Towards Fully Automated Phototransfection

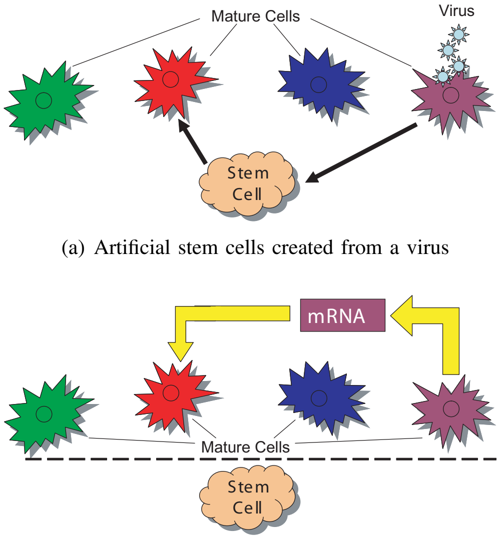
David J. Cappelleri, Ádám Halász, Jai-Yoon Sul, Tae Kyung Kim, James Eberwine and Vijay Kumar

***Abstract*— Flexible automation technologies have been applied to the manual phototransfection procedure on fibroblast and astrocyte cells. We have designed and implemented a framework for increased throughput of the entire process. Integrated image processing, laser target position calculation, and stage movements show a throughput increase of $> 23X$ over the current manual method while the potential for even greater throughput improvements ($> 110X$) is described. A software tool for automated single cell morphological measurements has also been constructed and shown to be able quantify changes in the cell before and after the process, successfully characterizing them, using metrics such as cell perimeter, area, major and minor axis length, and eccentricity values.**

I. INTRODUCTION

There has been much research performed on stem cells in recent years [2], [12], [17]. This research is popular since these cells contain a blue-print of how to build everything in your body, as they are naturally produced during embryonic development. However, with the possibility of using stem cells in cell replacement therapies for various illnesses a more ready and less controversial source of stem cells has been sought. One approach is to create them artificially. This can be done by using viruses to deliver a set of transcription factor cDNAs into mature cells that will then dedifferentiate these cells into an induced-pluripotent stem cell (iPS) [16]. Signals can then be sent to this iPS cell to lead it down a desired developmental pathway to create specified cell types. This procedure is shown schematically in Figure 1(a). Creating therapeutically relevant cells in this manner suffers from the difficulty in programming stem cells to become a particular cell type. An alternative approach is the direct reprogramming of one cell type into another cell type using the Transcriptome Induced Phenotype Remodeling approach (TIPeR) [15] whereby populations of RNA are introduced into a host cell in an effort to reprogram that host cell. It attempts to wipe out the current instruction set that is in place in the host cell and replace it with another. A key feature of the TIPeR procedure is introducing the RNA population into the host cell. One method for performing TIPeR is through the use of phototransfection to transiently introduce holes into the host cell through which mRNA populations can diffuse. Once the holes reseal the introduced mRNA will be translated and produce functional proteins that can modify host cell phenotype. Phototransfection provides a means for performing functional genomics manipulations on individual cells. This process is pictured schematically in Figure 1(b).

The current, manual phototransfection procedure consists of the following steps:



(a) Artificial stem cells created from a virus

(b) Phototransfection process with donor mRNA

Fig. 1. Artificial stem cell creation vs. phototransfection

- Locate cells of interest on a marked cover-slip and record their locations on paper with same markings
- Identify the cytosol and edges of the cell of interest
- Define a region to apply a laser to poke a hole in the cell membrane, considering that firing the laser on dendrite (arm) portion of the cell or on the cytosol region of the cell will damage it.
- Locally apply mRNA of a donor cell to the target cell using pipette
- Observe cell changes at various time intervals (hours, days, weeks)

This process is very tedious the overall yield rate is 70-80%, meaning that the cells survive the process not that they are necessarily changing from one type of cell to another. The throughput is 20 cells/hour. The goal here is to apply flexible automation techniques in order to increase the throughput to about 360 cells/hour. This is important in order to rapidly explore many different amounts and types of donor mRNAs, perform various functional tests to see what genes have been expressed and to fill out microarrays for data analysis and fine tuning of the overall procedure. It is also desired to be able to quantify the cell morphology for comparisons before/after the process to use as one measure to verify that the cell is indeed changing from one type of cell to the other.

