Toward a Fully Automated High-Throughput Phototransfection System

David J. Cappelleri,[1] Adam Halasz,[2] Jai-Yoon Sul,[3] Tae Kyung Kim,[3] James Eberwine,[3] and Vijay Kumar[4]*

[1]*Department of Mechanical Engineering, Stevens Institute of Technology, Hoboken, NJ*
[2]*Department of Mathematics, West Virginia University, Morgantown, WV*
[3]*Department of Pharmacology, PENN Genome Frontiers Institute, University of Pennsylvania, Philadelphia, PA*
[4]*Department of Mechanical Engineering, GRASP Laboratory, University of Pennsylvania, Philadelphia, PA*

Keywords: robotics, image segmentation, computer vision, phototransfection, cell morphology

We have designed and implemented a framework for creating a fully automated high-throughput phototransfection system. Integrated image processing, laser target position calculation, and stage movements show a throughput increase of $>23\times$ over the current manual phototransfection method although the potential for even greater throughput improvements ($>110\times$) is described. A software tool for automated off-line single-cell morphological measurements, as well as real-time image segmentation analysis, has also been constructed and shown to be able to 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.
(JALA 2010;15:329–41)

DJC is a recipient of a 2009 ALA Young Scientist Award from the IEEE Robotics and Automation Society (www.ieee-ras.org).

*Correspondence: Dr. David J. Cappelleri, Department of Mechanical Engineering, Stevens Institute of Technology, 1 Castle Point on Hudson, Hoboken, NJ 07030, USA; Phone: +1.201.216.5072; Fax: +1.201.216.8315; E-mail: David.Cappelleri@stevens.edu

1535-5535/$36.00

doi:10.1016/j.jala.2010.03.003

Introduction

Stem cell research has become very prevalent in recent years.[1–3] Stem cells are naturally produced by the body during embryonic development; thus, this research is popular because these cells contain a blueprint of how to build everything in our body. 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 by using viruses to deliver a set of transcription factor complementary DNAs into mature cells that will then dedifferentiate these cells into an induced pluripotent stem (iPS) cell.[4] Signals (instructions) can then be sent to this iPS cell to lead it down a desired developmental pathway to create specific cell types. This procedure is shown schematically in Figure 1A. 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 (TIPeR) approach,[5,6] 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

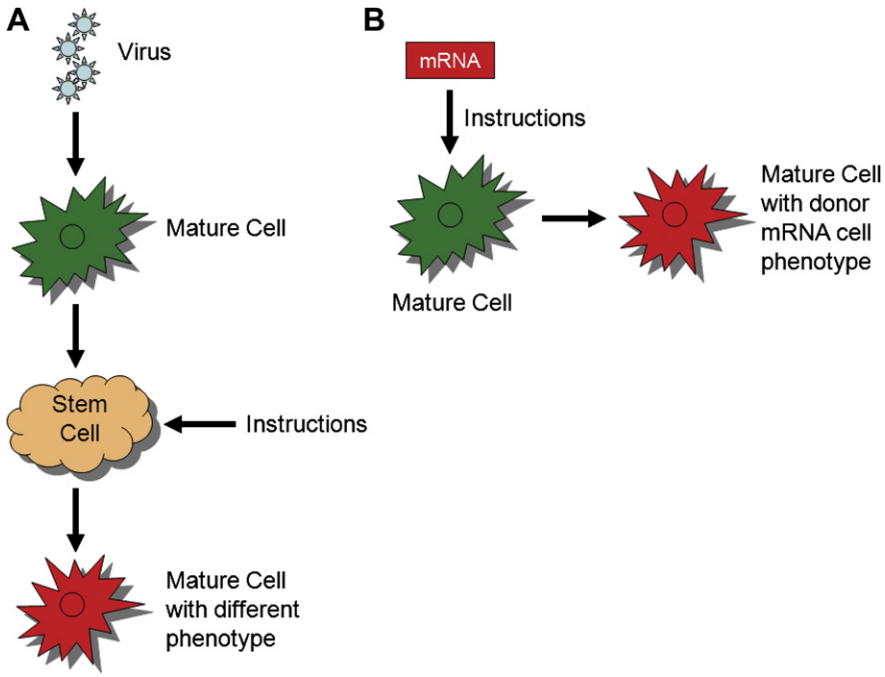

Figure 1. Artificial stem cell creation versus phototransfection for changing cell phenotype. (A) Artificial stem cells (iPS) created from a virus followed by instruction (cell signaling) to induce phenotype change. (B) Phototransfection: combination of laser poration of cell along with extracellular delivery of mRNA to induce phenotype change.[5]

population into the host cell. One method for performing TIPeR is through the use of transfection [7–12] to transiently introduce holes into the host cell through which messenger RNA (mRNA) populations can diffuse. Once the holes reseal, the introduced mRNA will be translated and produce functional proteins that can modify the host cell phenotype. Transfection in the work of Barrett et al.[5] is performed with a titanium sapphire laser, and this combination of cell poration with the extracellular delivery of mRNA is termed "phototransfection." Phototransfection provides a means for performing functional genomics manipulations on individual cells and is pictured schematically in Figure 1B in contrast to the iPS-based approach for changing cell phenotypes. The current, manual phototransfection procedure consists of the following steps:

1. Locate cells of interest on a marked coverslip and record their locations on paper with the same markings.
2. Identify the cytosol and edges of the cell of interest.
3. 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.
4. Locally apply mRNA of a donor cell to the target cell using a pipette.
5. Observe cell changes at various time intervals (hours, days, and weeks) and repeat as needed.

This process is very tedious and inefficient. The overall yield rate of the cells just surviving the process is 70%–80%, not that they are necessarily changing from one type of cell to another. The current morphological measurements here are also inadequate. They only provide a cell area metric in which the user first traces the cell border in one particular program and saves it as a cell boundary image. This is followed by importing this new 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 actually measure the area of cell.

