td>
<td>(\delta \leq 1.5\ mm) and (\phi \leq 15^\circ)</td>
</tr>
<tr>
<td>Splay and Planar Orientation</td>
<td>(\phi \leq 25^\circ) and (\theta) independent</td>
</tr>
</tbody>
</table>

Table 5.2: Parametric bound for well-behaved strands. Bounds for individual parameters were consolidated from the results in Table 5.1.

<table>
<thead>
<tr>
<th>Constraints</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wavelength</td>
<td>(\Lambda \geq 60\ mm)</td>
</tr>
<tr>
<td>Amplitude</td>
<td>(\lambda \leq 4\ mm)</td>
</tr>
<tr>
<td>Planar Orientation</td>
<td>(\theta) independent</td>
</tr>
<tr>
<td>Splay Angle</td>
<td>(\phi \leq 15^\circ)</td>
</tr>
<tr>
<td>Random bound</td>
<td>(\delta \leq 1.5\ mm)</td>
</tr>
</tbody>
</table>

In each simulation, some parameters were varied against others in a total of 4 distinct ways. The conditions on these parameters required to obtain above 90% accuracy are summarized in Table 5.1. Interestingly, the bounds on amplitude and randomness remained constant across tests, at 4\(\ mm\) and 1.5\(\ mm\) respectively. The bound on splay angle, however, differed from test to test: when varied against planar orientation, the maximum splay angle that achieved 90% accuracy was 25\(^\circ\), when varied against the random bound, that same angle was 15\(^\circ\).
Table 5.2 consolidates the bounds on each parameter. A well-behaved strand should conform to these parametric bounds. In the future, it could be expected that clinical datasets with a high percentage of well-behaved strands may result in more accurate registrations. However, it should be noted that each individual simulation tested a uniform strand set, so data with strands that varied greatly in their parameters were not considered.

5.2 Results with Clinical Data

The heuristic parameters for the algorithm, listed in the Appendix material as Table C.1, were selected manually to maximize accuracy. A feasibility study for clinical use was conducted on nine clinical datasets. Because the algorithm was 100% accurate for all nine clinical datasets and terminated in under 5.9s, the algorithm appeared to be accurate enough and fast enough for use in a clinical setting.

The clinical data collected, however, was from just one clinical center and one clinician at that center. Although these heuristic parameters produced effective registrations on this dataset, different clinical implants may have different alignment or strand patterns. These heuristic parameters should be carefully evaluated prior to actual clinical use.

5.2.1 A Heuristic Algorithm for Input Errors

When the greatest connection cost of a reconstruction with incorrect connections was compared to those with only correct connections, it was found that incorrect connections produced noticeably higher costs. With incorrect connections, the datasets
has greatest connection costs of 36.06 or greater. With only correct connections, the greatest connection costs of the datasets were 29.16 or lower. Thus, the average of these two numbers, 32.62, was chosen as the bound on greatest connection costs: below this value, datasets were assumed to have correct connections; above this value, datasets were assumed to likely have at least one incorrect connection.

Using this threshold value, the algorithm was able to determine the correct number of strands for each dataset, without user input. The specific threshold may need to be adjusted for different or larger datasets.

5.2.2 Strand Attributes

Examining the strands with wavelengths below $10^{2.5} \approx 316$, we found strong matches that had a range of wavelengths from 18-160mm; about 8% of the strands had a wavelength below 4mm, whereas only 37% of strands had a wavelength above 60mm. Overall, 33% conformed to the conditions on well-behaved strands. For strands with a wavelength below 60mm, 97% had an amplitude below 2mm. Thus, although the wavelength was shorter, the amplitude was also smaller so the strands were reconstructed well by the algorithm. Including those strands with short wavelengths ($\leq 60mm$) and small amplitudes ($\leq 2mm$) in the definition of well-behaved, 97% of strands could be defined as well-behaved. For a dataset with similar wavelengths and amplitudes, it is reasonable to suppose that the registration accuracy of the algorithm should remain high.
CHAPTER 5. DISCUSSION AND CONCLUSIONS

5.3 Contributions

This thesis addressed four main objectives within the problem of seed-bead registration.

5.3.1 Development of a brachytherapy registration algorithm that used strands as a constraint

To effectively register seeds to beads, the algorithm proceeded in four steps:

1. Aligning the bead set
2. Reconstructing strands
3. Matching needles-to-strands, seeds-to-beads
4. Finding the optimal transformation

As has been described, this algorithm performed well on both simulated and limited clinical data.

5.3.2 Discerning, via simulation, how some variations in implanted seed sets might affect registration accuracy

Simulations covered a wide range of conditions: wavelength, amplitude, planar orientation, splay, and random error were all varied. A summation of the findings is:

Wavelength and amplitude. Simulations suggested that wavelength should be below 60mm, or amplitude below 4mm, to obtain over 90% registration accuracy.
Planar orientation. Registration accuracy did not appear to be affected by planar orientation.

Splay. Registration accuracy was maintained above 90\% for a splay angle less than 25°.

Random translation. For a splay angle less than 15°, random error less than 1.5mm maintained a registration accuracy above 90\%.

These conditions are expected to apply to by needles in clinical datasets, although variation of needles within a dataset was not studied in this work.

5.3.3 Demonstration of an algorithm that could successfully register implanted seed sets to plans

On nine clinical datasets, registration accuracy was perfect at 100\%. The algorithm terminated in under 5.9s with a mean of 3.49s with a 2.67GHz processor and 6GB of RAM.

5.3.4 Determination of relevant quantitative properties of implant sets

Strands were found to be roughly planar, with a sine wave providing a better fit than a quadratic equation, but not as good a fit as a cubic. Nonetheless, for these purposes, the RMS error a sine wave was 0.174mm ± 0.170mm which was quite sufficient.

The distribution of wavelengths and amplitudes was ascertained from the sine fit. About 92\% of the strands were found to have an amplitude below 2mm, well within the capable range of the algorithm.
Of high clinical relevance is that the peripheral zone of the prostate was found to contain the shortest wavelengths, whereas the rest of the prostate contained larger amplitudes. The physical region that might pose a problem for the algorithm in the future is the interface between the two zones, which may contain strands with shorter wavelengths and larger amplitudes than this algorithm could robustly manage.

5.4 Future Work

A number of directions can be envisioned from the foundations laid in this project.

5.4.1 Extending the scope of the algorithm to loose seeds

The algorithm, as proposed, was targeted for stranded seeds. In clinical use it is possible to implant “loose” seeds, which are not interconnected. In such cases, inter-seed distances vary substantially. Even with parameter changes, a successful registration might not be consistently found for loose seeds. Reconstructing pairs or triplets of beads could be attempted, as opposed to registering longer strands. Other techniques, including dose-to-dose registration, could also be performed alone or combined with the strand-based algorithm.

5.4.2 Creating an anatomical seed misplacement model

A full characterization of misplacement would inform future development of techniques. A vector field of seed misplacement could be created to describe misplacement of individual seeds in the prostate. A caution is that the algorithm was not sensitive to global translations or rotations of the bead set with respect to the seed
set: only the optimal overall transformation was found. Thus, vector models created with direct bead-seed registration may not be accurate. One alternative, when using fluoroscopy-ultrasound registration, is to define the anatomy of the prostate around the bead set to assist in registering the anatomy to the plan. With such an anatomical registration it might be possible to determine whether any global translations were present, and then to define a vector field that would characterize misplacement as local translation.

