Title

IMPROVING IMAGING BY USING GENETIC ALGORITHMS TO FIND ARBITRARY PROFILES OF BESSEL BEAM LASER

By

Leon Hunter

Project Advisor

Dr. Tomasz Smolinski[1], Dr. Thomas Planchon[2]

Date

December 2015

Abstract

Adaptive optics is a process used to enhance the performance of an optical system by reducing the effect of wave front distortions using an active optic (deformable mirror or spatial light modulator SLM). One approach to achieving such an enhancement is the use of genetic algorithms to control the optimization of the active optics. The goal of the project was to develop a genetic algorithm to later control a spatial light modulator (SLM) used for shaping the profile of a laser beam. A spatial light modulator is a device capable of manipulating amplitude, phase, or polarization of light waves in space and time. Naturally, the shape of a laser follows a Gaussian function. There is a lot of interest in using shapes other than Gaussian to improve the techniques of measurement for cymatic devices. This talk will present the results of an experiment performed using a genetic algorithm used to optimize a SLM. Various measurements of resulting laser profiles from our simulated experiments will be presented.

Background

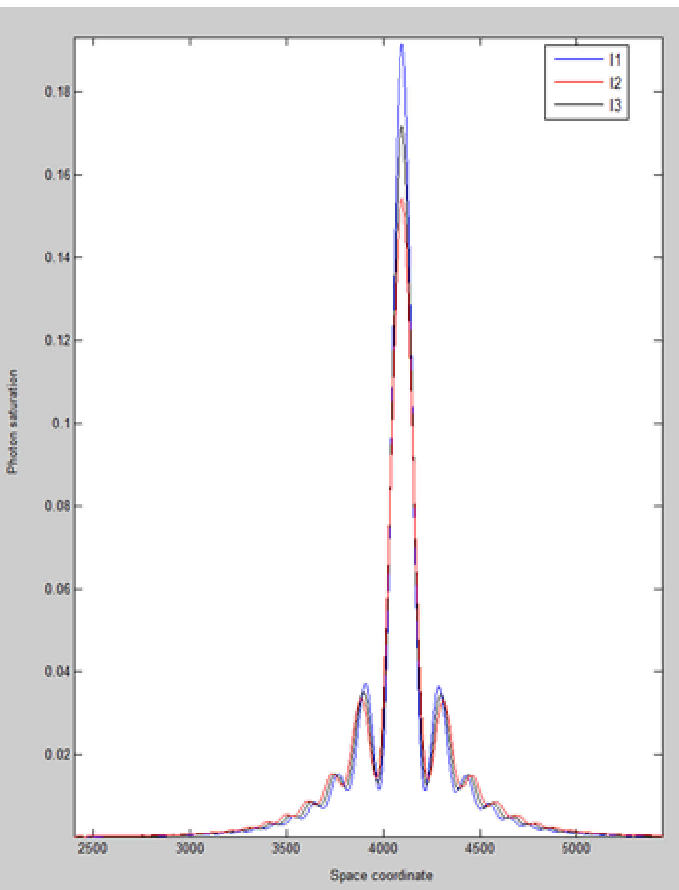
In the Biophotonic Imaging Laboratory, nematodes named "Caenorhabditis elegans" (C. elegans) are imaged for tissue samples and cellular analysis. The current imaging technique uses a Guassian emission of light. With this technique, a saturation of photons is beamed at the focal point of the light being emitted. This high intensity emission can induce biological samples with phototoxicity, and subsequently destroy the specimen and resulting image.