The throughput for the manual phototransfection process is 20 cells/h. The goal here is to apply flexible automation techniques to increase the throughput to about 360 cells/h. This is important to rapidly explore many different amounts and types of donor RNAs, 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 better quantify the cell morphology (CM) 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 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, and injecting/ejecting materials in/out of cells is becoming an active research area.[13,14] These types of cell manipulation tasks are important for the characterization and manipulation of single embryo cells in applications such as cloning, gene expression analysis, cell replacement therapy,[15] intracytoplasmic sperm injection, 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.[16–20] There is also recent work on integrating electroporation into a robotic manipulation system for autonomous injections of single cells.[21] Various types of other platforms for laboratory automation have also been presented. A "tower-based configuration" for the automatic execution of various biotechnology (genomics and proteomics) protocols is presented in the work by Najmabadi et al.,[22] whereas Choi et al.[23] present a robotic platform for clinical tests suitable for small- or medium-sized laboratories using mobile robots. A high-throughput automated genome and chemical analysis system is shown in Meldrum et al.,[24] and an automated microscope platform for biological studies, drug discovery, and medical diagnostics is illustrated in Potsaid et al.[25] 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 the works by Kaber et al.[26] and Kuncov-Kallio and Kallio,[27] respectively. Also, previous work on image segmentation techniques for biological applications is shown in Makkapati et al.[28] and Makkapati[29] to properly identify the presence of tuberculosis in biologically stained images and for autofocusing of images of blood smears containing red blood cells, respectively. An automated microscope system for monitoring the vitality of neuron cells that relies on identifying fluorescent makers has been presented in Arrasate and Finkbeiner.[30] In Geisler et al.,[31] an integrated system for simultaneous measuring of fluorescence microscopic and integrated sensor-based data is presented as a possible enabling technology for future screening assays. Also taking advantage of fluorescent markers is the work by Neumann et al.[32] in which an automated platform for high-content RNA interference (RNAi) screening that uses time-lapse fluorescence microscopy of live HeLa cells expressing histone–GFP to report on chromosome segregation and structure is reported. An automated platform for high-throughput cell phenotype screening combining human live cell arrays, screening microscopy, and machine-learning-based classification methods based on the identification of the subcellular localization of marker proteins as indicators for the cellular state is described in Conrad et al.[33]

The work presented this article describes a framework for fully automating the phototransfection process of single cells (astrocytes and fibroblasts). Two approaches to handle the main automation challenge of processing the cell images in real time and off-line for morphological comparisons are presented. A software analysis tool for automating cell morphological measurements for quantitative comparison of images of the cells before and after the process is described. This is followed by a detailed description of a proof-of-concept implementation, the framework for automating the current manual phototransfection process along with the estimated process throughput results. Recommendations for further improvements are also provided.

Framework for process automation

A framework to automate the actual single-cell phototransfection process has been developed and is pictured schematically in Figure 2. The first step in automating the phototransfection process is to instrument an optical microscope with a motorized stage for closed-loop positioning of the coverslips under the microscope field of view (FOV) (Fig. 2A). Once this is done, a global and a local map of each coverslip can be constructed, as seen in Figure 2B. The stage can be indexed and sequential image captures of the FOVs in specific locations on the coverslip performed. A mosaic of all these images can be used to build a global map. This map of the entire coverslip can then be stored for comparison and analysis at different time intervals. Local maps for individual FOVs of the coverslip can also be created, where image processing will be performed. In the individual FOVs, standard computer vision techniques, such as edge detection, image erosion, dilation, filtering, and filling,[34] can be used to segment the cell body and processes (dendrite) area from the background in each image (Fig. 2C). Local and global image data can then be compiled consisting of cell body coordinate locations, sizes, contour profile statistics, and processes (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 coverslip. These locations will be high-curvature regions on the cell body, away from the dendrites and cytosol of the cell, as shown in Figure 2D. Once all locations are set, coordinated stage movements followed by laser firing (Fig. 2E), micromanipulator positioning of the injection pipette, mRNA release (Fig. 2F), and stage repositioning can be executed across the entire global map of the coverslip, greatly increasing throughput. Once all the FOVs on a particular coverslip have been phototransfected, the process will be repeated on the next coverslip in the petri dish.

The main challenge in automating the phototransfection process is in identifying the appropriate features of the cells in the image to direct the laser beam to create the pores in the cell membrane where the mRNA can diffuse into it. These features can be identified with image segmentation techniques, and then, these segmented images can be used to automatically determine morphological measures of the cells (for comparison before and after the process) as well as the laser target firing locations.

Segmentation and automated CM 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