5.4.3 Using strands for ultrasound-fluoroscopy registration

Ultrasound-fluoroscopy registration is currently performed by using additional fiducial markers that are visible in both ultrasound and fluoroscopy. Intrinsic registration, using only beads and strands in each imaging modality, eliminates the need for these fiducials. Needle tracts are readily visible in ultrasound, whereas the implanted seeds are not. Beads are readily visible in fluoroscopy, whereas needle tracts are not. In the course of bead-seed registration, the algorithm considered here reconstructed strands that followed the path of these needle tracts. By registering such reconstructed needle tracts in fluoroscopy to those readily visible in ultrasound, it might be possible to produce an accurate and robust intrinsic registration accomplished.

5.5 Conclusion

I have contributed an efficient and effective method for registering implanted seeds to planned seeds using strands as a constraint. This algorithm has been well characterized in simulation, as well as tested on clinical data. Furthermore, I have modeled
and demonstrated a simple characterization of the reconstructed strands. I hope that this method will prove useful for characterizing and eventually reducing misplacement and, as a result, complications in prostate brachytherapy for a prevalent form of cancer in men.
Bibliography


[63] O. Rodrigues. Des lois générales qui gissent les déplacements d’un système solide dans l’espace, et de la variation des coordonnées provenant de ces
d placements consid rs ind pendamment des causes qui peuvent les produire. *Journal de Math matiques Pures et Appliqu es*, 5:380, 1840.


deferred treatment or watchful waiting among men with prostate cancer in a


[70] K.M. Snyder, R.G. Stock, S.M. Hong, Y.C. Lo, and N.N. Stone. Defining the
risk of developing grade 2 proctitis following 125i prostate brachytherapy using
a rectal dose-volume histogram analysis. *International Journal of Radiation
Oncology* *Biology* *Physics*, 50(2):335–341, 2001.

[71] A.N. Solan, J.A. Cesaretti, N.N. Stone, and R.G. Stock. There is no correlation
between erectile dysfunction and dose to penile bulb and neurovascular bundles
following real-time low-dose-rate prostate brachytherapy. *International Journal
of Radiation Oncology* *Biology* *Physics*, 73(5):1468–1474, 2009.

study for i-125 prostate implants. *International Journal of Radiation Oncology*

[73] Richard G. Stock, Nelson N. Stone, Jamie A. Cesaretti, and Barry S. Rosen-
stein. Biologically effective dose values for prostate brachytherapy: Effects on
psa failure and posttreatment biopsy results. *International Journal of Radiation
Oncology* *Biology* *Physics*, 64(2):527 – 533, 2006.

accuracy as a function of seed localization uncertainty in permanent prostate


Appendix
## Appendix A

### Notation

This section summarizes the notation used in this thesis. The notation can be categorized as general, specific to strand/needle representation, and specific to simulation.

#### A.1 General Notation

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$i, j, k$</td>
<td>Generic indices</td>
</tr>
<tr>
<td>$\bar{x}, \bar{y}, \bar{z}$</td>
<td>Coordinate axes</td>
</tr>
<tr>
<td>${P}$</td>
<td>Projection-image coordinate frame (2D)</td>
</tr>
<tr>
<td>${F}$</td>
<td>Fluoroscopic coordinate frame (3D)</td>
</tr>
<tr>
<td>${I}$</td>
<td>Ideal coordinate frame (3D)</td>
</tr>
<tr>
<td>PE</td>
<td>Projection error</td>
</tr>
<tr>
<td>LE</td>
<td>Localization error</td>
</tr>
</tbody>
</table>


## A.2 Notation for Strands, Needles, and Related Concepts

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\vec{f}_j(\rho)$</td>
<td>Strand $j$</td>
</tr>
<tr>
<td>$\vec{I}_j(\varrho)$</td>
<td>Needle $j$</td>
</tr>
<tr>
<td>$\vec{F}_b_i$</td>
<td>Column vector of the bead, indexed in the bead set as $i$</td>
</tr>
<tr>
<td>$\vec{I}_s_i$</td>
<td>Column vector of seed, indexed in the seed set as $i$</td>
</tr>
<tr>
<td>$FB_k$</td>
<td>$3 \times \eta_b$ matrix containing bead $i$ in column $i$. $k$ represents the number of times the bead set has been transformed.</td>
</tr>
<tr>
<td>$IS$</td>
<td>$3 \times \eta_s$ matrix containing seed $i$ in column $i$.</td>
</tr>
<tr>
<td>$\vec{F}_j b_i$</td>
<td>Column vector of the bead, assigned to a strand $j$ at the $i^{th}$ position</td>
</tr>
<tr>
<td>$\vec{I}_j s_i$</td>
<td>Column vector of the seed, assigned to a needle $j$ at the $i^{th}$ position</td>
</tr>
<tr>
<td>$FB_j$</td>
<td>$3 \times \kappa_j$ matrix containing all the beads on strand $j$.</td>
</tr>
<tr>
<td>$\vec{b}_j$</td>
<td>Centroid of strand $j$</td>
</tr>
<tr>
<td>$\vec{s}_k$</td>
<td>Centroid of needle $k$</td>
</tr>
<tr>
<td>$\eta_b$</td>
<td>Number of beads in the bead set</td>
</tr>
<tr>
<td>$\eta_s$</td>
<td>Number of seeds in the seed set</td>
</tr>
<tr>
<td>$\kappa_j$</td>
<td>Number of beads on strand $j$</td>
</tr>
<tr>
<td>$\omega_j$</td>
<td>Number of seeds on needle $j$</td>
</tr>
<tr>
<td>Symbol</td>
<td>Definition</td>
</tr>
<tr>
<td>----------</td>
<td>-----------------------------------------------------------------</td>
</tr>
<tr>
<td>$\vec{a}$</td>
<td>Needle axis</td>
</tr>
<tr>
<td>$\vec{p}$</td>
<td>Parameters for the linear model of the needle axis</td>
</tr>
<tr>
<td>$E, e$</td>
<td>Error</td>
</tr>
<tr>
<td>$\vec{u}$</td>
<td>Axis of rotation</td>
</tr>
<tr>
<td>$\psi$</td>
<td>Angle of rotation</td>
</tr>
<tr>
<td>$c(i, j), C(i, j)$</td>
<td>Costs between objects</td>
</tr>
<tr>
<td>$\vec{d}_{ij}$</td>
<td>Distance between beads</td>
</tr>
<tr>
<td>$l_k$</td>
<td>Spacing distance $k$</td>
</tr>
<tr>
<td>$f_k$</td>
<td>Frequency of spacing $k$</td>
</tr>
<tr>
<td>$\mu_k$</td>
<td>If only beads with a spacing distance of $l_k$ are connects, $\mu_k = \eta_b - f_k$ beads are to be left unassigned</td>
</tr>
<tr>
<td>$\vec{v}(j, k)$</td>
<td>Vector distance between a bead $j$ and a seed $k$</td>
</tr>
</tbody>
</table>
### A.3 Notation for Simulations

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\vec{r}$</td>
<td>Remote center of rotation</td>
</tr>
<tr>
<td>$j^{m_i}$</td>
<td>Cumulative distance along strand/needle $j$ from bead/seed 1 to bead/seed $i$</td>
</tr>
<tr>
<td>$\tau_j(\Gamma)$</td>
<td>Cumulative distance along strand $j$ from bead 1 to the point parameterized by $\Gamma$</td>
</tr>
<tr>
<td>$\Lambda$</td>
<td>Amplitude</td>
</tr>
<tr>
<td>$\lambda$</td>
<td>Wavelength</td>
</tr>
<tr>
<td>$\theta$</td>
<td>Planar orientation</td>
</tr>
<tr>
<td>$\phi$</td>
<td>Splay angle</td>
</tr>
<tr>
<td>$\delta$</td>
<td>Randomness bound</td>
</tr>
<tr>
<td>$\epsilon$</td>
<td>Error (linear regression)</td>
</tr>
<tr>
<td>$g(\omega)$</td>
<td>Objective function for a “good” sine fit</td>
</tr>
</tbody>
</table>
Appendix B

The Hungarian Algorithm

An excellent description of the Hungarian Algorithm and its variants were presented by Cooper [13]. The following explanation of the Hungarian Algorithm owes some of its structure to this excellent text.