There is some related work on automated systems to improve the efficiency, productivity, quality, and reliability for procedures and processes in the life sciences. Applying microrobotic and flexible automation technologies to micromanipulation tasks such as single cell holding, moving, injecting/ejecting materials in/out of cells is becoming an active research area [7], [21]. These types of cell manipulation tasks are important for the characterization and manipulation

D. Cappelleri is with the Multo-Scale Robotics and Automation Lab, Department of Mechanical Engineering, Stevens Institute of Technology, Hoboken, NJ USA dcappell@stevens.edu

V. Kumar is with the GRASP Lab, Department of Mechanical Engineering, University of Pennsylvania, Philadelphia, PA USA kumar@grasp.upenn.edu

J. Sul. T. Kim, and J. Eberwine are with the PENN Genome Frontiers Institute, Dept. of Pharmacology, University of Pennsylvania, Philadelphia, PA USA {jysul, kit, eberwine}@mail.med.upenn.edu

A. Halász is with the Dept. of Mathematics, West Virginia University, Morgantown, WV USA halasz@math.wvu.edu

of single embryo cells in applications such as cloning, gene expression analysis, cell replacement therapy [14], intracytoplasmic sperm injection (ICSI) and embryo pronuclei DNA injection. Much work has been done on creating automated systems to increase the survival and success rates of these types of procedures [8], [10], [19], [20].

Various types of platforms for laboratory automation have also been presented. Choi *et al.* [6] present a robotic platform for clinical tests suitable for small or medium sized laboratories using mobile robots. A "tower-based configuration" for the automatic execution of various biotechnology (genomics and proteomics) protocols is presented in [13]. Studies to identify current and future approaches to the design of highly automated systems for life science processes involving humans in control loops in applications such as high-throughput compound screening and high-performance analytical chemistry, adherent cell culturing, and the cultivation of primary and stem cells have been explored in [9] and [11], respectively.

The work presented this paper describes automation for the life science application of phototransfection of single cells. A software analysis tool for automating cell morphological measurements for quantitative comparison of images of the cells before and after the process is described first. This is followed by a description of the framework for automating the current manual phototransfection process along with proof-of-concept implementation results. Finally, recommendations for further improvements are provided.

II. AUTOMATED CELL MORPHOLOGY MEASUREMENTS

Images of the phototransfected cell are observed and recorded before and after the process, at different time intervals, to assess morphological changes in the cell. Cell characterization with morphological measures is one way that biologists can assess the success of the overall procedure, along with other functional tests. However, this is not an easy task. The problem in comparing two different images of the same cell before and after phototransfection is that the changes in the cell are hard to discern because of changes in illumination, camera viewpoint and background in both images. Image segmentation techniques, borrowed from the computer vision literature, are used here to segment the image of the cell from the background in order to compare both images of the cell before and after the process without ambiguities. From a properly segmented image, the morphology is quantified by computing measures such as cell area, perimeter, major axis length, minor axis length, and eccentricity.

Initially, images of the cells were segmented using graph-theoretic clustering techniques, using the image pixels as nodes in the graph [1]. Once a connected, weighted graph is constructed from the image of interest, a graph-cutting algorithm can be executed in order to segment the image. Graph-cutting techniques tackle the minimum cut problem: finding a cut in the graph that has the minimum cost among all the cuts. The algorithm from Boykov and Kolmogorov [3], that is used here, solves this problem problem by finding the maximum flow from the "Source" nodes to the "Sink" nodes in the graph. The output of the algorithm is a label for each node in the graph (pixel in the image) assigned to be either the "Sink" or "Source". For this application, the "Sink" corresponds to pixels in the background of the image while the "Source" corresponds to pixels belonging to the cell. Edge weights between the nodes in the graph are computed using a weighted sum of distance (A_d), pixel intensity (A_i), and texture (A_t) affinity measures for particular nodes. The affinity measures between two nodes, N1 and N2, are listed in (1)-(3), while the corresponding edge weight, E, is given in (4)

$$
\begin{aligned}
A_{dN1,N2} &= exp\{-(P_{N1}-P_{N2})^2/(2\sigma_d^2)\} && (1)\\
A_{iN1,N2} &= exp\{-(I_{N1}-I_{N2})^2/(2\sigma_i^2)\} && (2)\\
A_{tN1,N2} &= exp\{-(T_{N1}-T_{N2})^2/(2\sigma_t^2)\} && (3)\\
E_{N1,N2} &= w_1A_d + w_2A_i + w_3A_t && (4)
\end{aligned}
$$

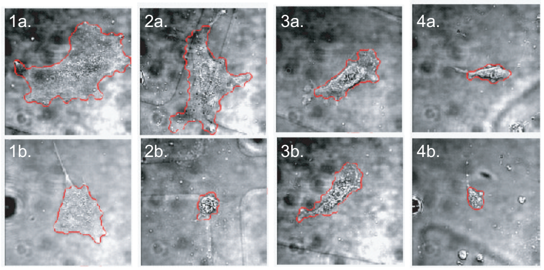
where P_N = position of node N, I_N = pixel intensity value of node N, T_N = average change in pixel value intensity between pixels in a image patch surrounding node N, and the σ parameters are chosen to yield large affinity values for similar pixels while yielding low affinity values for dissimilar pixels. The weights, w_1, w_2, and w_3, are user defined and each are ≤ 1 while their sum = 1.

