Visual Tracking of Surgical Tools in Retinal Surgery using Particle Filtering

Mentor: Dr. Rogerio Richa

Group 14
William Yang and David Li
Computer Integrated Surgery II

5/10/2012

Johns Hopkins University

Background

Vitreoretinal surgery is undertaken to treat eye problems involving the retina, macula, and vitreous fluid and is considered one of the most difficult types of surgeries to perform. Some eye problems treated by vitreoretinal surgery include macular degeneration, retinal detachment, and diabetic retinopathy. During vitreoretinal surgery, small implements are inserted into the ocular vitreous cavity through small incisions and are manipulated within through a microscope to perform the procedure. Potential complicating factors include difficult visualization of surgical targets, poor ergonomics, lack of tactile feedback, and the requirement for high precision and accuracy. Since the surgery is performed using indirect visualization, the surgeon faces limited field and clarity of view, depth perception, and illumination, which hinders identification and localization of surgical targets and leads to long operating times and risks of surgical error.[5]

Problem

While there are many tools available and in development to help surgeons with hand tremor, such as the microsurgical robot, intraoperative force transduction sensors, and intraoperative optical coherence tomography (OCT) retinal scans[1], there has not been a method of detecting and tracking the surgical tool implements from the viewpoint of the optical microscope that is robust and precise enough for practical use in a clinical scenario.[4] This is beneficial not only for direct practical use by surgeons, but is also important for visualization applications such as intraoperative OCT, where the position of the detecting tip is required for complete understanding of the data collected.

Current methods of dealing with this issue range from gradient-descent algorithms using SSD and MI to manual observation. As the primary operator, the surgeon desires a solution to this problem that uses current equipment, is noninvasive, and is easy to use without being unwieldy. As the purchaser, the hospital desires a solution to this problem that is easy and fast to implement and is inexpensive.

Project Goal

We have developed and characterized a direct visual tracking method for retinal surgical tools using mutual information and particle filtering.

Relevance of our Approach

In determining the location of the tool during surgery, we have used template-based registration with Mutual Information (MI) as the similarity measure. We do this instead of using the Sum of Square Differences (SSD) or the Normalized Cross Correlation (NCC) as the similarity measure as MI is more robust in presence of changes in illumination, rotation, scale, and limited texture information. As we will show, mutual information with the proper templates perform significantly better than SSD or NCC as a similarity measure.

Also known as a condensation algorithm, we have used a set of particle filters to track the motion of the tool. Gradient descent methods, which are traditionally used, suffer from problems with local minima which occur when there are large displacements in the tool position. In addition, a particle filter supports alternative hypothesis tracking, so that it is unlikely an incorrect hypothesis will become “sticky.”

We will show that our tracking method is robust, supports alternative hypotheses, and shows tracking capability on par with current tool tracking methods today.

Technical Approach

Composite Dual Particle Filter

Two particle filters were implemented in tandem and iterated consecutively. One particle filter has three dimensions: x-position (X), y-position (Y), and angle (θ). That particle filter detects the position and angle of the shaft. The other particle filter has one dimension: z-position (Z). This particle filter is dependent on the position of the first particle filter and detects the position of the tip. The z-position is the distance up and down on the angle of the shaft from the x-position and y-position specified by the shaft particle filter.

The position (X,Y) determines the position of the upper right corner of the image the shaft template is being compared against. The tip is centered along the axes of the middle of the shaft template, and has the estimated tip at the halfway point. The position of the tip (XTip, YTip) is then specified using the following formulae.

$$\mathrm{XTip} = \cos\theta * \left(Z + \frac{shaftTemplateRows}{2} + \frac{tipTemplateCols}{2}\right) + X$$

$$\mathrm{YTip} = \sin\theta * \left(Z - \frac{shaftTemplateRows}{2} + \frac{tipTemplateCols}{2}\right) + Y$$

Figure 1: Templates for the particle filters (A) The template for the shaft used in the shaft particle filter (B) The template for the tip used in the tip particle filter. Both are not to scale; the tip template should be as tall as the shaft in the shaft template.