B.1 The Assignment Problem

The goal of the assignment problem is to find a mapping of a domain set onto a range set, minimizing an objective function in the process. Distinguishing it from other optimization and mapping problems, the assignment problem has a number of features:

- The mapping is unique. One member of the domain set is assigned to exactly one member of the range set and vice-versa. The condition of uniqueness results in an invertible mapping.

- There is a cost to assign a member of the domain set to a member of the range
set. This cost is factored into the objective function.

- Each cost is *independent* of the mapping used. No matter what assignments are made between the domain set and the range set, the costs remain unaltered.

- The objective function is simply the *sum* of the cost of each assignment made.

These conditions are illustrated in a classic example of the assignment problem: the task of assigning workers to jobs to obtain the lowest total salary cost. Assigning workers to jobs is a unique assignment: each worker can only be assigned to one job and each job needs exactly one worker assigned to it. Furthermore, it costs a fixed amount for worker \( i \) to complete job \( j \). That cost does not change with the assignment of other workers. Finally, the total salary cost is the sum of the individual assignment costs.

Expressed mathematically, these conditions and the structures on which they operate form the basis of the assignment problem and its method of solution. For \( n \) workers and \( n \) jobs, an \( n \times n \) assignment matrix is constructed. This matrix will be modified again and again in order to optimize the objective function.

\[
A = \begin{bmatrix}
a_{11} & a_{12} & \cdots & a_{1n} \\
a_{21} & a_{22} & \cdots & a_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
a_{n1} & a_{n2} & \cdots & a_{nn}
\end{bmatrix}
\]  

(B.1)

\[a_{ij} \in \{0, 1\}\]  

(B.2)

If \( a_{ij} = 1 \) then worker \( i \) is said to be assigned to job \( j \). Similarly, if \( a_{ij} = 0 \), worker \( i \) is not assigned to job \( j \). The mapping is unique if the following constraint
APPENDIX B. THE HUNGARIAN ALGORITHM

is satisfied.

\[ \forall j, \sum_{i=1}^{n} a_{ij} = 1 \]  \hspace{1cm} (B.3)

\[ \forall i, \sum_{j=1}^{n} a_{ij} = 1 \]  \hspace{1cm} (B.4)

Each assignment of a worker to a job also has a cost associated with it. These costs are represented in a second \( n \times n \) matrix corresponding to the assignment matrix. The cost matrix contains parameters: they remain constant for any given problem instantiation.

\[
C = \begin{bmatrix}
    c_{11} & c_{12} & \cdots & c_{1n} \\
    c_{21} & c_{22} & \cdots & c_{2n} \\
    \vdots & \vdots & \ddots & \vdots \\
    c_{n1} & c_{n2} & \cdots & c_{nn}
\end{bmatrix}
\]  \hspace{1cm} (B.5)

The objective function, \( T \), is simply the sum of the costs corresponding to each assignment made.

\[
T = \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij}a_{ij}
\]  \hspace{1cm} (B.6)

Since \( a_{ij} \) is 1 or 0, depending on if the assignment has been made, \( T \) reflects only the costs of assignments in the mapping, and no others. Thus, with cost matrix \( C \) remaining fixed, \( A \) must be modified to minimize the objective function, \( T \).

A note on formulation: The Hungarian Algorithm is generally formulated such that the number of workers \( n \) equals the number of jobs \( m \). If \( n \neq m \), additional
rows or columns can be added to both the cost and assignment matrices, representing 'dummy’ workers or jobs. If these 'dummy’ objects have a fixed assignment cost of 0 (or another constant), then in no way will their assignment affect the objective function, $T$. Thus, the worker/job with the worst cost profile will be assigned to one of these dummy objects, thus tagging the worker/job as unassigned. We will proceed with the assumption that $n = m$, since we have satisfied that $n \neq m$ problem can be modified to an $n = m$ problem.

### B.2 Procedure of the Hungarian Algorithm

The Hungarian Algorithm aims to optimally solve the assignment problem by iteratively changing the assignment matrix $A$ until the optimal mapping has been found. The changes made to $A$ are governed by a series of rules, centering around the outcomes of operations on the cost matrix. To facilitate the proceeding discussion, a more compact representation of the assignment matrix and the manipulated cost matrix is introduced.

Since the assignment costs themselves to not change while finding the mapping, a manipulations matrix, $C'$ is defined. When the matrix is unmodified from the original cost matrix, $0C' = C$.

$$
0C' = \begin{bmatrix}
c'_{11} & c'_{12} & \ldots & c'_{1n} \\
c'_{21} & c'_{22} & \ldots & c'_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
c'_{n1} & c'_{n2} & \ldots & c'_{nn}
\end{bmatrix}
= \begin{bmatrix}
c_{11} & c_{12} & \ldots & c_{1n} \\
c_{21} & c_{22} & \ldots & c_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
c_{n1} & c_{n2} & \ldots & c_{nn}
\end{bmatrix} \quad \text{(B.7)}
$$

A series of operations will be introduced to modify $C'$ and assist in finding the
optimal mapping.

Since $A$ is a binary matrix of the same size as $C'$, the two matrices can be overlaid. If $a_{ij} = 1$, then $c_{ij}$ will be marked with an asterisk. If $a_{ij} = 1$, then $c_{ij}$ will be left lacking an asterisk. An example follows.

$$C' = \begin{bmatrix} 13 & 19 & 0 & 0 & 6 \\ 19 & 0 & 6 & 19 & 0 \\ 0 & 1 & 12 & 18 & 19 \\ 6 & 7 & 18 & 12 & 13 \\ 7 & 13 & 24 & 6 & 12 \end{bmatrix}, \quad A = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$C^* = \begin{bmatrix} 13 & 19 & 0 & 0 & 6^* \\ 19 & 0^* & 6 & 19 & 0 \\ 0 & 1 & 12 & 18 & 19 \\ 0^* & 1 & 12 & 6 & 7 \\ 1 & 7 & 18 & 0 & 6 \end{bmatrix}$$

If worker $i$ is assigned to job $j$, we call entry $c'_{ij}$ assigned. Theoretically, any entry in the $C^*$ can be assigned. In practice, to guarantee an optimal assignment, only zeros will be assigned. Accordingly, these zeros will be called assigned and unassigned zeros.

The Hungarian Algorithm involves three basic steps:

1. Subtracting the minimum value of each row and column from that row and column of the cost matrix. $C'$ is initialized as $C$. Negative values are then removed and zeros are introduced.
2. **Find the maximal assignment of zeros in the cost matrix.** The maximum number of allowed zeros, $k$, are assigned, forming $A$ as (represented in $C^*$). In the process, all assigned and unassigned zeros are ‘covered’: their rows and columns marked.

3. **Introducing additional zeros.** The covering generated in the previous step is used to subtract constants from the matrix, so as to introduce new zeros to matrix $C'$.

Steps 2 and 3 are then repeated until a complete assignment ($k = n$) is obtained.

A detailed description and example of the procedure of the algorithm follows. A series of proofs that solidify the reasoning of the algorithm is found in Section B.3.

1. **Subtracting the minimum value of each row and column from that row and column of the cost matrix.**

The first step in modifying the manipulation matrix $C'$ introduces zeros and removes negative values. Both tasks are accomplished by subtracting constants from full rows and columns of the manipulation matrix. Adding a constant to a row or column does not change resulting optimal mapping, so constants can be added and subtracted as necessary (see Section B.3).