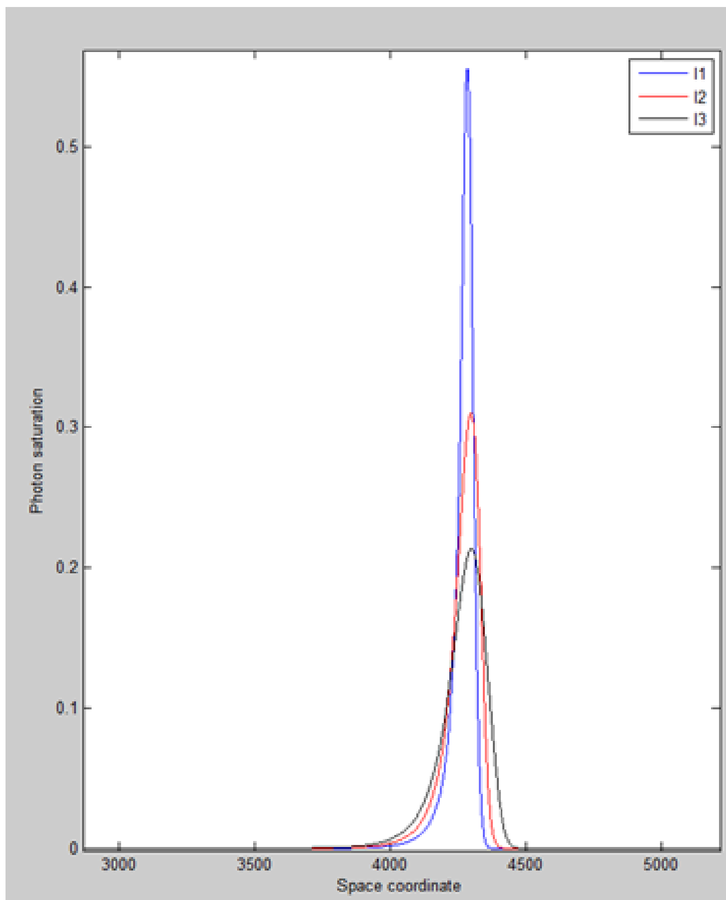
Objective

The spatial light modulator is the active optic used shape the profile of the laser beam. The objective is to optimize the shape of our SLM such that the measured planes before the focal point, at the focal point, and after the focal point, all have similar profiles.

Determining a good solution

The similarities of the profiles are determined by comparing each plane with every other plane. The first plane is compared to the second and last. Then, the second plane is compared to the last. In the most ideal case, the shape of the SLM will emit a beam whose planes are identical in terms of absolute difference. The images below are plots of two candidate solutions. On the left, three functions are super imposed on a single figure. It is visibly clear that their profiles are very similar. On the right, another candidate solution is plotted. This figure is an example of a bad solution as the profile of the planes do not share a strong likeness of each other.





Methodology

Evolutionary algorithm was the method used to optimize the shape of the SLM. The makings of Genetic Algorithms can be summarized as *parents*[1], *reproduction*[2], *offspring*[3], and *selection*[4]. **Parents** (chromosomes) contain properties (genes) to be reproduced and passed to offspring who may or may not survive depending on how well-suited they are for their environment (problem). **Reproduction** (hybridization) is the process of parents' genetic information being copied, mixed, and mutated. **Offspring** are the resulting entities whose properties are yielded by reproduction. **Selection** is the action(s) that decide which offspring are most suited for parenting. Selection is determined by the definition of the fitness function in a genetic algorithm. A fitness function evaluates the goodness of a parent (genetic solution).