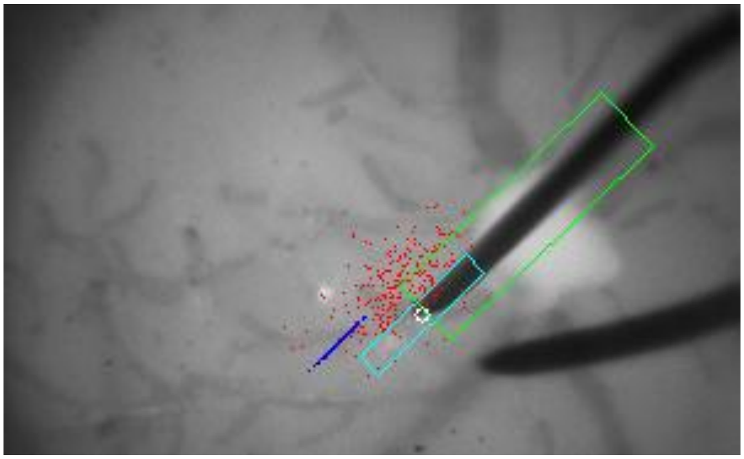


Figure 2: Depiction of the internal mechanisms of the dual particle filters during tracking of the tool tip in a retinal surgical image. (Green) The position of the hypothesis for the shaft particle tracker (Cyan) The position of the hypothesis for the tip particle tracker. (Red) The particles dispersed from the shaft particle tracker (Blue) The particles dispersed from the tip particle tracker. (White) The estimated tip of the tool

In addition, the shaft particle filter and the tip particle filter were used to update each other on their positions. This is used so that whenever the shaft particle filter is not tracking the shaft, the tip particle filter is turned off and the shaft particle filter is reinitialized with a larger delta. Alternatively, when the shaft particle filter is tracking the shaft, the tip particle filter is turned on and the shaft particle filter is reinitialized with a smaller delta. Finally, for every 20 frames, the shaft particle filter is reinitialized to the center of the tip particle filter. In other words, for each iteration that the composite dual particle filter is called for, the following processes happen.

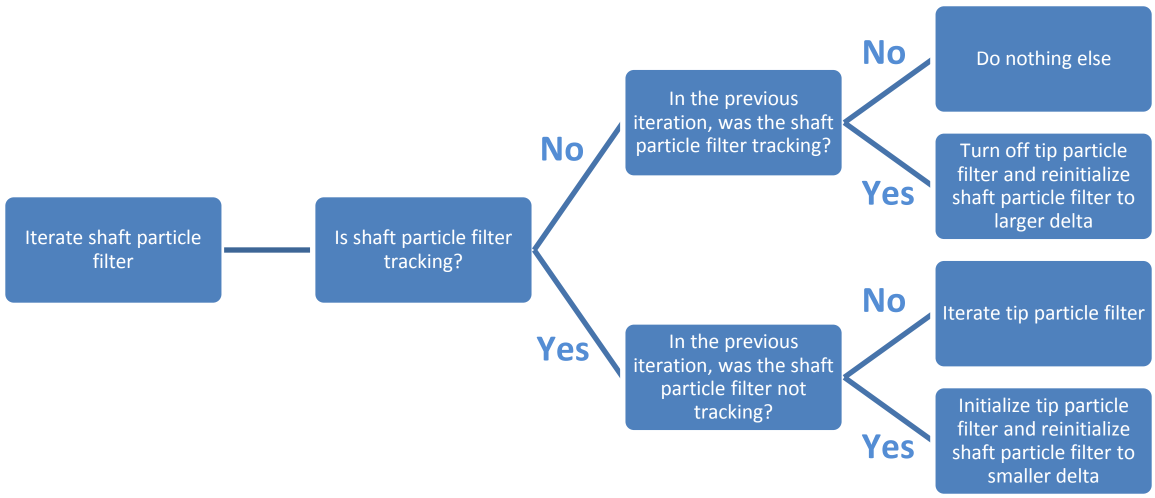


Figure 3: Schematic of the control processes behind the complete dual particle filter mechanism.

Particle Filter

The particle filter is essentially a two-step iterative process. With an initial constrained random sampling of particles around the last known location of the tool, the MI score is calculated for each particle. The MI score is thus the weight of the particle. The field of particles is then resampled so that each particle has equal weight, but the density of particles is proportional to the calculated MI score. The process is then iterated, and the location of the tool is determined based on the average of the particles weighted by their similarity score. Finally, the determination of whether the particle filter is tracking successfully or not is determined by obtaining the position of the most likely hypothesis, calculating the MI of that position against the template, and comparing that to a user-defined threshold. This threshold is constant for each template and similarity measure, but needs to be readjusted manually for each alteration in either template or similarity measure. Automation of this process is another potential venue for future innovation.