Row subtraction is represented as

$$
_{1}C' :=_{0} C' \square \alpha_{n \times 1} \mid (_{1}C'_{i} =_{0} C'_{i} - \alpha_{i}) \forall i, j \quad (B.8)
$$

Column subtraction is represented as

$$
_{1}C' :=_{0} C' \square \beta_{1 \times n} \mid (_{1}C'_{i} =_{0} C_{i} - \beta_{i}) \forall i, j \quad (B.9)
$$
To remove negative values and introduce zeros, the smallest value in each row of \( C' \) is subtracted from that row. To introduce more zeroes, the smallest value of each column is subtracted from that column. Algorithmically, the cost matrix is operated on as follows.

\[
C'_{i*} \leftarrow C'_{i*} - \text{argmin}(C'_{i*}) \\
C'_{*j} \leftarrow C'_{*j} - \text{argmin}(C'_{*j})
\]

Now that the operation of adding constants to rows and columns has been defined, the initial \( C' \) is modified to remove and negative values in the matrix. Specifically, the minimum value of each row is subtracted from that row of the matrix.

\[
C' = \begin{bmatrix}
13 & 0 & -19 & -19 & -13 \\
19 & -2 & 4 & 17 & -2 \\
3 & 4 & 15 & 23 & 22 \\
1 & 2 & 13 & 7 & 8 \\
1 & 7 & 18 & 0 & 0 \\
\end{bmatrix} \quad \square \quad \begin{bmatrix}
-19 \\
-2 \\
3 \\
1 \\
0 \\
\end{bmatrix}
\]

\[
\Rightarrow \begin{bmatrix}
13 & 19 & 0 & 0 & 6 \\
19 & 0 & 6 & 19 & 0 \\
0 & 1 & 12 & 18 & 19 \\
0 & 1 & 12 & 6 & 7 \\
1 & 7 & 18 & 0 & 0 \\
\end{bmatrix}
\]

The minimum value in each column of this matrix happens to be zero. If this
APPENDIX B. THE HUNGARIAN ALGORITHM

\[
C^* = \begin{bmatrix}
13 & 19 & 0 & 0^* & 6 \\
19 & 0^* & 6 & 19 & 0 \\
0 & 1 & 12 & 18 & 19 \\
0^* & 1 & 12 & 6 & 7 \\
1 & 7 & 18 & 0 & 6 \\
\end{bmatrix}
\]

(a) This assignment is not maximal, since more zeros can be assigned without violating any uniqueness constraint.

\[
C^* = \begin{bmatrix}
13 & 19 & 0^* & 6 & 0 \\
19 & 0^* & 6 & 19 & 0 \\
0 & 1 & 12 & 18 & 19 \\
0^* & 1 & 12 & 6 & 7 \\
1 & 7 & 18 & 0^* & 6 \\
\end{bmatrix}
\]

(b) This assignment is maximal, because no configuration of unique assignments produces a larger number of assigned zeros. The assignment is not complete, as \(k = 4\) while \(n = 5\).

Figure B.1: An example of a maximal and non-maximal assignment.

were not the case, the minimum value of each column would be subtracted from that column.

Now that all matrix elements are non-negative, the zeros in the matrix can be assigned to obtain our first value of \(A\).

2. Finding the maximal assignment of zeros in the cost matrix.

Zeros are introduced because they represent possible assignments. If \(c'_{ij} = 0\), then worker \(i\) may be assigned to job \(j\). If the assignment occurs, then \(c'_{ij}\) can be starred accordingly. If the number of assigned zeros, \(k\), equals the matrix dimension, \(n\), then the mapping is said to be complete. A maximal mapping occurs when the largest number of zeros \((k \leq n)\) is assigned without breaking the constraint of uniqueness. An example of a maximal and non-maximal assignment can be found in Figure B.1.

There is a particular method to finding a maximal assignment in a matrix of arbitrary size. The first part of finding a maximal assignment identifies free zeros and assigns them. The second part assigns the remaining zeros in a more
involved manner.

(a) **Finding and assigning free zeros.** A free zero is the only zero in a row or column. These zeros can be assigned without constraining other zeros in the matrix. To elaborate, if a row contains multiple zeros, the effect of assigning any one of them has an identical affect on the row: all other zeros are excluded from assignment. However, each of these zeros also have an effect on the columns in which they lie. A *row-free zero* is one that lacks zeros in its column and can therefore be assigned without effects beyond its row. A *column-free zero* is one which lacks other zeros in its row.

These zeros are the first to be assigned. In fact, assigning free zeros first is the addition to the Hungarian Algorithm that brings the time complexity from $O(n^4)$ to $O(n^3)$.

As soon as a row-free zero has been assigned, no other zero in that row is eligible for assignment. The row, called an *essential row* can be ignored, as no other zeros in them can be assigned without breaking the uniqueness constraint. *Essential columns* containing a column-free zero are similarly ignored. With additional unassigned zeros removed from the problem, further column-free and row-free zeros are then found.

First, row-free zeros are identified. In the following matrix, row 1 is an essential row because the 0 at $C_{13}$ has no other zeros in the column. Row 2 is an essential row because both zeros are row-free: they have no other zeros in their respective columns.
The next step is to find the essential columns in the matrix. The only essential column present is column 1. In fact, column 1 has two column-free zeros, in rows 3 and 4. The lesser of the two row indices is arbitrarily chosen for the assigned zero.

Now that all the essential rows and columns have been found, the process can be repeated. There are no more unique zeros in any rows or columns of the matrix as it stands. Nevertheless, by ignoring the zeros in ”covered” essential rows and columns from this point forward, more ”second-order” free zeros exist. Two such row-free zeros are found in positions (5,4) and (5,5). The zero in position (5,4) is arbitrarily assigned. Since they both lie in row 5, row 5 becomes an essential row.
The process is repeated until there are no free zeros of any order remaining.

(b) **Assigning the remaining zeros maximally.** Occasionally, there are zeros left unassigned and uncovered after the previous step has terminated. In this case, the zeros are not unique in their row or column: they are paired in rows and columns such that they prevent each other from being labeled unique. One such matrix is presented.

\[
\begin{pmatrix}
0^* & 2 & 0 & 0 \\
0 & 0^* & 4 & 0 \\
1 & 0 & 0^* & 5 \\
1 & 0 & 0 & 2
\end{pmatrix}
\]

Substructures of a similar form may be found within larger matrices. To solve these substructures, the first zero eligible for assignment is assigned in each row and column. The above matrix demonstrates the initial assignment. It is unique, but somewhat arbitrary and not maximal.

Since this is not a maximal assignment, each zero in an unassigned row or column is checked to see if it can be assigned to increase the total number of assignments. Thus, for each assigned zero in an unassigned row, a pattern
of the following type is constructed:

\[
\begin{align*}
C_{i_1,j_1} &= 0 \\
C_{i_2,j_1} &= 0^* \\
C_{i_2,j_2} &= 0 \\
C_{i_3,j_2} &= 0^* \\
&\vdots \\
C_{i_{m-1},j_m} &= 0 \\
C_{i_m,j_m} &= 0^* \\
C_{i_m,j_m} &= 0 \\
C_{i_m,j_m} &= 0^*
\end{align*}
\]

The pattern starts with the unassigned zero in an unassigned row. The second item is the assigned zero in the column of the first, unassigned zero. The third item is the unassigned zero with the lowest column index in the same row as the second item, and so on.

For our matrix, the first row with no assigned zero is row 4. The first unassigned zero in that row is in position (4,2).
The sequence, alternating rows and columns, is constructed below.

\[
\begin{bmatrix}
0^*_4 & 2 & 0 & 0^*_5 \\
0_3 & 0^*_2 & 4 & 0 \\
1 & 0 & 0^*_1 & 5 \\
1 & 0_1 & 0 & 2
\end{bmatrix}
\]

The sequence, alternating rows and columns, is constructed below.

\[
C_{4,2} = 0
\]
\[
C_{2,2} = 0^*
\]
\[
C_{2,1} = 0
\]
\[
C_{1,1} = 0^*
\]
\[
C_{1,4} = 0
\]

A series in a given matrix terminates on one of two conditions:

i. An assigned/unassigned zero does not exist in the current column/row.

ii. The only available assigned and unassigned zeros exist in rows and columns that have previously been traversed in the series.