Edge weights between the nodes in the graph and the "Sink" and "Source" nodes also need to be computed to complete the graph. Equations (5)-(11) are used for this. Here A_{dBkg}, A_{iBkg}, and A_{tBkg} are affinities associated with the background ("Sink") section of the image that are pre-computed from a set of training images.

$$
\begin{aligned}
A_{dAve} &= (A_{dN1,N2} + A_{dN1,N3})/2 && (5)\\
A_{iAve} &= (A_{iN1,N2} + A_{iN1,N3})/2 && (6)\\
A_{tAve} &= (A_{tN1,N2} + A_{tN1,N3})/2 && (7)\\
F_{N1} &= A_{dAve} + A_{iAve} + A_{tAve} && (8)\\
F_{Sink} &= A_{dBkg} + A_{iBkg} + A_{tBkg} && (9)\\
E_{N1,Sink} &= exp\{-(F_{N1}-F_{Sink})^2/(2\sigma_s^2)\} && (10)\\
E_{N1,Source} &= 1 - E_{N1,Sink} && (11)
\end{aligned}
$$

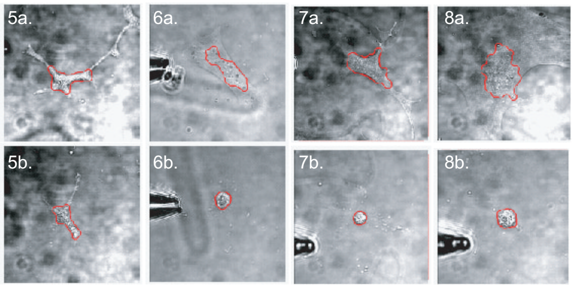
The raw output from the graph-cut algorithm needs to be filtered in order to come up with the final segmented image of the cell from the background. Image erosion and dialation steps are applied in Matlab® and the largest connected pixel region that is left is used as the segmented cell image and statistics reported on it. Figure 2(a) shows the result from this procedure on images of four fibroblast cells before and after the phototransfection process, with the segmented area overlayed on the original image. The images in the top row are before the process while the bottom row of images are after the process has been completed. The cell perimeter, area, major and minor axis, and eccentricity (in pixels) are calculated for each set of images and the corresponding changes in these morphological measures reported in the table below the images. These metrics show substantial changes after the phototransfection process has been performed. This indicates a successful phototransfection since the fibroblasts now are starting to look like the donor astrocyte cells and there are metrics to support this.

However, due to the inconsistencies in the lighting conditions, cover-slip markings, and textures of the backgrounds and cells in the images, consistent results for one set of system parameters across all data sets are difficult to achieve. Figure 2(b) shows examples of poor image segmentation, when only a subset region of at the actual cell is identified, using the same set of system parameters as those in Figure 2(a). Continuous tuning of the graph parameters can be performed to obtain acceptable results, however it is desired to keep these details transparent to the end-user



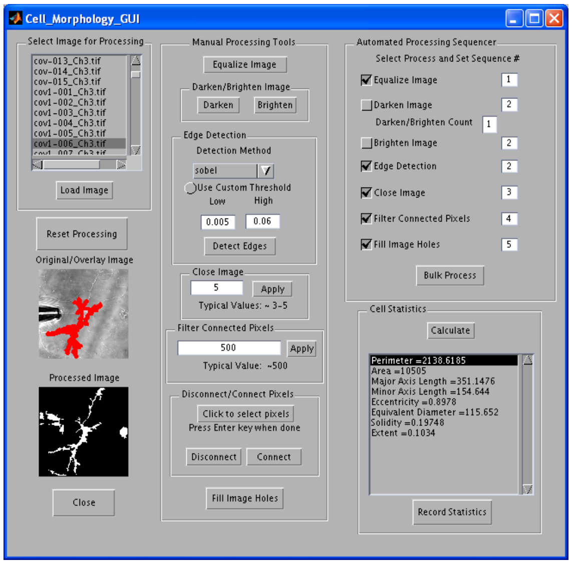
Cell #	Perimeter Change (%)	Area Change (%)	Major Axis Change (%)	Minor Axis Change (%)	Eccentricity Change (%)
1	-51	-63	-52	-26	-33
2	-76	-87	-72	-66	-13
3	+5	-9	+9	-9	+2
4	-37	-44	-46	-2	-25

(a) Good



(b) Bad

Fig. 2. Results for morphological measures from graph-cuts method



cov-013_Ch3.tif
cov-014_Ch3.tif
cov-015_Ch3.tif
cov1-001_Ch3.tif
cov1-002_Ch3.tif
cov1-003_Ch3.tif
cov1-004_Ch3.tif
cov1-005_Ch3.tif
cov1-006_Ch3.tif