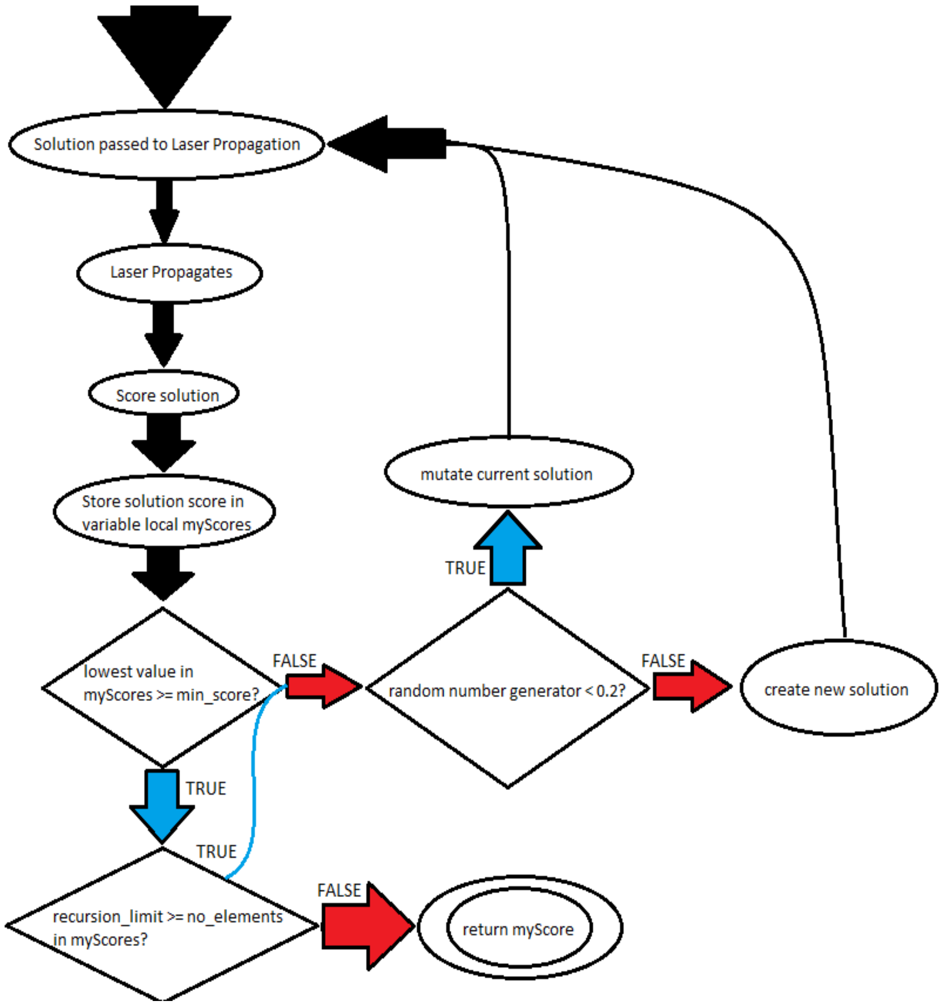
Approach

In the process of developing the code, 6 approaches were used. Each approach generates a polynomial function that is used to describe the shape of the SLM. Two approaches do not make use of MatLab's Global Optimization toolbox. Four approaches do. Of the four using the toolbox, two use a custom mutation function embedded in the fitness function.

The first, is Technique A. Technique A passes an arbitrary but symmetrically distributed polynomial function to the SLM. The second is Technique B. Technique B is like Technique A, but it does not impose symmetry on the function being passed. The third is Technique A0 which is similar to technique A, but it uses the toolbox. The fourth is Technique A1 which is comparable technique A0, but it uses the custom mutation method. The fifth is Technique B0 which is related technique B, but it uses the toolbox. The final is Technique B1, which resembles technique B0, but it uses the custom mutation method.

The matrix below is a breakdown of the techniques and their definitions.

Technique	**Toolbox**	**Custom Mutation**	**Symmetry Imposed**
A	no	no	yes
B	no	no	no
A0	yes	no	yes
A1	yes	yes	yes
B0	yes	no	no
B1	yes	yes	no



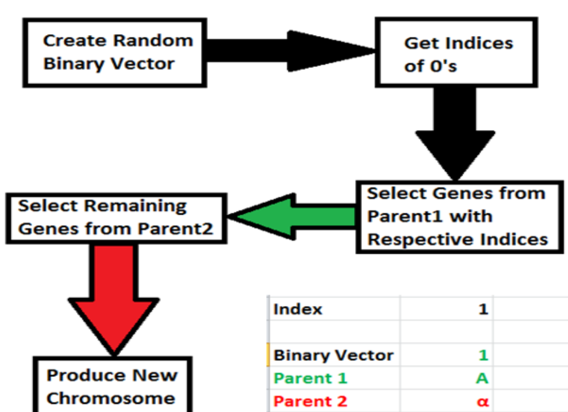
Custom Mutation Method

This finite state machine describes how the custom mutation method works. First, the solution is passed to the Laser Propagation. The laser propagates and returns a score. The score is stored in a vector called local_scores. If the lowest value in local_scores is greater than or equal to the minimum score allowed, then check if the recursion limit is greater than or equal to the vector length. If it is, then there is a 20% chance of creating a new solution and an 80% chance of mutating current solution. Then, pass the new entity, along with the vector, to the laser propagation and repeat the process. If recursion limit is less than vector length, or the lowest value in local scores is less than minimum scored allowed, then return the minimum value (best score) in the vector.