The goal is to find a set of zeros that both begins and terminates with an unassigned zero. If such a set is found, the assignments can be switched \((0 \leftrightarrow 0^*)\) to obtain an additional assigned zero.

In the example matrix, the sequence does terminate with an unassigned zero. The assigned zeros can thus become unassigned and the assigned zeros, unassigned. In this way, the number of assigned zeros, \(k\), is increased.
by one.

\[ C'_{4,2} = 0^* \]
\[ C'_{2,2} = 0 \]
\[ C'_{2,1} = 0^* \]
\[ C'_{1,1} = 0 \]
\[ C'_{1,4} = 0^* \]

Now that the assignment has been switched, the cost matrix is now fully assigned.

\[
\begin{bmatrix}
0 & 4 & 2 & 0 & 0^* \\
0^* & 0 & 2 & 4 & 0 \\
1 & 0 & 0^* & 5 \\
1 & 0^* & 0 & 2
\end{bmatrix}
\]

Should the sequence have terminated with an assigned zero, the last two items in the sequence – the last unassigned and assigned zero – are removed from it. The next candidate unassigned zero is then chosen in the same row as the removed unassigned zero. The process continues recursively until either an terminal unassigned zero is found and a switch is made, or the sequence has exhausted all possibilities and the assignment remains as is.

If a maximal assignment has not been reached, the process is repeated on
the next unassigned zero in an unassigned row. If there are yet unassigned zeros in unassigned columns, the process is repeated for these zeros with the obvious row/column modifications.

So far, only essential rows and columns have been covered with covering lines. When assignments are made using the sequence approach, the assigned zeros do not fall in an essential row or column. From Theorem 3, the columns of these additional assigned zeros are covered to ensure that all zeros are covered: assigned and unassigned.

3. **Introducing additional zeros**

With \( k \) covering lines in place – one for each of \( k \) assigned zeros – the assignment is evaluated for completeness.

If \( k = n \), an optimal assignment has been found and the algorithm can be terminated.

If \( k < n \), additional zeros can be introduced. From Theorem 4, if \( s \) is the smallest uncovered cost in the matrix, \( s \) is added to each covered line in the matrix. Then, \( s \) is subtracted from every row and thus every element in the matrix.

With this procedure, each covered line has \( s \) added and \( s \) subtracted from it. All covered values therefore remain unchanged. Values at the intersection of two covering lines have \( 2s \) added and \( m \) subtracted: they are increased by \( s \). Finally, and most importantly, values that are uncovered are decreased by \( s \). Since \( s \) is the smallest uncovered element, a new zero is introduced.
In the example, the smallest uncovered element is 1. $s = 1$ is added to all covered lines in the matrix and subtracted from all elements.

\[
\begin{pmatrix}
13 & 19 & 0^* & 0 & 6 \\
0 & 0^* & 6 & 19 & 0 \\
0^* & 1 & 12 & 18 & 19 \\
0 & 1 & 12 & 6 & 7 \\
7 & 18 & 0^* & 0 \\
\end{pmatrix}
\]

\[
\begin{pmatrix}
1 \\
1 \\
0 \\
0 \\
1 \\
\end{pmatrix}
\]

\[
\begin{pmatrix}
14 & 19 & 0^* & 0 & 6 \\
0 & 0^* & 6 & 19 & 0 \\
0^* & 0 & 11 & 17 & 18 \\
0 & 0 & 11 & 5 & 6 \\
7 & 18 & 0^* & 0 \\
\end{pmatrix}
\]

All the initial zeros have been preserved and new zeros have been introduced. With this new cost matrix, we return to step 2: finding the maximal assignment. The algorithm iterates between steps 2 and 3 until $k = n$. When $k = n$, the value of $A$ is extracted from the modification matrix $C^*$. The matrix $A$ represents a complete, optimal assignment.
B.3 Foundations of the Hungarian Algorithm

The three steps discussed in the preceding section rest on the foundation of four theorems, which will be proven below. The theorems address the soundness of the algorithm and its operations.

- If the objective function has a lower bound, then we are guaranteed to find an optimal mapping by achieving that lower bound. Proof 1 discusses how this lower bound is introduced.

- Since only zeros in $C^*$ are assigned, constants are subtracted from full rows and columns to introduce more zeros. Proof 2 ensures that this manipulation does not change the optimal mapping.

- One a maximal assignment of these zeros has been made, all zeros in the matrix must be covered. Proof 3 implies that for $k$ assigned zeros in a maximal assignment, $k$ lines can be used to cover every zero: assigned or unassigned.

- Proof 4 ensures that if the assignment is not complete ($z < n$), a new zero can be introduced and assigned.

To find the optimal assignment, the objective function must be minimized. Since the objective function has no lower bound, there is little way of knowing when a minimum has been reached. The success Hungarian Algorithm lies in its enforcing a lower bound on the objective function, by ensuring that only costs of zero can be assigned. If all other costs are positive, the lower bound on the assignment cost is 0.
Theorem 1 If all costs are subject to the restriction \( c_{ij} \geq 0 \), the cost function, \( T \), has a lower bound of 0.

Proof

\[
T = \sum_{i=1}^{n} \sum_{j=1}^{m} c_{ij}a_{ij}
\]

\( a_{ij} \in 0, 1 \)

\( c_{ij} \geq 0 \)

In order for the cost matrix to be minimized, \( c_{ij} \) must be at a minimum when \( a_{ij} = 1 \).

\( (a_{ij} = 1 \Rightarrow c_{ij} = 0) \iff T = 0 \)

Thus, \( T \) has a lower bound of 0.

In order to achieve this lower bound, only zeros can be assigned.

A complication arises when considering the initial values in the cost matrix: cost matrices may contain negative values. These negative values must be eliminated in order to satisfy the \( c_{ij} \geq 0 \) constraint. Furthermore, in order assign only zeros in \( C' \), and yet have a complete, optimal assignment, a method of introducing zeros into a cost matrix is necessary. To introduce zeros and remove negative values, constants are added to rows and columns of \( C' \).
Theorem 2 Adding values to rows or columns of the modification matrix may change the matrix itself, but does not change the optimal mapping.

Proof Given a set of constants, \( \alpha_i \), added to row \( i \) of the cost matrix, and \( \beta_j \), added to column \( j \) of the modification matrix,

\[
\begin{align*}
1c'_{ij} &= 0c'_{ij} + \alpha_i + \beta_j \\
T' &= \sum_{i=1}^{n} \sum_{j=1}^{m} 1c'_{ij} a_{ij} \\
&= \sum_{i=1}^{n} \sum_{j=1}^{m} (0c'_{ij} + \alpha_i + \beta_j) a_{ij} \\
&= \sum_{i=1}^{n} \sum_{j=1}^{m} 0c'_{ij} a_{ij} + \sum_{i=1}^{n} \sum_{j=1}^{m} \alpha_i a_{ij} + \sum_{i=1}^{n} \sum_{j=1}^{m} \beta_j a_{ij} \\
&= \sum_{i=1}^{n} \sum_{j=1}^{m} 0c'_{ij} a_{ij} + \sum_{i=1}^{n} \alpha_i (\sum_{j=1}^{m} a_{ij}) + \sum_{j=1}^{m} \beta_j (\sum_{i=1}^{n} a_{ij}) \\
&= \sum_{i=1}^{n} \sum_{j=1}^{m} 0c'_{ij} a_{ij} + \sum_{i=1}^{n} \alpha_i (1) + \sum_{j=1}^{m} \beta_j (1) \leftarrow (B.3),(B.4) \\
&= T + \sum_{i=1}^{n} \alpha_i + \sum_{j=1}^{m} \beta_j
\end{align*}
\]

The two objective function results are separated by a constant: \( \sum_{i=1}^{n} \alpha_i + \sum_{j=1}^{m} \beta_j \). Therefore, the same mapping minimizes both functions, no matter what values have been added to rows or columns.