Detection Method: sobel

○ Use Custom Threshold — Low 0.005, High 0.06

Close Image: 5 — Typical Values: ~ 3-5

Filter Connected Pixels: 500 — Typical Value: ~500

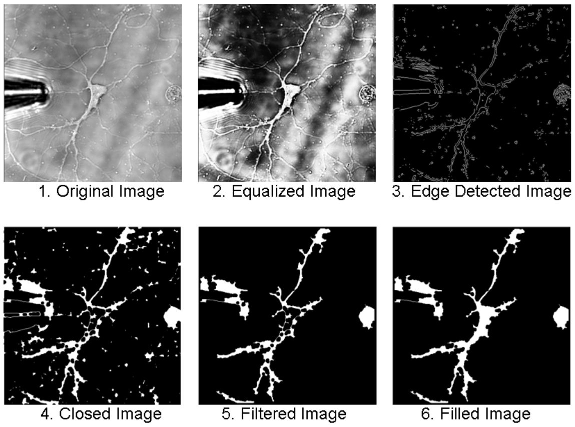
Press Enter key when done

Select Process and Set Sequence #

☑ Equalize Image 1
☐ Darken Image 2
Darken/Brighten Count 1
☐ Brighten Image 2
☑ Edge Detection 2
☑ Close Image 3
☑ Filter Connected Pixels 4
☑ Fill Image Holes 5

```
Perimeter =2138.6185
Area =10505
Major Axis Length =351.1476
Minor Axis Length =154.644
Eccentricity =0.8978
Equivalent Diameter =115.652
Solidity =0.19748
Extent =0.1034
```

(a)



(b)

Fig. 3. Image processing GUI and typical processing steps for segmentation

and instructions how to and what to change are not trivial. Therefore, a stand-alone, more user-friendly Matlab®-based software tool has been developed.

This software tool has been specifically designed and implemented for assessing morphological measures in the astrocyte and fibroblast cells before and after the phototransfection process. A screen shot of the AutoPT Cell Morphology (CM) Graphical User Interface (GUI) that operates the program is shown in Figure 3(a). It has been set up for individual image processing as well as the bulk processing of many images. Once the image to be analyzed has been loaded, the user can then choose from a number of different processing options in the *Manual Processing Tools* panel to apply to the image. These include: equalizing the image (i.e. evenly distributing intensity values throughout the range of intensity values in the image), image darkening/brightening, edge detection, image closing, connected pixel filtering (filtering out connected pixels smaller than specified size), and filling image holes. There is a choice of five common edge detection methods to apply that are all part of Matlab®'s Image Processing Toolbox. The processing can be done in any order, however, typically the order that the tools appear in the *Manual Processing Tools* panel is the order that they are executed. Figure 3(b) shows an original image and subsequently processed images after application of the manual processing tools in this order. There is also an option to manually select pixels in the processed image to either connect or disconnect them from the processed image. Once the image is properly segmented, the cell statistics for the largest connected pixel region are calculated and displayed in the CM GUI. These statistics include the perimeter, area, major axis length, minor axis length, eccentricity, equivalent diameter, solidity, and extent. The original image of the cell is then overlayed with the segmented image of the cell in both main GUI panel and in a separate window. A new image just of the segmented cell is also generated. The *Record Statistics* button can then be used to write this data to a text file and save the original cell image, segmented cell image, and overlay image of the cell in jpg format. The data file written also contains hyperlinks to these saved images.

Once suitable manual processing steps and parameters have been determined for a few test images, bulk processing of all the images in the active directory can be performed with these settings. Inside the *Automatic Processing Sequencer* panel, the process to be performed and sequence number can be selected and inputted. The processing steps will use the parameters set in *Manual Processing Tools* panel and execute the processing on all the images in the active directory, write the corresponding statistics to a text file, and record the original, cell, and overly images, as shown in Figure 4. There are also settings to record just the largest region, three largest, or all the connected pixel regions that are found.