The goal of this custom mutation method was to weed out solutions with very poor fitness scores. Intuitively, solutions with very bad scores must provide no value to the gene pool. However, after further analysis, it was observed that this did not always hold true. While an individual may not perform well against its environment, it still may contain some useful genetic information. Perhaps a gene is too prominent, or may it's so obscure (recessive) it seems to be nonexistent.

Local Optima

Nonetheless, it's these kinds of extreme fluctuations that help diversify a population. As diversity decreases, the likelihood of becoming stuck in a local optima increases. Local optima are relative best solutions within a neighbor solution set. Because of this, the algorithm may never achieve a global optima because its solutions seem to be ideal relative to other nearby solutions. Just by observing the image here, you can see how significant a single mutation can be to a population.



Index	1	2	3	4	5	6
Binary Vector	1	1	0	1	0	0
Parent 1	A	B	C	D	E	F
Parent 2	α	β	π	δ	ε	Φ
Offspring	A	B	π	D	ε	Φ

Reproduction

In a genetic algorithm, the crossover functions describe how two individuals combine information to yield offspring. For techniques not using the toolbox, new solutions combine with the better half of parents by adding or subtracting values from respective genes. For techniques using the toolbox, the crossover function uses a random binary scatter approach. First, a random binary vector is created. Then, the indices of its 0's are retrieved. After, genes from parent 1 with the same indices that have been retrieved are selected. Next, the remaining indices are matched with the genes from parent 2. Finally, a new chromosome is produced by merging the two sets of selected genes. The spreadsheet and flow chart wrapped within this paragraph describe how the toolbox's mutation method works.

Results

Each experiment was tested with 20 members, 20 genes, and a phase range of 100. It should be noted that the issue of solutions becoming stagnant in a local optima was typically not a problem for techniques using the custom mutation method. Though it did reduce the speed of execution, on average it improved the fitness scores. The charts below display the data for the corresponding methods used.

Technique A

	Trial	Min	Max	Average	Generations	Time
	1	2.4797	100	31.7926	25	2.6109
	2	1.9334	100	27.5218	50	4.2567
	3	1.6514	100	26.9806	75	6.3695
	4	1.2419	100	25.7951	100	8.6315
	5	1.1885	100	25.7734	125	10.8114
	6	0.7134	100	25.2715	150	12.9762
	7	1.0372	100	25.8997	175	15.0732
	8	0.7638	100	25.1863	200	17.0704
	9	0.8813	100	25.433	225	19.389
Average		1.3212	100	26.62822222	125	10.7988

Technique B

	Trial	Min	Max	Average	Generations	Time
	1	3.2392	100	31.2922	25	2.1514
	2	2.2396	100	29.1991	50	4.3428
	3	1.6779	100	27.7104	75	6.4864
	4	1.2932	100	25.4516	100	8.6214
	5	1.2706	100	26.3621	125	10.7345
	6	0.7535	100	25.4742	150	12.6401
	7	0.5646	100	24.8328	175	15.1333
	8	0.6197	100	25.5518	200	16.8736
	9	0.7447	100	24.5157	225	19.0277
Average		1.37811111	100	26.7099888	125	10.6679111

Technique A0

	Trial	Min	Max	Average	Generations	Time
	1	2.375096029	26.18608414	11.24207286	25	4.546429011
	2	2.447703888	25.4941616	7.809777662	50	7.562364029
	3	1.919184162	8.670991649	3.648313263	75	10.21557294
	4	6.033638411	15.70662572	9.745356333	100	12.92071264
	5	1.506364999	10.4017603	3.359089085	125	15.63538327
	6	1.685735516	13.73349819	5.849613189	150	18.5147353
	7	1.315223208	25.58476981	8.339630841	175	21.01185397
	8	1.56685733	15.56517137	11.2724216	200	24.10182537
	9	2.893241065	12.63153565	5.63366385	225	26.94676188
Average		2.415893845	17.10828871	7.43332652	125	15.71729316