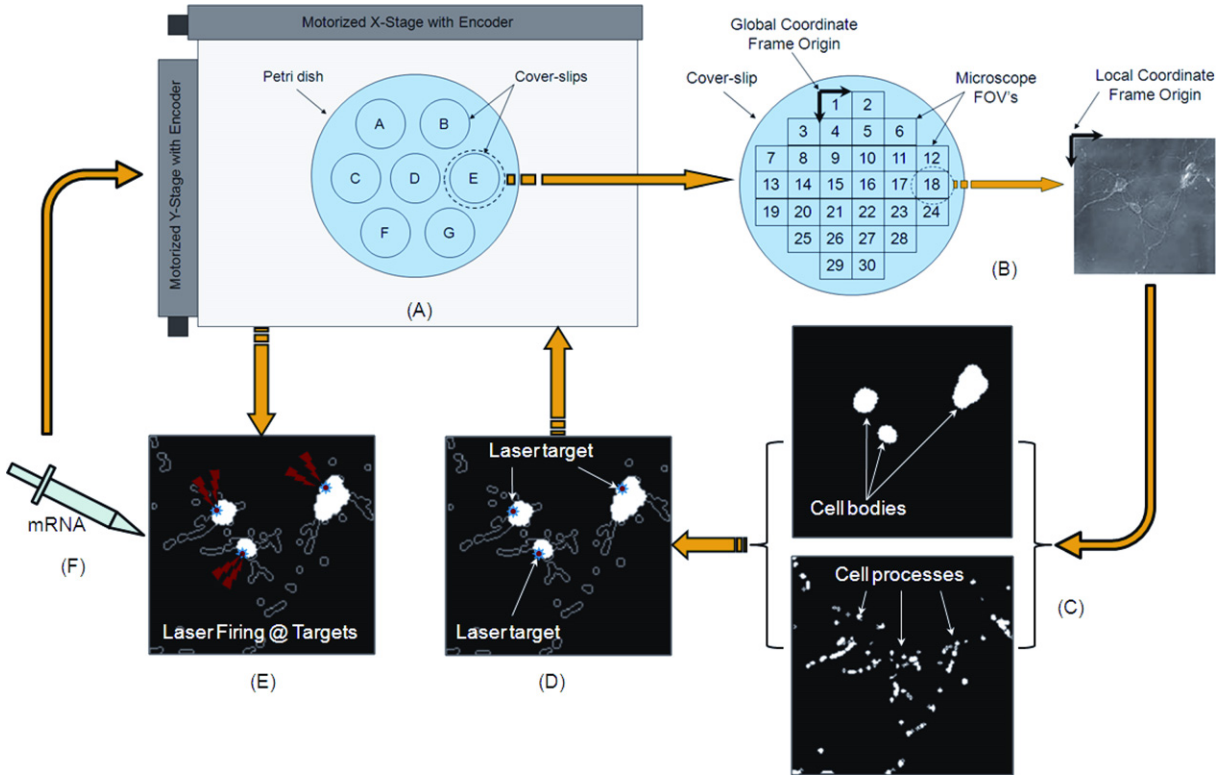

Figure 2. Process flow for AutoPT. (A) Instrumented microscope with fixed-position petri dish with coverslips of cells to be phototransfected. (B) Local mapping of individual coverslips for image processing and data retrieval. (C) Image segmentation to identify cell features and morphology. Morphological data are stored in a database for each cell in the FOV. (D) Laser target calculation is based on feature extraction and morphology. (E) Coordinated microscope stage movements and laser firing to administer laser at designated targets. (F) mRNA release into porated cells by pipette mounted on computer-controlled micromanipulator. The process loop then restarts on next FOV on coverslip. Once all FOVs on coverslip have been phototransfected, the process continues on next coverslip in the petri dish.

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 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, eccentricity, and equivalent diameter. This segmented image can also allow for robust image feature identification and laser target coordinate firing location calculations.

A schematic of some of the morphological measurements is shown in Figure 3. The image on the left represents the appearance of a cell before phototransfection, whereas the image on the right represents its morphology after phototransfection. Ideally, it will start to resemble and function like the donor cell, and the calculated morphology will be used as one way to quantify this change. The area of the segmented cell region of the image is defined as the actual number of pixels in the region. The perimeter metric is calculated by determining the distance between each adjoining pair of pixels around the border of the contiguous segmented cell region in the image. The major axis and minor axis lengths are lengths in pixels of the major and minor axes, respectively, of an ellipse that is fit to the segmented region that has the same normalized second central moments. The eccentricity measure is determined from this same ellipse and is the ratio of the distance between the foci of the ellipse and its major axis length. It is between 0 and 1. An ellipse with eccentricity = 0 is actually a circle, and an ellipse with eccentricity = 1 is a line segment. The equivalent diameter measure is a scalar value that specifies the diameter of a circle with the same area as the segmented region. It is computed as $\sqrt{(4 \times \text{Area})/\pi}$.

Initially, images of the cells were segmented using graph-theoretic clustering techniques, using the image pixels as nodes in the graph.[35] Once a connected, weighted graph is constructed from the image of interest, a graph-cutting algorithm can be executed 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,[36] which is used here, solves this problem by finding the maximum flow from the "Source" nodes to the "Sink" nodes in the graph (Fig. 4), that is, the maximum "amount of water" that can be sent from the "Source" to the "Sink" by interpreting graph edges as "pipes" with capacities equal to the edge weights. 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 the "Source." For this application, the "Sink"

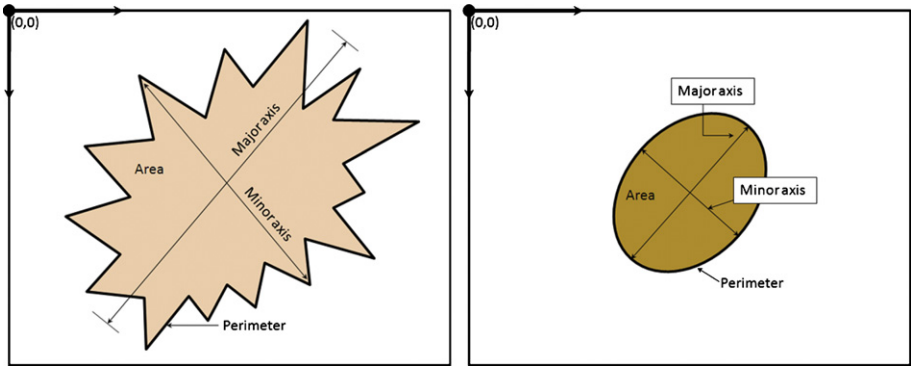

Figure 3. Schematic of a cell before (left) and after (right) phototransfection and associated morphological measures quantifying cell changes.

corresponds to pixels in the background of the image, whereas 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 values between similar nodes are large, whereas the affinity measures connecting different nodes are small. The distance affinity measure goes down sharply once the distance between the pixels is more than some threshold. The pixel intensity affinity is large for similar intensities and smaller as the intensity difference increases. Similarly, the texture affinities are large for pixels with similar surrounding textures and smaller as the difference increases. These three different affinity measures between two nodes, N1 and N2, are listed in Eqs. (1)–(3), whereas the corresponding edge weight, E, is given in Eq. (4):

$$A_{d\ \mathrm{N1,N2}} = \exp\left\{ -(P_{\mathrm{N1}} - P_{\mathrm{N2}})^2/(2\sigma_d^2)\right\} \tag{1}$$

$$A_{i\ \mathrm{N1,N2}} = \exp\left\{ -(I_{\mathrm{N1}} - I_{\mathrm{N2}})^2/(2\sigma_i^2)\right\} \tag{2}$$

$$A_{t\ \mathrm{N1,N2}} = \exp\left\{ -(T_{\mathrm{N1}} - T_{\mathrm{N2}})^2/(2\sigma_T^2)\right\} \tag{3}$$

$$E_{\mathrm{N1,N2}} = w_1 A_d + w_2 A_i + w_3 A_t, \tag{4}$$

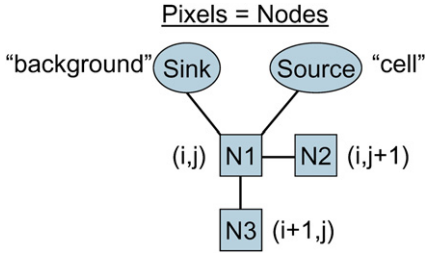

Figure 4. Connected graph from an image: image pixels = graph nodes.