Three image sets, each containing 5 pairs of images corresponding to the same cell before and after the phototransfec-

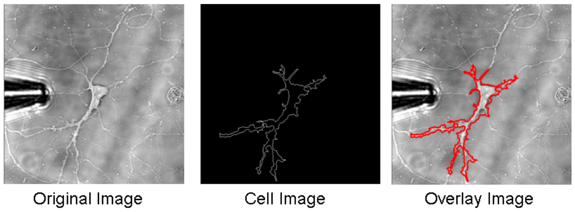


Fig. 4. Cell morphology bulk process image output

TABLE I

MANUAL PROCESSING AREA AND PROCESSING TIME COMPARISON

Image Pair	% Area Change^1	% Area Change^2
1-1	0	-6
1-2	-47	-47
1-3	-71	-74
1-4	-90	-87
1-5	-83	-75
Manual Processing Time	16 min	24 min
2-1	-81	-81
2-2	-94	-93
2-3	-95	-94
2-4	-93	-92
2-5	-89	-87
Manual Processing Time	20 min	20 min
3-1	-65	-68
3-2	-74	-75
3-3	-89	-89
3-4	-82	-78
3-5	-81	-79
Manual Processing Time	16 min	26 min

% Area Change^1: using data from AutoPT Cell Morph GUI analysis
% Area Change^2: using data from Traditional analysis

tion process, were used to compare the performance of this software tool to acquire morphological cell measurements against the traditional method. In the traditional method, the user first traces the cell border in one particular program. This is followed by importing this new cell boundary image to another program to fill in the region inside the cell border. This filled cell image is then imported back into the original program to measure the area of cell. The processing time to analyze each image set using this technique along with the percentage change in the area metric for each image pair were recorded and are listed in Table I (column 3). The same image sets were analyzed manually using the AutoPT CM GUI (Figure 3(a)) and the processing time for each set along with the percentage area change for each image pair recorded and also shown in Table I (column 2). In the case of image sets 1 and 3, the processing time using the AutoPT CM GUI tool is 33% and 38% faster than the traditional method, respectively. The processing time for image set 2 was about the same in both methods. The results for the percentage change in the area metric with the CM GUI program are all within 8% of the results produced with the traditional analysis method. This error is small and can be explained from the fact that the same person did not use both methods (one person used traditional methods while the other used the GUI) and some portions of the cell borders are subject to individual interpretation. It is also expected that more time gains will be realized once the user is more experienced with using the GUI and identifies the best combination of processing controls to segment particular types of images (this is the reason for similar processing times in image set 2). The CM GUI program is also more user-friendly and efficient since all the necessary processing steps are self-contained and there is no need to switch back and forth between different programs to perform the analysis. Further, using the CM GUI provides more than 6X the information than the alternate approach. As stated previously, in addition to the cell area metric, the GUI program yields metrics for the cell perimeter, major axis length, minor axis length, eccentricity, equivalent diameter, and others. This data for the three sets of test images is shown in Table II. For the metrics listed here, they are all substantially decreased (by an average of 58%) after the phototransfection process. The traditional analysis method cannot provide these extra morphological measurements.

TABLE II

MORPHOLOGICAL CHANGES - % CHANGE FROM ORIGINAL

Image Pair	Perimeter	Area	Major Axis	Minor Axis	Eccentricity	Equiv. Dia
1-1	21	0	6	3	4	0
1-2	-16	-47	-24	-36	2	-27
1-3	-47	-71	-43	-49	49	-46
1-4	-81	-90	-83	-58	-47	-69
1-5	-77	-83	-78	-52	-24	-59
2-1	-53	-81	-74	-33	-7	-57
2-2	-82	-94	-81	-71	-16	-76
2-3	-88	-95	-88	-79	-37	-77
2-4	-83	-93	-87	-60	-46	-74
2-5	-79	-89	-84	-42	-76	-67
3-1	-33	-65	-34	-40	5	-41
3-2	-8	-74	2	-43	18	-49
3-3	-80	-89	-83	-43	-41	-67
3-4	-71	-82	-62	-60	-4	-58
3-5	-61	-81	-35	-71	324	-56

The *Bulk Process* function in the AutoPT Morphology GUI was also used to process the three sets of test images. Depending on the processing parameters selected, the processing time to analyze the set ranged from 4.5 to over 40 minutes (using a laptop running Windows XP, with 1.80 GHz Pentium M processor and 1 GB RAM). In each case, data for every connected pixel region greater than 500 pixels was recorded, which depending on the settings can result in a lot of extra processing time. Due to the inconsistencies in the images (lighting conditions, focal length, pipette placement, etc.) it was hard to identify one set of image parameters to successfully segment each cell image. In the best cases, a particular processing parameter set was able to segment about 60% of the images in the set within an acceptable tolerance. Another set of parameters was then selected to try and process the rest of the images. This is then repeated until all the images in the set have acceptable results or the remaining images can just be processed manually. Procedures can be put in place to standardize the image capture settings during the process and produce more consistencies among all the images in an image set that will increase the efficiency and results of the bulk processing of the images. This is a topic of future work.