Mutual Information

Before computation of the MI similarity measure is performed, an intensity co-histogram between the template image and the image is calculated. To improve efficacy and efficiency of the similarity measure, the co-histogram was computed using 8 bins as determined by previous literature.[2] The sums of probabilities for both joint and individual intensity histograms was computed using simple summation. The calculation of the MI score involves calculating the joint entropy between the template (I*) and image (I), then subtracting the images' individual entropies:

$$h(I) = -\mathrm{sum}_r[p_I(r) \log(p_I(r))]$$

$$h(I, I*) = -\mathrm{sum}_{r,t}[p_{II*}(r, t) * \log(p_{II*}(r, t))]$$

$$MI(I, I*) = h(I)+h(I*)-h(I, I*)$$

MI is effectively a measure of the quantity of shared information between the two images being compared.

Results

Qualitative Comparison

Qualitative comparison was performed to distinguish the tracking capabilities of particle filters with either SSD, NCC, or MI as similarity measures. This was performed using two sample videos: one of a computer vision book translating and rotating along a multicolored background and another of a vitreoretinal surgical procedure.

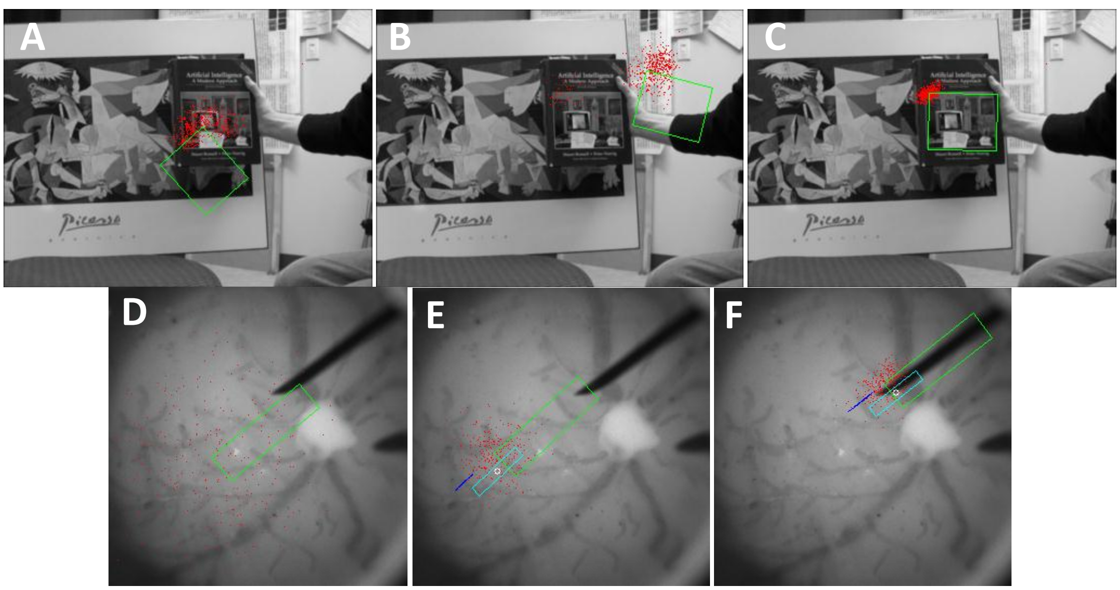


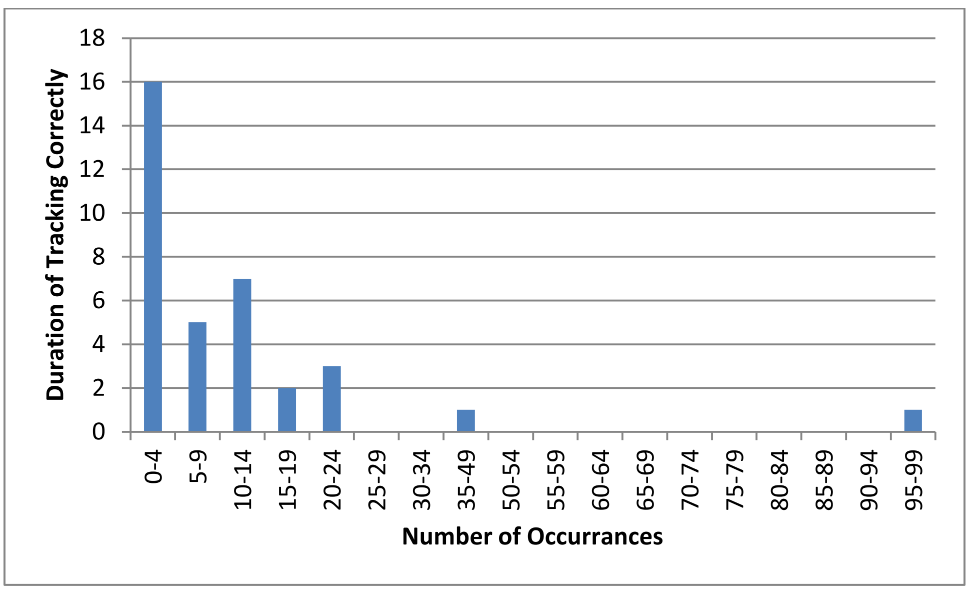
Figure 4: Frames from the visual tool tracking method using (A-C) a video of a book rotating and translating and (D-F) a video of a vitreoretinal surgery, using (A,D) SSD, (B,E) NCC, and (C,F) MI as similarity measures.