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 and low affinity values for dissimilar pixels. The weights, w_1, w_2, and w_3, are user defined, and each is ≤ 1 and 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. Eqs. (5)–(11) are used for this. Here, $A_{d\ Bkg}$, $A_{i\ Bkg}$, and $A_{t\ Bkg}$ are distance, intensity, and texture affinities associated with the background ("Sink") section of the image that are precomputed from a set of training images.

$$A_{d\ \mathrm{Ave}} = \frac{A_{d\ \mathrm{N1,N2}} + A_{d\ \mathrm{N1,N3}}}{2} \tag{5}$$

$$A_{i\ \mathrm{Ave}} = \frac{A_{i\ \mathrm{N1,N2}} + A_{i\ \mathrm{N1,N3}}}{2} \tag{6}$$

$$A_{t\ \mathrm{Ave}} = \frac{A_{t\ \mathrm{N1,N2}} + A_{t\ \mathrm{N1,N3}}}{2} \tag{7}$$

$$F_{\mathrm{N1}} = A_{d\ \mathrm{Ave}} + A_{i\ \mathrm{Ave}} + A_{t\ \mathrm{Ave}} \tag{8}$$

$$F_{\mathrm{Sink}} = A_{d\ \mathrm{Bkg}} + A_{i\ \mathrm{Bkg}} + A_{t\ \mathrm{Bkg}} \tag{9}$$

$$E_{\mathrm{N1,Sink}} = \exp\left\{ -(F_{\mathrm{N1}} - F_{\mathrm{Sink}})^2/(2\sigma_s^2)\right\} \tag{10}$$

$$E_{\mathrm{N1,Source}} = 1 - E_{\mathrm{N1,Sink}}. \tag{11}$$

The raw output from the graph-cut algorithm needs to be filtered to come up with the final segmented image of the cell from the background. Image erosion and dilation steps are applied in Matlab (The MathWorks Inc., Natick, MA), and the largest connected pixel region that is left is used as the segmented cell image and statistics are reported on it.

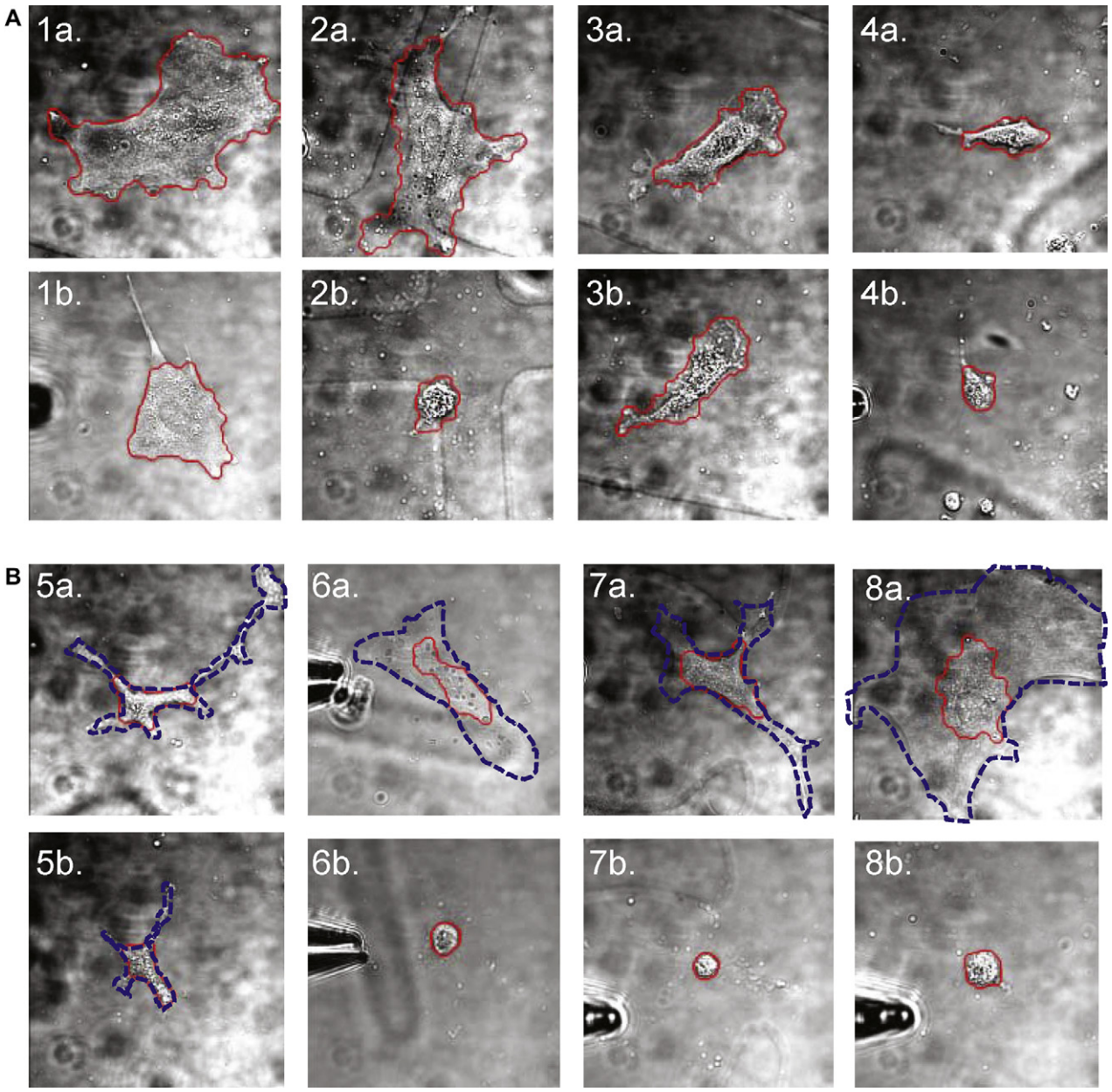

Figure 5. Images of four fibroblast cells before and after the phototransfection process, with the segmented areas from the graph-cuts method overlaid (red) on the original images. The images in the top row are before the process, whereas those in the bottom row are after the process has been completed. (A) Good segmentation results from the graph-cuts method. (B) Bad segmentation from the graph-cuts method: actual cell contour shown with dashed blue line.

Figure 5A shows the result from this procedure on images of four fibroblast cells before and after the phototransfection process, with the segmented areas overlaid on the original images. The images in the top row are before the process, whereas those in the bottom row are after the process has been completed. The cell perimeter, area, major and minor axes, and eccentricity (in pixels) are calculated for each set of images and the corresponding changes in these morphological measures reported in Table 1. These metrics show substantial changes after the phototransfection process has been performed. This indicates a successful phototransfection because the fibroblasts now are starting to look like the donor astrocyte cells, and there are metrics to support this.