III. FRAMEWORK FOR PROCESS AUTOMATION

A framework to automate the actual single cell phototranfection process has been developed. The first step in automating the phototransfection process is to instrument a microscope with a motorized stage for closed loop positioning of the cover-slips under the microscope field of view (FOV). Once this is done, a global and local map of each cover-slip can be constructed, as seen in Figure 5. The stage can be indexed and sequential image captures of FOV's in specific locations on the cover-slip performed. A mosaic of all these images can be used to build the global map. This map of the entire cover-slip can then be stored for comparison and analysis at different time intervals. Local maps for individual FOV's of the cover-slip can also be created, where image processing will be performed.

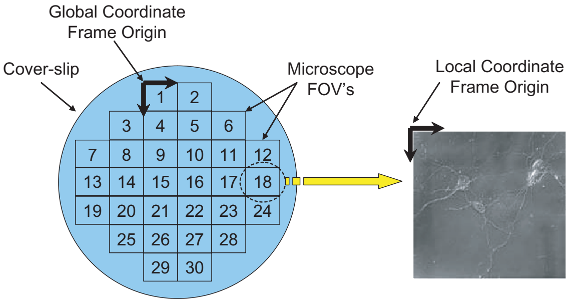


Fig. 5. Automating the phototransfection process

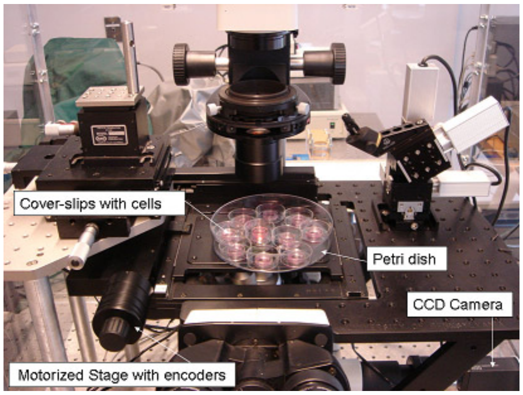


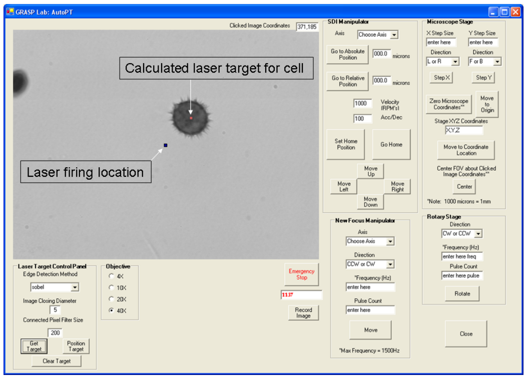
Fig. 6. AutoPT: Hardware Setup

In the individual FOV's, standard computer vision techniques, such as edge detection, image erosion, dilation, filtering, filling [18], can be used to segment the cell body and dendrite area from the background in each image. Local and global image data can then be compiled consisting of cell body coordinates locations, sizes, contour profile statistics, and dendrite section areas and locations. A program can be written to automatically determine suggested laser target firing locations based on image data for each FOV on the cover-slip. These locations will be high curvature regions on the cell body, away from the dendrites and cytosol of the cell. Once all locations are set, coordinated micromanipulator positioning of the injection pipette, laser firing, mRNA release, and stage positioning can be executed across entire global map of the cover-slip all at once, greatly increasing throughput.

A. AutoPT: Proof-Of-Concept Implementation

A proof-of-concept implementation for automating this phototransfection process has been accomplished using the flexible automation micro/meso-scale manipulation system from [4], [5]. The system setup can be seen in Figure 6. Here, an inverted optical microscope, motorized XY stage, and CCD camera are the pertinent pieces of hardware being utilized. Typically, a 40X objective is used to image the cells for this application. The computer controlled micromanipulator can be used to position a pipette for dispensing mRNA and in the future, a laser will be incorporated into the system and focused to fire at the center of the image in the FOV.

The control software to operate the system is written in Visual C#.Net, leveraging the Windows .NET framework, enabling easy integration of software modules that can reside on different workstations. The software includes (a) real-time image capture of images from the microscope; (b) control of the motorized stages; and (c) a simple GUI for the operator to specify the type of cell he/she is interested in by inputting relevant image processing parameters (see Figure 7). The image processing routines are written in Matlab® (version 7.2.0.232, R2006a) using functions from the Image Processing Toolbox.