Adding constants to rows and columns does not change the optimal mapping. Negative values are removed from the initial modification matrix, \( 0C' \) in this way:
the smallest value in each row is subtracted from that row.

Now that the modification matrix contains some zeros, a procedure must be developed for assigning them. In the Hungarian Algorithm, lines are covered if they are (1) essential rows, or (2) columns containing an assigned zero not in an essential row (called a supplementary column for the purposes of this section). This next proof shows that if an unassigned zero is left uncovered, then it can be assigned.

This system of assigning zeros is based on the work of two Hungarian mathematicians that give the algorithm its name.

### Theorem 3

If a maximal assignment in a modification matrix consists of $k$ assigned zeros, $k$ lines may be used to cover all zeros – assigned and unassigned – in the matrix.

### Proof

Kuhn’s algorithm rests on the foundation laid Hungarian mathematicians: Dénes Kőnig and Jenő Egerváry. While Kőnig’s work is often mentioned in passing, the Hungarian Algorithm leans heavily on his work both theoretically and conceptually. Kőnig proved, in the language of bipartite graphs, that a maximum $k$ assigned zeros correspond to a minimum of $k$ lines needed to cover the zeros [39]. Egerváry extended this to apply to weighted bipartite graphs, equivalent to arbitrary cost matrices [22].

A simple proof of Kőnig’s theorem can be found as a proof-by-contradiction. There are many ways of assigning $k$ lines to cover $k$ assigned zeros. One way to do so is by covering essential rows and supplementary columns - those columns with assigned zeros in non-essential rows. Recall that the assigned zero in an essential row has no unassigned zeros in its column. A supplementary column contains any assigned zero not in an essential row. Thus, any assigned zero either falls in an
essential row or a supplementary column. We prove that an unassigned zero in position \((i, j)\) is not covered by an essential row or supplementary column can be assigned and thus covered.

Since all \(0^*\) are covered, if there is a \(0^*\) in column \(j\), that \(0^*\) must consequently be in an essential row. However, recall that the assigned zero in an essential row has no unassigned zeros in its column. The presence of the unassigned zero proves that such a \(0^*\) cannot exist. Similarly, if there is a \(0^*\) in row \(i\), that \(0^*\) must fall in a supplementary column. However, this is impossible as the unassigned zero in position \((i, j)\) has no zeros in its column: its presence forces row \(i\) to be an essential row. Thus, if the unassigned zero is uncovered, it is able to be assigned. And, furthermore, since there are \(k\) lines for \(k\) assignments, and no uncovered zeros possible, König’s theorem is proven for this construction.

All zeros can be covered with \(k\) lines. The next step is to increase the number of assignable zeros to increase \(k\). When \(k = n\), the assigned zeros in \(C^*\) represent the optimal assignment \(A\).

**Theorem 4** Where a maximum \(k\) zeros can be assigned in a \(C^*\) matrix, If \(k = n\), the maximal assignment is complete and optimal. If \(k < n\), a new zero can be introduced into the matrix.

**Proof** König’s result implies that a complete assignment, with \(n\) assigned zeros, requires \(n\) lines to cover all zeros. Since \(n\) lines allow for one line in every row, they are able to cover all elements in the matrix, including the zeros.

If the maximal assignment contains \(k < n\) assigned zeros, another assigned zero can be introduced with the following procedure.
1. Arrange \( k \) lines to cover all zeros.

2. Let \( m \) be the minimum uncovered cost. Subtract \( m \) from every element in the matrix, and add \( m \) to every covered line.

Uncovered elements of the matrix are reduced by \( m \), covered elements of the matrix are unchanged, and doubly covered elements are increased by \( m \). Thus, a new zero is introduced to the matrix.

Assigning only zeros in \( C^* \) therefore guarantees an optimal assignment, if that assignment is complete. \( C^* \) is modified by adding constants to rows and columns without changing the resulting optimal assignment. A maximal assignment can be found by ensuring that all zeros are covered: either in an essential row or a supplementary column. Finally, the covering can be used to introduce new zeros into uncovered spaces. These new uncovered zeros are assigned and covered in the next algorithm iteration. When the number of assigned zeros, \( k \), equals the matrix dimension \( n \), the assignment is complete and optimal. \( A \), extracted from \( C^* \), is the solution to the Hungarian Algorithm.
Appendix C

Sensitivity to Parametric Variations

Algorithmic accuracy for simulations and clinical data was obtained using a single parameter set. To determine how sensitive the algorithm is to changes in these parameters, algorithmic accuracy was measured over a range of parameters.

The cost function used to reconstruct needles took the following form.

\[ C(i, j) = \alpha_1 \cdot (c_d)^{\beta_1} + \alpha_2 \cdot (c_\phi)^{\beta_2} \]  \hspace{1cm} (3.10)

\( \alpha_1 = 0.0521 \) \hspace{1cm} Weight for distance cost function

\( \beta_1 = 1.5 \) \hspace{1cm} Exponent for distance cost function

\( \alpha_2 = 0.9479 \cdot s \) \hspace{1cm} Weight for angular cost function

\( \beta_2 = 4 \) \hspace{1cm} Exponent for angular cost function

Table C.1: Strand reconstruction parameters used for simulations and clinical results. 

\( s \) is the spacing (in cm) currently being sought in the bead set.
Table C.2: Distribution of costs in the bead cloud. The mean of the costs for the most common spacing is reported, as is the standard deviation of those costs.

Needle reconstruction parameters were manually selected and are reported in Table C.1. These parameters produced perfect reconstructions in all 9 clinical datasets. The perfect reconstructions and resulting perfect registrations were used as the known correspondence between beads and seeds. Accuracy could then be calculated as the percentage of correct matchings obtained with the new parameter set.

Parameters were varied, first to determine the ratio between $\alpha_1$ and $\alpha_2$. Then, $\beta_1$ and $\beta_2$ were varied against each other.

### C.1 Distance and angular costs

The distance cost, $c_d$, and the angular cost, $c_\phi$, were calculated as

\[
\begin{align*}
c_d(i, j) &= |d_{ij} - l_k| \\
c_\phi(i, j) &= \cos^{-1} \left( \frac{\vec{a} \cdot (\vec{d}_{ij})}{d_{ij}} \right)
\end{align*}
\]
The distance cost was represented in millimeters, and the angular cost, in radians. Thus, while distance cost between two beads was unbounded, the angular error was bounded at π. To determine how these costs varied, the average and standard deviation for each cost was calculated for all bead pairs. Values are reported in Table C.2.

C.1.1 Hypothesis

From the values in Table C.2, the ratio of distance to angular cost was found.

\[
\frac{c_d}{c_\phi} = 14.25 \pm 0.80
\]

It was therefore expected that, when \( \beta_1 = \beta_2 \), the optimal values of \( \alpha_1 \) and \( \alpha_2 \) would occur near

\[
\frac{\alpha_1}{\alpha_2} = \frac{1}{14.25}
\]

\[
\frac{\alpha_1}{1 - \alpha_1} = \frac{1}{14.25} \quad \leftarrow \quad (C.1)
\]

\[
14.25 \alpha_1 = 1 - \alpha_1
\]

\[
\alpha_1 = \frac{1}{15.25}
\]

\[
\alpha_1 = 0.0656
\]

If distance and angular costs are scaled such that their means are equal, the standard deviations of the both bead pair costs come into closer alignment (\( \sigma_{c_\phi} 14.25 = 10.77 \)).