However, because of the large changes in CM, inconsistencies in the lighting conditions, coverslip 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 5B shows examples of poor image segmentation, when only a subset region of the actual cell is identified, using the same set of system

Table 1. Morphological measurements from graph-cuts method with good segmentation results

Cell number	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

parameters as those in Figure 5A. 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, and instructions how and what to change are not trivial to the expected end user (biologist). Therefore, a stand-alone, more user-friendly Matlab-based software tool has been developed. (Note: the term "acceptable" is used in comparison with the results obtained from manual operation or calculation methods. Acceptable performance is deemed within 10% of these manually obtained values.)

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 automated phototransfection (AutoPT) CM Graphical User Interface (GUI) that operates the program is shown in Figure 6A. 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 6B 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 overlaid 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 be used to write these 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 can be selected and the corresponding sequence number entered. 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 overlay images, as shown in Figure 7. 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 five pairs of images corresponding to the same cell before and after the phototransfection 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 the 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 2 (column 3). The same image sets were analyzed manually using the AutoPT CM GUI (Fig. 6A), and the processing time for each set along with the percentage area change for each image pair are recorded and also shown in Table 2 (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, whereas 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

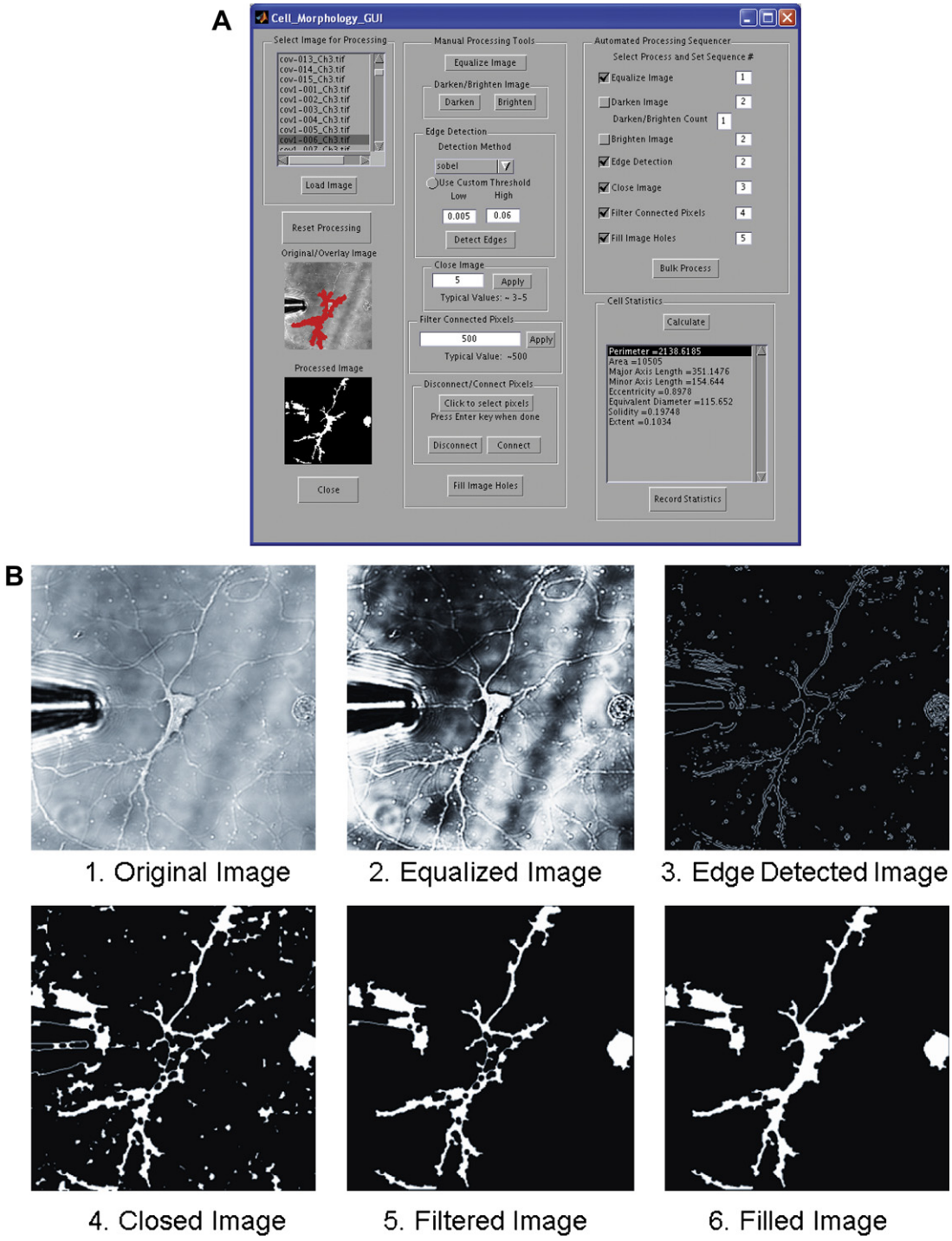

Figure 6. CM image processing toolbox GUI and typical segmentation processing steps. (A) CM GUI Front Panel Display. (B) Sample output from the execution of typical processing steps in the order presented.

in image set 2). The CM GUI program is also more user-friendly and efficient because 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. Furthermore, using the CM GUI provides more than 6× 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. These data for the three sets of test images are shown in Table 3. 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.

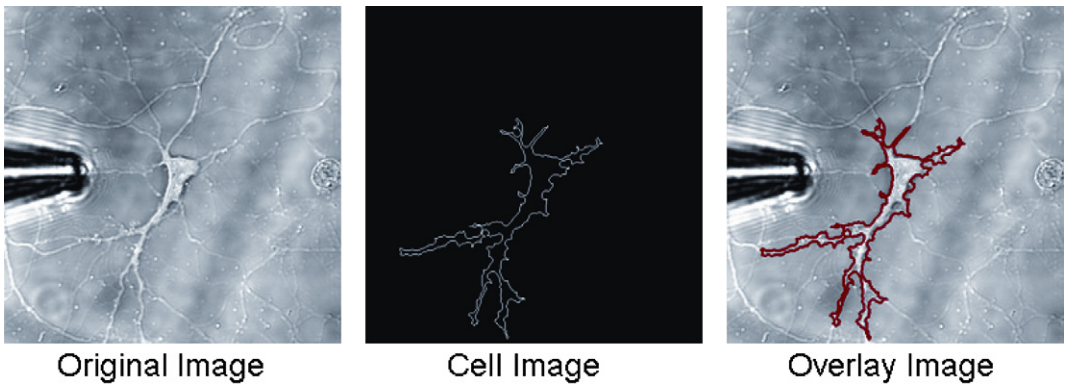

Figure 7. CM bulk processing image output.