In both cases, MI performed very well, tracking the book or tool tip in a reliable manner. In contrast, SSD and NCC did not perform well at all, losing the book or tool tip within minutes and being drawn to various alternative hypotheses on the background. This is due to the reliance of SSD and NCC on near-constant illumination, rotation, scale, and texture information. In contrast, mutual information relies on the joint distribution of intensities, allowing for comparisons between multi-modal images such as the templates used for MI and the actual image. This robustness also allows us to use more idealized depictions of the templates instead of photo-realistic ones, which are impermeable to changes in intensity and texture information.

Error Analysis

To measure error analysis, we will use the first fully annotated and freely available image data set for tool detection in *in vivo* retinal microsurgery.[4] Error analysis will consist of two parts: tool detection using mutual information and tool tracking using particle filtering.

To analyze the efficacy of particle filtering as a tool tracking algorithm for vitreoretinal surgery, the complete particle filtering with mutual information algorithm will be evaluated on video sequences. At any frame, the tracking is said to have failed whenever the true position of the terminating end of the shaft of the tool is greater than some threshold σ. Whenever that happens, a note is made and the tracking algorithm is reinitialized using ground truth to continue analysis. After the sequence is complete, the number of frames successfully tracked continuously is plotted against the number of events as a histogram and is fitted to a geometric distribution. The lower the fitted probability, the better the robustness; current gradient descent tracking algorithms using SSD or MI have a p-value of around 0.1, with a maximum around 0.2.[4]



While many instances of annotated data sets were not available due to logistics issues, ground truth for one data set was obtained through manual determination for each frame. Using this data set, it was possible to obtain a histogram of the number of frames successfully tracked continuously. It should be noted that in this instance, the tool tracking method was not reinitialized to ground truth whenever deviations from correct tracking was detected. In many instances, this was impossible, as the template for the shaft or tip could not fit into the upper right corner where the correct position of the tip was. [4]

Fitting this to an exponential distribution using the fitdist function in MATLAB, it was possible to obtain an estimated p-value of 0.3434. This is comparable to many current state-of-the-art algorithms today, though not significantly better. However, it should be noted that the method for determining this p-value is not the same as literature estimations for other methods; reinitialization to ground truth was not performed. This both demonstrates a higher tendency to avoid false hypotheses and hints at better performance than presently suggested.[4]

Reading List

1. Balicki, M., Han, J., Iordachita, I., Gehlbach, P., Handa, J., Taylor, R., and Kang, J. (2009). Single Fiber Optical Coherence Tomography Microsurgical Instruments for Computer and Robot-Assisted Retinal Surgery. *MICCAI 2009*, 108-115
2. Dame, A. and Marchand, E. (2010). Accurate real-time tracking using mutual information. *IEEE Int. Symp. on Mixed and Augmented Reality*, *ISMAR'10*, 47-56.
3. Isard, M. and Blake, A. (1998). Condensation – conditional density propagation for visual tracking. *Int. Journal of Computer Vision*, 29, 5-28.
4. Richa, R. et al. (2012). An Evaluation Framework for in vivo Microretinal Tool Detection and Tracking. *MICCAI*
5. Richa, R. et al. (2012). Hybrid SLAM for Intra-operative Information Augmentation in Retinal Surgery. *MICCAI*
6. R. Shams and N. Barnes. Speeding up mutual information computation using NVIDIA CUDA hardware. *Digital Image Computing Techniques and Applications*, pages 555–560, 2007.