If the bead sets are scaled such that they are similarly distributed, the best accuracy should occur at \( \beta_1 \approx \beta_2 \). This assumes that the costs which represent correct
assignments are also similarly distributed.

C.2 Weighting of $\alpha_1$ and $\alpha_2$

Since scaling of the cost function, $C(i, j)$, does not affect the outcome of the Hungarian Algorithm, $\alpha_1$ and $\alpha_2$ can be factored. Thus, it is the ratio between $\alpha_1$ and $\alpha_2$ rather than their absolute values that affects the outcome of the algorithm. A relationship between $\alpha_1$ and $\alpha_2$ was imposed for testing.

$$\alpha_1 + \alpha_2 = 1 \quad (C.1)$$

For this test specifically, $\beta_1$ has been held equal to $\beta_2$

$$\beta_1 = \beta_2 \quad (C.2)$$

To test the effect of ratio of $\alpha_1$ to $\alpha_2$ on accuracy, different ratios were tested for different values of $\beta_1 = \beta_2$. $\alpha_1$ and $\beta_1$ were therefore varied independently of each other, while $\alpha_2$ and $\beta_2$ were determined by Equations C.1 and C.2 respectively.

C.2.1 Broad survey: $0.1 < \alpha_1 < 0.9$

First, the distance weight, $\alpha_1$, was varied from 0.1 to 0.9 in steps of 0.1. Angular weight was calculated from the distance weight; it was varied from 0.9 to 0.1 in steps of 0.1. The exponents of the two costs were kept equal and varied from 1 to 8 in steps of 1. The results are shown in Figure C.1.

As $\alpha_1$ decreased towards 0.1, registration accuracy increased. Every value of $\alpha_1$
(a) Registration accuracy over a range of distance weights ($0.1 < \alpha_1 < 0.9$) and exponents ($1 < \beta_1 < 8$) values

(b) Standard deviation of the sample for the registration accuracies in (a).

Figure C.1: Registration accuracy results over different values of $\alpha_1$ ($\alpha_2 = 1 - \alpha_1$) and $\beta_1$ ($\beta_1 = \beta_2$). Accuracy results from all nine clinical bead sets were averaged to generate each data point.
was significantly different from every other \((p < 0.038)\), with the exceptions of the \(\alpha_1 = 0.2\) and \(0.3\) \((p = 0.11)\) and \(\alpha_1 = 0.1\) and \(0.8\) \((p = 0.19)\). The lack of a significant result between the \(0.1\) and \(0.8\) may have been due to the the variability at an \(\alpha_1\) value of \(0.8\).

At small values of \(\alpha_1\), small values of \(\beta_1\) appear to have improved registration accuracy. Specifically, \(\beta_1 = 1\) achieved statistical significance when compared with \(\beta_1 = 2\) \((p = 0.010)\), \(3\) \((p = 0.016)\), and \(5\) \((p = 0.046)\). However, at higher values of \(\alpha_1\), higher \(\beta_1\) values performed substantially better. The only statistically significant difference at higher \(\beta_1\) values was between \(\beta_1 = 6\) and \(7\) \((p = 0.050)\). All other comparisons did not achieve statistical significance \((p > 0.508)\), though some values were close to 5% significance.

The maximum accuracy in the graph was 99.07% and occurred when \(\alpha_1 = 0.2\) and \(\beta_1 = 1.5\). To ensure an average registration accuracy above 97%, \(\alpha_1\) must remain less than \(0.3\) \((\alpha_2 < 0.7)\) and \(\beta_1 = \beta_2\) must remain below 2. This suggests that angular weighting plays an important role in registration accuracy. Furthermore, \(\beta_1\) and \(\beta_2\) should remain small so as not to impose too steep a penalty on the cost functions.

**C.2.2 Exploring a maximum: \(0.05 < \alpha_1 < 0.3\)**

More detail was exposed around \(\alpha_1 = 0.2\) and \(\beta_1 = 1.5\), the maximum of the previous graph. \(\alpha_1\), was varied from 0.05 to 0.3 in steps of 0.05. The exponents of the two costs were again kept equal and varied from 1 to 3 in steps of 0.5. The results are shown in Figure C.2.

There were a series of local maxima that appear along a diagonal of the graph. However, because there were no consistent patterns of change across \(\alpha_1\) values, only
Figure C.2: Registration accuracy results over different values of $\alpha_1$ ($\alpha_2 = 1 - \alpha_1$) and $\beta_1$ ($\beta_1 = \beta_2$), centered around the maximum of Figure C.1. Accuracy results from all nine clinical bead sets were averaged to generate each datapoint.
one pair of values showed a significant difference: $\alpha_1 = 0.15$ and $0.3$ ($p = 0.038$). Other pairs of values did not achieve statistical significance ($p > 0.0729$). Only one value of $\beta_1$, $\beta_1 = 3$, achieved statistical significance with all others ($p < 0.042$).

There are three local maxima with registration accuracies of 99.91%, at $\beta_1 = 1.0$, 1.5, and 2.0 and $\alpha_1 = 0.25$, 0.15, and 0.10 respectively. Furthermore, as long as either $\alpha_1$ remained less than 0.20 or $\beta_1$ remained less than 2.5, an accuracy above 97% was achieved.

### C.2.3 The original hypothesis: $0.01 < \alpha_1 < 0.08$

Since distance and angular cost would achieve the same mean with a small value of $\alpha_1$ ($\alpha_1 = 0.0656$), the original hypothesis suggested that smaller values of $\alpha_1$ might produce improved results. To investigate this, $\alpha_1$ was varied from 0.01 to 0.08 in steps of 0.01. $\beta_1$ was independently varied from 1 to 4 in steps of 0.5. Results are shown in Figure C.3.

Within these bounds, there is a trend towards higher accuracies for larger values of $\alpha_1$ and $\beta_1$. Local maxima can be found at larger values of $\alpha_1$ and $\beta_1$ while smaller values in both produce decreases in accuracy. If only one of the two values is large or small, there seems to be little effect on accuracy, producing a plateau at 97.02% accuracy.

Because of this correlated effect of $\alpha_1$ and $\beta_1$, differences when changing any one variable tended not to be statistically significant. Only the smallest value of $\alpha_1$, 0.01 was statistically different from other values up to and including $\alpha_1 = 0.07$ ($p < 0.040$). $\alpha_1 = 0.01$ and $\alpha_1 = 0.08$ fell out of the bounds of statistical significance ($p = 0.052$). All other pairs of $\alpha_1$ values did not achieve statistical significance ($p > 0.169$).
Figure C.3: Registration accuracy results over different values of $\alpha_1$ ($\alpha_2 = 1 - \alpha_1$) and $\beta_1$ ($\beta_1 = \beta_2$), covering the hypothesis value of $\alpha_1 = 0.0656$. Accuracy results from all nine clinical bead sets were averaged to generate each data point.
Similarly, only $\beta_1 = 1$ was statistically different from all other exponent values ($p < 0.029$). Other pairs of $\beta_1$ values did not achieve statistical significance ($p > 0.0897$).

The maximum of the graph obtained an accuracy measure of 99.91% and occurred when $\alpha_1 = 0.06$ and $\beta_1 = 3$. Furthermore, as long as $\alpha_1$ was greater than 0.02 and $\beta_1$ was greater than 1.5, accuracy remained above 97%.

### C.3 Weighting of $\beta_1$ and $\beta_2$

To explore the effects of different cost-function exponents on the accuracy of the algorithm, $\beta_1$ and $\beta_2$ were varied independently of each other. While $\beta_1$ and $\beta_2$ were varied, $\alpha_1$ and $\alpha_2$ were held constant.