Changes required for automation

The *Bulk Process* function in the AutoPT Morphology GUI was also used to automate the processing of the three sets of test images. Using a laptop running Windows XP, with 1.80 GHz Pentium M processor and 1 GB RAM, and depending on the processing parameters selected, the processing time to analyze the set ranged from 4.5 to more than 40 min. In each case, data for every connected pixel region greater than 500 pixels were recorded, which depending on the settings can result in a lot of extra processing time. Because of 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. This is also the case when processing the cell images in real time during the phototransfection procedure when trying to calculate the laser target positions on the cell. In the best cases when using the off-line *Bulk Process* functionality, a particular processing parameter set was able to segment about 60% of the images in the set within an acceptable tolerance. To process the rest of the images, another set of parameters is selected. This is repeated until all the images in the set have acceptable results or the remaining images can just be processed manually. Standardized procedures to determine the image capture settings during the process are required to produce more consistencies among all the images in an image set to increase the efficiency and results of both the bulk processing and real-time processing of the images. Also, optimized code is needed to further increase the processing speeds. Thus, one cannot simply automate a manual process without considering the impact of the manual procedures on the automation task at hand.

Proof-of-concept implementation for auto-PT

A proof-of-concept implementation for automating this phototransfection process has been accomplished using the flexible automation micro/meso-scale manipulation system from Cappelleri[37] and Cheng et al.[38] The system setup can be seen in Figure 8A. Here, an inverted optical microscope (Eclipse TEU2000-U, Nikon Corporation, Tokyo, Japan), motorized *XY* stage (H107 ProScan II, Prior Scientific Inc., Rockland, MA), and CCD camera (XC-77, Sony Corporation, Tokyo, Japan) are the pertinent pieces of hardware being used. There is also a four-axis computer-controlled micromanipulator (MX7600R, Siskiyou Corporation, Grants Pass, OR) and associated controller (MC2000, Siskiyou Corporation, Grants Pass, OR) available for use in the test bed. The computer-controlled micromanipulator can be used to position a pipette for dispensing mRNA. Typically, a 40× objective is used to image the cells for this application. The phototransfection process uses a titanium sapphire laser to perforate the cell membranes, and the laser can be directed to any region of the microscope FOV to administer the laser beam. There is

Table 2. Manual processing area metric and processing time comparison

Image pair	% Area change[a]	% Area change[b]
1–1	0	−6
1–2	−47	−47
1–3	−71	−74
1–4	−90	−87
1–5	−83	−75
Manual processing time (min)	16	24
2–1	−81	−81
2–2	−94	−93
2–3	−95	−94
2–4	−93	−92
2–5	−89	−87
Manual processing time (min)	20	20
3–1	−65	−68
3–2	−74	−75
3–3	−89	−89
3–4	−82	−78
3–5	−81	−79
Manual processing time (min)	16	26

[a]Using data from AutoPT CM GUI analysis method.
[b]Using data from the traditional analysis method.

Table 3. Morphological changes—% change from original

Image pair	Perimeter	Area	Major axis	Minor axis	Eccentricity	Equivalent diameter
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

currently no laser in this implementation, but when incorporated into this system in the future, it will be 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 (1) real-time image capture of images from the microscope; (2) control of the motorized stages; and (3) a simple GUI (Fig. 8B) for the operator to specify the type of cell he/she is interested in by entering relevant image processing parameters. The image processing routines are written in Matlab (version 7.2.0.232, R2006a) using functions from the Image Processing Toolbox.

The *Laser Target Control Panel* found in the lower left corner of the GUI (Fig. 8B) allows the user to specify the parameters for the image processing code in Matlab. The tunable parameters include the type of edge detection method to use (Canny, Sobel, Roberts, Prewitt, and Laplacian), the diameter for the image closing operation, and pixel size for a connected pixel filtering procedure. Note that these parameters are set just once. The *Get Target* button calculates 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 recommends 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 pink. Once the laser target position has been established, the *Position Target* button can be used to have the motorized stage automatically translate the cell in the *XY* plane so that the calculated laser target position is now at the center of the image where the laser will be parked (laser firing location in Figure 8B). The *Clear Target* button is used to reset the laser target position in the computer memory and move the stage back to its original position. By coupling the *Get Target* and the *Position Target* functions with the laser firing and mRNA release from a pipette mounted on the motorized manipulator in the system (as planned in the future), the system will be completely automated.

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 s to segment and identify a target location for the cell and translate the XY stage to move the cell's laser target to the center of the image for eventual laser firing and mRNA release. (The laser firing and mRNA release can be done practically simultaneously and is the easiest and fastest part of the phototransfection process, taking about 1–2 s to do manually.) The 30-s processing time corresponds to a throughput of about 120 cells/h, which is a 6× improvement over the current manual procedure (20 cells/h). By coupling all the software modules more efficiently (eliminating the C# wrappers with Matlab software) and by processing all cells in the FOV (typically four to six), the throughput is expected to increase to more than 500 cells/h. This is greater than a 25× improvement. Also, using a 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-h day at a rate of 500 cells/h projects to a throughput of 6000 cells/12-h day.