Management Summary

Division of Labor

	Initial Particle Filter	Mutual Information	Porting Code to CISST	Porting Code to CUDA	Restructuring Code	Particle Filter for Tip	Pairing Particle Filters	Optimization and Comments	Error Analysis
William Yang	X		X	X	X			X	
David Li		X				X	X	X	X

In all cases, the primary coder codes the initial implementation while the partner checks the code for errors and suggests improvements and optimizations. Both group members are responsible for on-going documentation and preparing demos. David wrote the main draft of the paper, as well as the poster presentations, while William revised and gave advice.

Future Work

Working Parallelization

Ideally, the MI score calculation stage of the particle filter would be run in parallel for each particle. This would enable the particle filter to be very robust without sacrificing the performance and frame rate needed to perform a live surgery; without massively parallel computational capacity, the number and density of particles would need to be reduced in order to allow the tracker to run at an acceptable frame rate, which in turn reduces the robustness and accuracy of the tracker. However, although parallelization was achieved using OpenMP, an extension of OpenCV, further parallelization was not achieved.

In addition, previous papers have proposed the possibility of calculating MI itself in parallel, within each particle. Though this will most likely not affect frame rate to as high of a degree as the above, it is still something to look into and possibly implement.[6]

GPU

GPU implementation is an extension of parallelization and would greatly improve the efficiency of our tool tracking method; however, since a working parallelization was not achieved, GPU implementation was not as well. Once a working parallelization is achieved, GPU implementation should follow.

Error Analysis

Ideally, an accurate quantifiable method for determining the performance of a tool tracking method will be to initialize the tracker on multiple videos and determine the time it takes for the tool tracking method to leave the ground truth for some threshold σ. A histogram can then be generated from the multiple intervals it takes for the tool tracking method to diverge from the ground truth. As this can be represented as a geometric distribution, the histogram can be fitted to a exponential distribution and the p-value calculated. Once multiple retinal surgical videos with ground truth can be collected, this analysis should be easy.

Dealing with Specularity, Background Features, and Shadows

Many of the alternative hypotheses that the current iteration of the tool tracking method gets stuck on are due to either specularities on the metal surface of the tool tip or background features on the retinal surface. To deal with these issues, segregation of the tool tip using disparity maps has been proposed. It may be possible to combine features from disparity mapping and mutual information to obtain a better response.

Finally, a major alternative hypotheses that has not been resolved is the shadow of the tool tip itself. Because the shadow of the tool tip appears almost identical to the actual tool tip, mutual information cannot detect a significant difference between the two. In addition, the variation in mutual information calculated as the shaft tracker moves from the actual shaft to the shaft shadow is within normal variation while just tracking the shaft, meaning that thresholding or modes cannot be used to solve this issue. Again, disparity maps may be a great help in this problem, as the tool shadow is always projected on the retinal surface behind the tool tip.

Deliverables

We achieved our maximum deliverable, which was successful parallelization of the particle filter. However, since we added further improvements such as pairing of two particle filters and tip centering, as well as threshold determination of tracking at around the midpoint of our project, we were forced to take parallelization down in favor of these added improvements. Currently, parallelization is not implemented, though our maximum deliverable has been achieved.

Lessons Learned

We learned that installing OpenCV, CISST, and CUDA is very important. One of the largest problems we had was that David was unable to install CISST and CUDA on his computer. Because of that, David worked primarily on OpenCV code and old video files, as .avi files can't be run on OpenCV code very effectively. Related to this is the fact that optimization of code for

CISST and parallelization should be a top priority, as we had to restructure a great deal of code to support porting.

We also learned that while particle filtering and mutual information is robust, there are many areas which can be improved on our initial algorithm. Such improvements, such as tracking the tip and linking two particle filters together, as well as determining thresholding for whether tracking is being performed or not, have greatly improved the efficacy of our tracking method.

Dependencies

- Development environment for Milestones 1 and 2
  - Resolved (Visual Studio/OpenCV)
- Development environment for Milestones 3 and 4
  - Will work with Rogerio (CISST libraries)
- Access to CUDA-enabled GPU for Milestone 5
  - Resolved for offline development; will work with Rogerio for online testing
- J-Card access to robotorium
  - Resolved
- Use of microretinal surgery workstation
  - Worked with pre-recorded data
  - Scheduled time for workstation and worked on workstation

Management Plan

We have held weekly meetings with Dr. Richa every Wednesday. The project status and timeline was reassessed every week. Programming and peer code review proceeded on a continuous basis and is subject to source code revision control. In-person or electronic meetings were held as needed in order to discuss approaches and test code. Documentation of code has been on-going.