Clicked Image Coordinates 371,185

Axis: Choose Axis
000.0 microns
000.0 microns
1000 Velocity (RPM's)
100 Acc/Dec

X Step Size: enter here
Y Step Size: enter here
Direction: L or R
Direction: F or B
Stage XYZ Coordinates: X,Y,Z
*Note: 1000 microns = 1mm

Edge Detection Method: sobel
Image Closing Diameter: 5
Connected Pixel Filter Size: 200

Objective: ○ 4X ○ 10X ○ 20X ● 40X

1137

Axis: Choose Axis
Direction: CCW or CW
*Frequency (Hz): enter here
Pulse Count: enter here
*Max Frequency = 1500Hz

Direction: CW or CCW
*Frequency (Hz): enter here freq
Pulse Count: enter here pulse

Fig. 7. AutoPT: Software GUI

Using the *Laser Target Control Panel* found in the lower left corner of the GUI, the user specifies the parameters for the image processing code in Matlab®. Parameters that are set include the type of edge detection method to use (Canny, Sobel, Roberts, Prewitt, Laplacian), the diameter for the image closing operation, and pixel size for a connected pixel filtering procedure. Note that these parameters are set once. Once these values are set, the user can then click the *Get Target* button to calculate a recommended laser target firing location of the cell of interest in the FOV. This button saves the image from the current image frame along with the specified parameters and then calls Matlab® to perform the necessary calculations to segment the image of the cell from the background and recommend image coordinates to fire the laser. In this current implementation, this position is just determined as the centroid of the cell body. However, more sophisticated metrics to calculate the laser target position can easily be applied here instead. The laser target position information is then sent back to the main control program and drawn on the screen in blue. Once this position is verified as being accurate, the user can then click the *Position Target* button and the motorized stage will then translate the cell in the XY plane so that the laser target position calculated is now at the center of the image where the laser is parked. By coupling the *Get Target* and the *Position Target* function with the laser firing and mRNA release from a pipette mounted on the motorized manipulator in the system, the system will be completely automated. Once finished, the *Clear Target* button can be used to reset the laser target position in the computer memory and move the stage back to it's original position.

B. Estimated Throughput

On a single control computer running Windows XP, with a 2.39 GHz Pentium 4 processor and 1 GB of RAM, it takes 30 seconds to segment and identify a target location for the cell and translate the XY stage so that the laser can be used at the target location. This corresponds to a throughput of 120 cells/hour, which is a 6X improvement over the current manual procedure (20 cells/hour). By coupling all the software modules more efficiently (eliminating the C# wrappers with Matlab software) and by processing all cells in the field of view (typically 4-6), the throughput is expected to increase to over 500 cells/hour. This is greater than an 25X improvement. Also, using a newer, faster computer would further decrease the cycle time. This system can also be run continuously, only needing a human to be there to replenish a new batch of cells and remove the processed ones. Assuming a 12-hour day at a rate of 500 cells/hour projects to a throughput of 6000 cells/12-hr day.

As proof-of-concept for the increased time gains from using one integrated program, the C# program functionality was converted to a Matlab® program capable of acquiring images from the CCD camera, processing the image, calculating laser target positions, and moving the XY stage. Running everything in the same program reduced the process time to about 8 seconds from 30 seconds. This corresponds to a throughput of 450 cells/hour, a 23X improvement from the current manual process. Again, assuming that all the cells (typically 4-6) in the FOV can be processed with minimal increased computational overhead, a potential throughput of 2250 cells/hour (113X improvement) is estimated. In practice, this fully integrated program can not be written in Matlab® since the images from the confocal microscope, that is used for the actual procedure, are captured with a photomultiplier tube (PMT). The PMT is not compatible with Matlab's image acquisition toolbox which has been used here to capture images from the CCD camera in the test setup. Therefore, custom software is required to capture the PMT images, perform the appropriate image segmentation, calculate laser target positions, and translate the XY stage in order to achieve these further throughput gains.

To get the maximum possible throughput out of the entire system, considerations for automatically refilling the micropipette with mRNA should be made along with investigations on how to move the processed cover-slip out of the way, store it in an organized manner while feeding in the next one to be processed, with as limited human interactions as possible.

IV. SUMMARY

Work on applying flexible automation technologies to the single cell manipulation process of phototransfection is presented in this paper. Phototransfection is presently done manually in a very tedious manner. A framework for fully automating this procedure has been designed and proof-of-concept implementation achieved. Computer vision techniques are used to identify the cell of interest in the FOV and determine target locations for the laser beam. A control program takes this information and coordinates movements of the computer controlled XY stage, translating the coordinates of the laser target location to a predefined, fixed, laser firing location. A $23X$ improvement is possible with this implementation with room for improvement to greater than $110X$ described.

Images of the phototransfected cell have been observed before and after the process and a software tool developed to assess morphological changes in the cell as a way to characterize them and assess the efficacy of the phototransfection process. Image segmentation algorithms were used to segment the cell from the background in order to compare both images of the cell without ambiguities. From the properly segmented image, the morphology is quantified by computing measures such as cell area, asymmetry, perimeter, and eccentricity. Results show a notable decrease in the metrics after the process has been performed, a throughput increase over manual cell morphology measurements, a 6X gain in the number of measurements made, and a more efficient and user-friendly software tool for cell morphological analysis produced.