As proof of concept for the increased time gains from using one integrated program, the C# program functionality

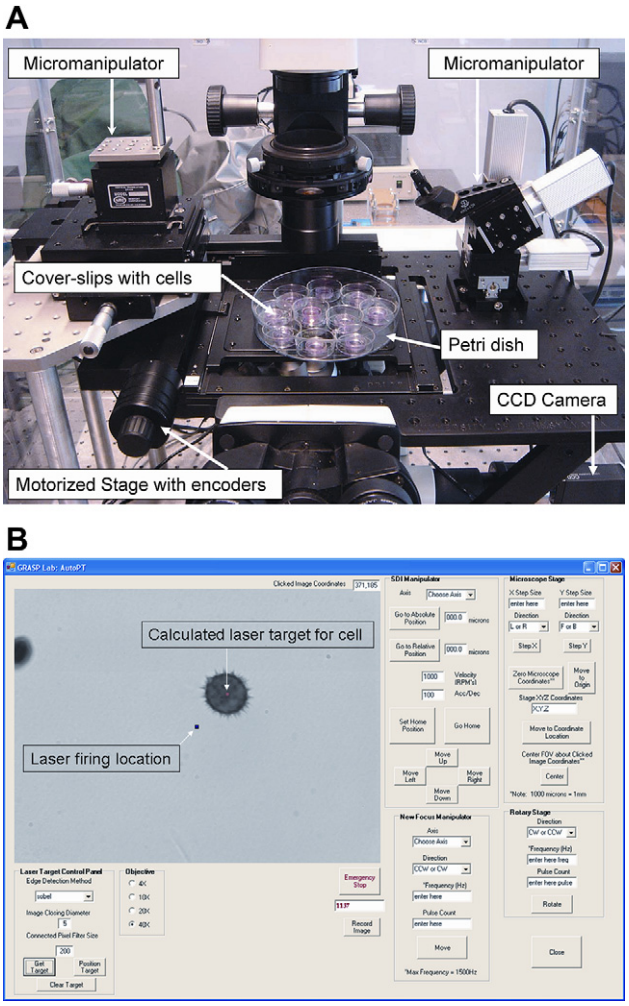

Figure 8. Proof-of-concept implementation of AutoPT platform. (A) Hardware setup consisting of an inverted optical microscope, CCD camera, motorized XY stage, micromanipulators, and petri dish holding coverslips with cells for phototransfection. (B) Software GUI for operating the AutoPT platform.

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 from 30 s down to approximately 8 s. This corresponds to a throughput of 450 cells/h, a 23× improvement from the current manual process. Again, assuming that all the cells (typically four to six) in the FOV can be processed with minimal increased computational overhead, a potential throughput of 2250 cells/h (>113× improvement) is estimated. Subsequent segmentation on images with four- to six-cell entities has indeed shown no marked increase in the overall processing time. However, in practice, this fully integrated program cannot be written in Matlab because the images from the confocal microscope, which are 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 to achieve these further throughput gains. Another option would be to add an additional optical port to the microscope or an external optical system that a compatible CCD camera could be mounted and hooked into the Matlab interface. Considering of both these options are areas of future work. 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 coverslip out of the way and store it in an organized manner while feeding in the next one to be processed, with as limited human interactions as possible.

Summary

Work toward fully automating the single-cell manipulation process of phototransfection is presented in this article. 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 23× improvement is possible with this implementation with room for improvement to greater than 110× described. Images of the phototransfected cell have been observed before and after the process, and a software tool has been 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 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 CM measurements, a 6× gain in the number of measurements made, and a more efficient and user-friendly software tool for cell morphological analysis.

Acknowledgments

The authors gratefully acknowledge funding from National Science Foundation Grant IIS-0413138, Department of Education Graduate Assistance in Areas of National Need (GAANN) Grant P200A060275, the Keck Foundation, and the National Institutes of Health Director's Pioneer Award Program DP1-OD-04117 to support this work and Kitty Wu for discussions on the manual phototransfection procedure.

Competing Interests Statement: The authors certify that all financial and material support for this research and work are clearly identified in the manuscript.