#### C.3.1 A first local maximum: $\alpha_1 = 0.25$, $\alpha_1 = 0.75$

In Section C.2.2, one of the local maxima with 99.91% registration accuracy was found at $\alpha_1 = 0.25$. In this test, $\alpha_1$ was held at 0.25 while $\alpha_2$ was held at 0.75. $\beta_1$ was varied from 1 to 3 in steps of 0.5 and $\beta_2$ was independently varied from 1 to 5 in steps of 0.5. Results are found in Figure C.4.

Lower values of $\beta_1$ and $\beta_2$ appear to produce better registrations. Specifically, $\beta_1 = 2.5$ was significantly different from all other $\beta_1$ values ($p < 0.015$), as was $\beta_1 = 3.0$ ($p < 0.004$). All other $\beta_1$ pairs were not significantly different from each other ($p > 0.433$).

As $\beta_2$ changes, accuracies did become statistically different from one another. $\beta_2 = 1.0$ was significantly different from $\beta_2 > 4.0$ ($p < 0.010$); $\beta_2 = 1.5$ was significantly
(a) Registration accuracy over a range of distance cost exponents ($1 < \beta_1 < 3$) and angular cost exponents ($1 < \beta_2 < 5$) values.

(b) Standard deviation of the sample for the registration accuracies in (a).

Figure C.4: Registration accuracy results over different values of $\beta_1$ and $\beta_2$. $\alpha_1$ has been held at 0.25 and $\alpha_2$ has been held at 0.75. Accuracy results from all nine clinical bead sets were averaged to generate each datapoint.
different from $\beta_2 > 3.5$ ($p < 0.027$); $\beta_2 = 2.0$ was significantly different from $\beta_2 > 3.0$ ($p < 0.013$). Values of $\beta_2$ greater than 2.5 were significantly different from all other values of $\beta_2$ greater than 2.5 ($p < 0.037$), with the exception of $\beta_2 = 4.5$ and 5.0 ($p = 0.208$). Any pairs unmentioned did not achieve statistical significance at the 5% level ($p > 0.089$).

Three maxima with 99.91% registration accuracy occurred in this test. The first two occurred at $\beta_1 = 1.5$ and $\beta_2 = 1.5$ and 2.0, respectively. The third maximum occurred at a $\beta_1$ value of 2.0 and a $\beta_2$ value of 2.0.

**C.3.2 A second local maximum: $\alpha_1 = 0.15$, $\alpha_1 = 0.85$**

In Section C.2.2, another local maximum with 99.91% registration accuracy was found at $\alpha_1 = 0.15$. In this test, $\alpha_1$ was held at 0.15 while $\alpha_2$ was held at 0.85. $\beta_1$ was varied from 1 to 3 in steps of 0.5 and $\beta_2$ was independently varied from 1 to 5 in steps of 0.5. Results are found in Figure C.5.

Instead of a sloping trend, excellent accuracies were found for mid-range values of both $\beta_1$ and $\beta_2$. A similar pattern of statistical significance arose as in the previous section. $\beta_1 = 2.5$ was significantly different from all other $\beta_1 < 2.5$ values ($p < 0.032$). $\beta_1 = 3.0$ achieved statistical significance with $\beta_1$ values less than 2.5 ($p < 0.013$). All other $\beta_1$ pairs were not significantly different from each other ($p > 0.342$).

As $\beta_2$ changes, accuracies did become statistically different from one another. $\beta_2 = 4.0$ was significantly different from $\beta_2 > 3.0$ ($p < 0.021$); $\beta_2 = 4.5$ was significantly different from $\beta_2 > 2.0$ ($p < 0.026$); $\beta_2 = 5.0$ was significantly different from $\beta_2 > 2.0$ ($p < 0.003$). The transition from $\beta_2 = 1.5$ to 3.0 was also significant ($p = 0.050$). All other $\beta_2$ pairs did not achieve statistical significance ($p > 0.077$).
APPENDIX C. SENSITIVITY TO PARAMETRIC VARIATIONS

Figure C.5: Registration accuracy results over different values of $\beta_1$ and $\beta_2$. $\alpha_1$ has been held at 0.15 and $\alpha_2$ has been held at 0.85. Accuracy results from all nine clinical bead sets were averaged to generate each data point.

(a) Registration accuracy over a range of distance cost exponents ($1 < \beta_1 < 3$) and angular cost exponents ($1 < \beta_2 < 5$) values.

(b) Standard deviation of the sample for the registration accuracies in (a).
In this test, one maximum achieved 100% registration accuracy, occurring at $\beta_1 = 1.0, \beta_2 = 3.0$. Four datapoints achieved 99.91% accuracy, at $\beta_1 = 1.5, 1.5, 2.0$ and $\beta_2 = 3.0, 2.5, 1.5$ respectively.

C.3.3 The original hypothesis: $\alpha_1 = 0.0656, \alpha_1 = 0.9344$

In Section C.2.3, a local maximum was found at $\alpha_1 = 0.06$ and $\beta_1 = 3$. This value, near the $\alpha_1$ hypothesis value of 0.0656, suggests that exploring the ratio of $\beta$ values near the hypothesis will be fruitful. In this test, $\alpha_1$ was held at 0.0656 while $\alpha_2$ was held at 0.9344. $\beta_1$ was varied from 1 to 3 in steps of 0.5 and $\beta_2$ was independently varied from 1 to 5 in steps of 0.5. Results are found in Figure C.6.

The accuracies in the graph are consistently high, with a slight trend towards higher accuracies at low values of $\beta_1$ and high values of $\beta_2$. There was only one significant difference amongst $\beta_1$ values: $\beta_1 = 2.5$ and $\beta_1 = 3.0$ were significantly different ($p = 0.005$). All other values did not achieve statistical significance ($p > 0.218$).

Changes in $\beta_2$ produced a few more statistically significant accuracies. $\beta_2 = 3.5$ was significantly different from $\beta_2 \leq 2.0$ ($p < 0.036$); $\beta_2 = 1.0$ was significantly different from $\beta_2 = 4.0$ ($p = 0.049$); $\beta_2 = 4.0$ was significantly different from $\beta_2 = 5.0$ ($p = 0.024$). All other $\beta_2$ pairs did not achieve statistical significance ($p > 0.054$).

In this test, two maxima achieved 100% registration accuracy, occurring when $\beta_2 = 4.0$ and $\beta_1 = 1.0, 1.5$. All $\beta$ values except one produced a registration accuracy over 97%. In addition, 35% of the $\beta$ pairs tested produced an accuracy above 99%.
APPENDIX C. SENSITIVITY TO PARAMETRIC VARIATIONS

1. Registration accuracy over a range of distance cost exponents ($1 < \beta_1 < 3$) and angular cost exponents ($1 < \beta_2 < 5$) values.

2. Standard deviation of the sample for the registration accuracies in (a).

Figure C.6: Registration accuracy results over different values of $\beta_1$ and $\beta_2$. $\alpha_1$ has been held at 0.0656 and $\alpha_2$ has been held at 0.9344. Accuracy results from all nine clinical bead sets were averaged to generate each datapoint.
C.4 Implications

The four dimensional parameter space, while difficult to comprehensively analyze, did appear to be well behaved. Since accuracy relies on a discrete number of assignments, accuracy values are discontinuous. Small changes in parameters do however produce small changes in the algorithm’s accuracy, making the space amenable to a simulated annealing algorithm or manual optimization.

In absolute terms, the algorithm has been shown highly effective on nine clinical datasets over a variety of parameters. Large areas of the parameter space produced registration accuracies greater than 97%, and datapoints exceeding 99% accuracy were easily found.

In future, an initial parameter selection near 0.0656 could be selected, with additional tuning on a representative sample of clinical bead sets.