Technique B0

	Trial	Min	Max	Average	Generations	Time
	1	2.396517512	22.08539149	11.62266273	25	11.57976782
	2	6.595947103	27.69994395	16.48966686	50	16.8130631
	3	1.528976548	20.55319045	10.59278772	75	22.26252037
	4	5.526031235	24.15033437	14.50807493	100	27.63766115
	5	0.778762308	17.96877927	3.958468798	125	32.77318135
	6	0.976371167	23.34985832	6.600583403	150	38.81865665
	7	1.376445656	7.364492046	3.175170996	175	43.89494256
	8	1.193198043	2.823644761	1.75171061	200	49.27092991
	9	0.782907205	12.23159538	6.284580969	225	55.32165521
Average		2.350572975	17.58080334	8.331523002	125	33.15248646

Technique A1

	Trial	Min	Max	Average	Generations	Time
	1	3.275089079	12.79587342	7.971924274	25	15.99948975
	2	1.424477025	10.38600148	3.327231195	50	27.40438605
	3	2.834552521	13.14058917	7.681852234	75	44.2654757
	4	2.152144719	12.59280853	5.519030073	100	58.42645849
	5	1.480320228	3.769585004	2.450107909	125	66.82004038
	6	1.69393666	11.98393381	3.781654041	150	84.23749116
	7	1.396036017	11.87146934	4.702425996	175	95.18521732
	8	1.718166553	12.81087339	5.332014958	200	107.0010916
	9	6.636606149	11.53052264	8.90337982	225	121.8851728
Average		2.512369883	11.20907298	5.518846722	125	69.02498037

Technique B1

	Trial	Min	Max	Average	Generations	Time
	1	3.249269076	15.17835798	6.756562602	25	31.36025268
	2	1.901733254	9.354598562	5.112369528	50	58.35341138
	3	1.689522637	16.04906662	6.968447009	75	89.36450574
	4	1.03646409	8.889509481	2.943115064	100	114.8213722
	5	1.203097873	16.54777272	4.4815786	125	137.2877485
	6	1.995608936	11.39444139	4.681090168	150	172.5121384
	7	1.33774664	7.128371596	2.660799007	175	167.0930122
	8	1.641259836	11.44810944	3.790804407	200	209.7839115
	9	1.680538617	10.37701517	5.893703919	225	219.3592058
Average		1.748360106	11.81858255	4.809830034	125	133.3261732

Conclusion

The scoring method of the no toolbox projects are fallible. If the minimum intensity of a plane is less than 0.2, then it is given the score of 100. This causes the calculated mean values to become inflated, making it difficult to identify why the values are so large on average without background on the internal mechanics of the fitness function. It also fails to discern the difference between a useful solution with low intensity, and one that is not useful; they are both given the same grade. Instead, the method for evaluating fitness should be normalized. However, the projects not using the toolbox seem to be very capable of producing viable solutions. In fact, according to the data, the best fitness scores from all the populations come from the two not using the toolbox. These two solutions converge more quickly on average. One possible explanation is that using the toolbox is more computationally intensive as it tries to optimize the problem. Also, there is the possibility that a default option in the GA has been overlooked and is interfering with speed and accuracy of convergence.

The chart below summarizes the data as it displays the averages of each column from each respective technique-chart.

Toolbox	Technique	Min	Max	Average	Time
No	A	1.321177778	100	26.62822222	10.79875556
No	B	1.378111111	100	26.70998889	10.66791111
Yes	A0	2.415893845	17.10828871	7.43332652	15.71729316
Yes	B0	2.350572975	17.58080334	8.331523002	33.15248646
yes	A1	2.512369883	11.20907298	5.518846722	69.02498037
yes	B1	1.748360106	11.81858255	4.809830034	133.3261732

Future Work

In the future, I would like this project to interface with real hardware rather than simulated. This way, we can establish the feasibility of these methods. After demonstrating proof of concept, the solution can be applied to various fields of science faced with similar cymatic issues.