References

1. Bianco, P.; Robey, P. Stem cells in tissue engineering. *Nature* **2001,** *414(6859)*, 118–121.
2. Lovell-Badge, R. The future for stem cell research. *Nature* **2001,** *414(6859)*, 88–91.
3. Temple, S. The development of neural stem cells. *Nature* **2001,** *414(6859)*, 112–117.
4. Takahashi, K.; Yamanaka, S. Induction of pluripotent stem cells from mouse embryonic and adult fibroblast cultures by defined factors. *Cell* **2006,** *126*, 663–676.
5. Barrett, L. E.; Sul, J. Y.; Takano, H.; Van Bockstaele, E. J.; Haydon, P. G.; Eberwine, J. H. Region-directed phototransfection reveals the functional significance of a dendritically synthesized transcription factor. *Nat. Methods* **2006,** *3*, 455–460.
6. Sul, J.; Wu, C.; Zeng, F.; Jochems, J.; Lee, M.; Kim, T.; Peritz, T.; Buckley, P.; Cappelleri, D.; Maronski, D.; Kim, M.; Kumar, V.; Meaney, D.; Kim, J.; Eberwine, J. Transcriptome transfer produces a predictable cellular phenotype. *Proc. Natl. Acad. Sci. U.S.A.* **2009,** *106(18)*, 7624–7629.
7. Stevenson, D.; Gunn-Moore, F.; Campbell, P.; Dholakia, K. Review: single cell optical transfection. *J.R. Soc. Interface* **2010,** doi:10.1098/rsif.2009.0463 (published online 11 January 2010).
8. Brown, C. T. A.; Stevenson, D. J.; Tsampoula, X.; McDougall, C.; Lagatsky, A. A.; Sibbett, W.; Gunn-Moore, F.; Dholakia, K. Enhanced operation of femtosecond lasers and applications in cell transfection. *J. Biophotonics* **2008,** *1*, 183–199.
9. Tsampoula, X.; Garces-Chavez, V.; Comrie, M.; Stevenson, D. J.; Agate, B.; Brown, C. T. A.; Gunn-Moore, F.; Dholakia, K. Femtosecond cellular transfection using a nondiffracting light beam. *Appl. Phys. Lett.* **2007,** *91*, 053902.
10. Yao, C.-P.; Zhang, Z.-X.; Rahmanzadeh, R.; Huettmann, G. Laser-based gene transfection and gene therapy. *IEEE Trans. Nanobioscience* **2008,** *7(2)*, 111–119.
11. Baghdoyan, S.; Roupioz, Y.; Pitaval, A.; Castel, D.; Khomyakova, E.; Papine, A.; Soussaline, F.; Gidrol, X. Quantitative analysis of highly parallel transfection in cell microarrays. *Nucleic Acids Res.* **2004,** *32(9)*, e77.
12. Uchugonova, A.; Konig, K.; Bueckle, R.; Isemann, A.; Tempea, G. Targeted transfection of stem cells with sub-20 femtosecond laser pulses. *Opt. Express* **2008,** *16*, 9357–9364.
13. Hamilton, S.; Russo, M. Tutorial/workshop: introduction to laboratory automation. Proceedings of IEEE Conference on Automation Science and Engineering (CASE), Scottsdale, AZ; 2007.
14. Zhang, M.; Felder, R.; Kim, E.; Nelson, B.; Pruitt, B.; Zheng, Y.; Meldrum, D. Editorial: special issue on life science automation. *IEEE Trans. Autom. Sci. Eng.* **2006,** *3(2)*, 137–140.
15. Park, J.; Jung, S.; Kim, Y. Design and fabrication of an integrated cell processor for single embryo cell manipulation. *Lab Chip* **2005,** *5*, 91–96.
16. Huang, H.; Sun, D.; Mills, J.; Cheng, S. Integrated vision and force control in suspended cell injection system: towards automatic batch biomanipulation. Proceedings of IEEE International Conference on Robotics and Automation (ICRA), Pasadena, CA; 2008.
17. Kimura, Y.; Yanagimachi, R. Intracytoplasmic sperm injection in the mouse. *Biol. Reprod.* **1995,** *52*, 709–720.
18. Wang, W.; Liu, X.; Sun, Y. Autonomous zebrafish embryo injection using a microrobotic system. Proceedings of IEEE Conference on Automation Science and Engineering (CASE), Scottsdale, AZ; 2007.
19. Wang, W.; Sun, Y.; Zhang, M.; Anderson, R.; Langille, L.; Chen, W. A microrobotic adherent cell injection system for investigating intracellular behavior of quantum dots. Proceedings of IEEE International Conference on Robotics and Automation (ICRA), Pasadena, CA; 2008.
20. Desai, J.; Pillarisetti, A.; Brooks, A. Engineering approaches to biomanipulation. *Annu. Rev. Biomed. Eng.* **2007,** *9*, 35–53.
21. Sakaki, K.; Dechev, N.; Burke, R.; Park, E. Development of an autonomous biological cell manipulator with single-cell electroporation and visual servoing capabilities. *IEEE Trans. Biomed. Eng.* **2009,** *56(8)*, 2064–2074.
22. Najmabadi, P.; Goldenberg, A.; Emili, A. A scalable robotic-based laboratory automation system for medium-sized biotechnology laboratories. Proceedings of IEEE Conference on Automation Science and Engineering (CASE), Edmonton, Canada; 2005.
23. Choi, B.; Jin, S.; Shin, S.; Koo, J.; Ryew, S.; Kim, M.; Kim, J.; Son, W.; Ahn, K.; Chung, W.; Choi, H. Development of flexible laboratory automation platform using mobile agents in the clinical laboratory. Proceedings of IEEE Conference on Automation Science and Engineering (CASE), Washington, D.C.; 2008.
24. Meldrum, D. R.; Fisher, C. H.; Moore, M. P.; Saini, M.; Holl, M. R.; Pence, W. H.; Moody, S. E.; Cunningham, D. L.; Wiktor, P. J. ACAPELLA-5K, a high-throughput automated genome and chemical analysis system. *Proceedings of IEEE International Conference on Intelligent Robots and Systems (IROS)*, Las Vegas, NV; **2003,** *3*, 2321–2328.
25. Potsaid, B.; Finger, F.; Wen, J. Automation of challenging spatial-temporal biomedical observations with the adaptive scanning optical microscope (ASOM). *IEEE Trans. Autom. Sci. Eng.* **2009,** *6(3)*, 525–535.
26. Kaber. D.; Stoll, N.; Thurow, K. Human-automation interaction strategies for life science applications: implications and future research. Proceedings of IEEE Conference on Automation Science and Engineering (CASE), Scottsdale, AZ; 2007.
27. Kuncov-Kallio, J.; Kallio, P. Lab automation in cultivation of adherent cells. *IEEE Trans. Autom. Sci. Eng.* **2006,** *3(2)*, 177–186.
28. Makkapati, V.; Agrawal, R.; Acharya, R. Segmentation and classification of tuberculosis bacilli from ZN-stained sputum smear images. Proceedings of IEEE Conference on Automation Science and Engineering (CASE), Bangalore, India; 2009.
29. Makkapati, V. Improved wavelet-based microscope autofocusing for blood smears by using segmentation. Proceedings of IEEE Conference on Automation Science and Engineering (CASE), Bangalore, India; 2009.
30. Arrasate, M.; Finkbeiner, S. Automated microscope system for determining factors that predict neuronal fate. *Proc. Natl. Acad. Sci. U.S.A.* **2005,** *102(10)*, 3840–3845.
31. Geisler, T.; Ressler, J.; Harz, H.; Wolf, B.; Uhl, R. Automated multiparametric platform for high-content and high-throughput analytical screening on living cells. *IEEE Trans. Autom. Sci. Eng.* **2006,** *3(2)*, 169–176.

32. Neumann, B.; Held, M.; Liebel, U.; Erfle, H.; Rogers, P.; Pepperkok, R.; Ellenberg, J. High-throughput RNAi screening by time-lapse imaging of live human cells. *Nat. Methods* **2006,** *3(5)*, 385–390.
33. Conrad, C.; Erfle, H.; Warnat, P.; Daigle, N.; Lörch, T.; Ellenberg, J.; Pepperkok, R.; Eils, R. Automatic identification of subcellular phenotypes on human cell arrays. *Genome Res.* **2004,** *14*, 1130–1136.
34. Trucco, E.; Verri, A. *Introductory Techniques for 3-D Computer Vision*. Prentice Hall: Upper Saddle River, NJ; 1998.
35. Forsyth, D.; Ponce, J. *Computer Vision: A Modern Approach*. Prentice Hall: Upper Saddle River, NJ; 2003.
36. Boykov, Y.; Kolmogorov, V. An experimental comparison of mincut/max-flow algorithms for energy minimization in vision. *IEEE Trans. Pattern Anal. Mach. Intell.* **2004,** *26(9)*, 1124–1137.
37. Cappelleri, D. Flexible automation of micro and meso-scale manipulation tasks with applications to manufacturing & biotechnology. Ph.D. Dissertation. University of Pennsylvania, Philadelphia, PA; 2008.
38. Cheng, P.; Cappelleri, D.; Gavrea, B.; Kumar, V. Planning and control of meso-scale manipulation tasks with uncertainties. Proceedings of Robotics: Science and Systems, Atlanta, GA USA; 2007.