V. ACKNOWLEDGMENTS

The authors gratefully acknowledge funding from NSF Grant IIS-0413138, Dept. of Education GAANN Grant P200A060275, the Keck Foundation and the NIH Director's Pioneer Award Program, DP1-OD-04117 to support this work and Kitty Wu for discussions on the manual phototransfection procedure.

REFERENCES

[1] D. Forsyth adn J. Ponce. *Computer Vision: A Modern Approach*. Prentice Hall, Upper Saddle River, NJ, 2003.

[2] P. Bianco and P. Robey. Stem cells in tissue engineering. *Nature*, 414(6859):118–121, November 2001.

[3] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on* PAMI, 26(9):1124–1137, September 2004.

[4] D. Cappelleri. *Flexible Automation of Micro and Meso-Scale Manipulation Tasks with Applications to Manufacturing & Biotechnology*. Ph.d. dissertation, University of Pennsylvania, Philadelphia, PA, August 2008.

[5] P. Cheng, D. Cappelleri, B. Gavrea, and V. Kumar. Planning and control of meso-scale manipulation tasks with uncertainties. In *Proceedings of Robotics: Science and Systems*, Atlanta, GA, USA, June 2007.

[6] B. Choi, S. Jin, S. Shin, J. Koo, S. Ryew, M. Kim, J. Kim, W. Son, K. Ahn, W. Chung, and H. Choi. Development of flexible laboratory automation platform using mobile agents in the clinical laboratory. In *IEEE Conference on Automation Science and Engineering (CASE)*, Washington DC, USA, August 23-26 2008.

[7] S. Hamilton and M. Russo. Tutorial/workshop: Introduction to laboratory automation. IEEE Conference on Automation Science and Engineering (CASE), September 2007.

[8] H. Huang, D. Sun, J. Mills, and S. Cheng. Integrated vision and force control in suspended cell injection system: Towards automatic batch biomanipulation. In *IEEE International Conference on Robotics and Automation (ICRA)*, Pasadena, CA, USA, May 19-23 2008.

[9] D. Kaber, N. Stoll, and K. Thurow. Human-automation interaction strategies for life science applications: Implications and future research. In *IEEE Conference on Automation Science and Engineering (CASE)*, Scottsdale, AZ, USA, Sept 22-25 2007.

[10] Y. Kimura and R. Yanagimachi. Intracytoplasmic sperm injection in the mouse. *Biol. Reprod.*, 52:709–720, 1995.

[11] J. Kuncov-Kallio and P. Kallio. Lab automation in cultivation of adherent cells. *IEEE Transactions on Automation Science and Engineering)*, 3(2):177–186, April 2006.

[12] R. Lovell-Badge. The future for stem cell research. *Nature*, 414(6859):88–91, November 2001.

[13] P. Najmabadi, A. Goldenberg, and A. Emili. A scalable robotic-based laboratory automation system for medium-sized biotechnology laboratories. In *IEEE Conference on Automation Science and Engineering (CASE)*, Edmonton, Canada, August 1-2 2005.

[14] J. Park, S. Jung, and Y. Kim. Design and fabrication of an integrated cell processor for single embryo cell manipulation. *Lab Chip*, 5:91–96, 2005.

[15] J. Sul, C. Wu, F. Zeng, J. Jochems, M. Lee, T. Kim, T. Peritz, P. Buckley, D. Cappelleri, D. Maronski, M. Kim, V. Kumar, D. Meaney J. Kim, and J. Eberwine. Transcriptome transfer produces a predictable cellular phenotype. *In review: Proceedings of the National Academy of Sciences*, 2009.

[16] K. Takahashi and S. Yamanaka. *Cell*, 126:663–676, 2006.

[17] S. Temple. The development of neural stem cells. *Nature*, 414(6859):112–117, November 2001.

[18] E. Trucco and A. Verri. *Introductory Techniques for 3-D Computer Vision*. Prentice Hall, Upper Saddle River, New Jersey, 1998.

[19] W. Wang, X. Liu, and Y. Sun. Autonomous zebrafish embryo injection using a microrobotic system. In *IEEE Conference on Automation Science and Engineering (CASE)*, pages 363–368, Scottsdale, AZ, USA, September 22-25 2007.

[20] W. Wang, Y. Sun, M. Zhang, R. Anderson, L. Langille, and W. Chen. A microrobotic adherent cell injection system for investigating intracellular behavior of quantum dots. In *IEEE International Conference on Robotics and Automation (ICRA)*, Pasadena, CA, USA, May 19-23 2008.

[21] M. Zhang, R. Felder, E. Kim, B. Nelson, B. Pruitt, Y. Zheng, and D. Meldrum. Editoral: Special issue on life science automation. *IEEE Transactions on Automation Science and Engineering)*, 3(2):137–140, April 2006.