MATTOS, LEONARDO SERRA. Towards the Automation of Embryonic Stem Cell Microinjections into Blastocysts. (Under the direction of Edward Grant.)

The purpose of the research has been to increase the consistency and efficiency rates of blastocyst microinjections through automation. The research involved the design, implementation, and evaluation of a novel biomanipulation system that is a test-bed for applying intelligent control algorithms. The microinjection process was controlled from a computer via a joystick or by software controllers. These included real-time video processing for the acquisition of experimental data and control. Teleoperated microinjections under the control of both expert and novice operators showed that the system is effective, easy to use, and capable of eliminating the need for the extensive training of microinjection personnel. Experimental results showed that all operators obtained a microinjection success rate over 80%, demonstrating a significant improvement over the traditional manual microinjections. Furthermore, blastocysts injected using this system were more likely to develop to term, and to yield chimeras, than blastocysts injected using the traditional manual method. The experiments also highlighted common problems encountered during the blastocyst microinjection stage, allowing the design and development of effective control algorithms to guide the teleoperated and automatic microinjections. Overall, this research contributed to the full automation of blastocyst microinjection by: 1) significantly improving the microinjection process; 2) significantly improving the microinjection efficiency; 3) creating a new system design optimized for computer controlled microinjections; 4) implementing and evaluating speed-up methods.
that enable real-time template matching; 5) creating new algorithms to identify and analyze blastocyst images; 6) designing and conducting preliminary tests with control algorithms that automate the microinjection process.
TOWARDS THE AUTOMATION OF EMBRYONIC STEM CELL MICROINJECTIONS INTO BLASTOCYSTS

by

LEONARDO SERRA MATTOS

A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

ELECTRICAL ENGINEERING

Raleigh, North Carolina
May 2007

APPROVED BY:

Dr. Donald Bitzer

Dr. Troy Nagle

Dr. Edward Grant
Chair of Advisory Committee

Dr. John Muth

Dr. Randy Thresher
DEDICATION

To my family,

with all of my heart.
BIOGRAPHY

Leonardo Serra Mattos grew up in Brazil and began working with robots and automation systems while attending the Campinas’ Technical College (COTUCA-UNICAMP, Campinas, Brazil), from which he earned a Technical degree in Electronics in 1992. Leonardo maintained his involvement in the automation area while attending college at the University of Sao Paulo (USP, São Carlos, Brazil), where he earned a Bachelor of Science degree in Electrical Engineering in 1998.

After college Leonardo gained industry exposure and hands-on experience with robots and automated machines while working for Fanuc America Corporation in the United States. His interests in research and development of new technologies, however, took him back to school, and Leonardo started graduate studies in 2001.

Leonardo joined the Center for Robotics and Intelligent Machines (CRIM) in 2002, where he got involved in several multi-disciplinary research projects. These included research in biorobotics, medical robotics, e-textiles, evolutionary robotics, instrumentation, acoustic arrays, rehabilitation engineering, and remote sensing.

Mattos received his Master of Sciences degree in Electrical Engineering from the North Carolina State University (NCSU, Raleigh, USA) in 2003, after which he continued his graduate studies pursuing a Doctor of Philosophy degree also in Electrical Engineering. His efforts towards that degree were concluded with this dissertation.

Leonardo Mattos is a member of IEEE, IEEE Robotics and Automation Society, and Eta Kappa Nu. He has been certified as Engineering Intern (E.I.) by the National Council of Examiners for Engineering and Surveying (NCEES) since 2003.
ACKNOWLEDGEMENTS

I would like to express my deep gratitude to my advisor, Dr. Edward Grant, and to Dr. Randy Thresher for their continuous involvement and support during this research. Many thanks to Kim Kluckman as well, for her ideas, enthusiasm and all of the help provided during the setup and experimentation phases of this research.

I would also like to thank Dr. John Muth and Dr. Troy Nagle, from the NCSU’s ECE Department, and Dr. Marzluff and Dr. Corvinus, from the UNC’s School of Medicine, for the acquisition of equipment employed in this research.

In addition, I would like to thank all members of the Center for Robotics and Intelligent Machines for their friendship, help, and support.
# TABLE OF CONTENTS

LIST OF TABLES .........................................................................................................................................ix

LIST OF FIGURES ..........................................................................................................................................x

1. INTRODUCTION TO BIOMANIPULATION ................................................................................................................................. 1

1.1. BIOMANIPULATION TECHNIQUES........................................................................................................................................... 3

  1.1.1. Contact Techniques for Biomanipulation .......................................................................................................................... 4

    1.1.1.1. Manual Mechanical Micromanipulation ......................................................................................................................... 4

    1.1.1.2. Atomic Force Microscopy/Manipulation ....................................................................................................................... 5

  1.1.2. Non-Contact Techniques for Biomanipulation .......................................................................................................................... 7

    1.1.2.1. Electrowetting ........................................................................................................................................................................ 7

    1.1.2.2. Dielectrophoresis ................................................................................................................................................................. 9

    1.1.2.3. Laser Trapping .............................................................................................................................................................. 10

    1.1.2.4. Electro-Rotation .................................................................................................................................................................. 11

    1.1.2.5. Thermal Gelation ......................................................................................................................................................... 11

1.2. MICROINJECTION TECHNOLOGY ....................................................................................................................................... 12

1.3. BIOMANIPULATION AUTOMATION .................................................................................................................................... 12

  1.3.1. Computer Vision for Biomanipulations ................................................................................................................................. 13

  1.3.2. Foremost Efforts Toward Embryo Biomanipulation Automation ............................................................................................... 15

1.4. ES CELLS AND THE PROCESS OF GENE TRANSFER INTO THE MOUSE  ........................................................................................................... 18

  1.4.1. Details of the Blastocyst Microinjection Process .................................................................................................................... 20

1.5. CONTRIBUTIONS TOWARDS BIOMANIPULATION AUTOMATION ............................................................................................... 22

2. THE SEMI-AUTOMATED MICROINJECTION SYSTEM .................................................................................................................. 25

  2.1. MAIN COMPONENTS ........................................................................................................................................................... 25

  2.2. SYSTEM CONFIGURATION .................................................................................................................................................... 28

  2.3. SOFTWARE UNITS ................................................................................................................................................................. 29

  2.4. ROBOT CONTROL .................................................................................................................................................................. 32

  2.5. XY STAGE ............................................................................................................................................................................ 34

  2.6. THE INTERACTIVE USER INTERFACE .................................................................................................................................... 35

  2.7. CONCLUSIONS ....................................................................................................................................................................... 38

3. EXPERIMENTS AND ISSUES RELATED TO SEMI-AUTOMATED BLASTOCYST MICROINJECTION ................................................................................................................................. 40

  3.1. EVALUATION OF THE ROBOT CONTROL SYSTEM .................................................................................................................. 41

  3.2. SEMI-AUTOMATED MICROINJECTION EXPERIMENTS ........................................................................................................ 43

  3.3. ANALYSIS OF FAILED MICROINJECTIONS .............................................................................................................................. 50

  3.4. RULES FOR BLASTOCYST MICROINJECTIONS ........................................................................................................................ 54

  3.5. BLASTOCYST ORIENTATION ADJUSTMENTS .................................................................................................................................. 56

    3.5.1. Blastocyst rotation method based on fluid flow ................................................................................................................... 57

    3.5.2. Push-pull blastocyst rotation method ................................................................................................................................. 58

    3.5.3. Alternative method for blastocyst rotation......................................................................................................................... 59

  3.6. CONCLUSIONS ....................................................................................................................................................................... 59

4. VIDEO PROCESSING ALGORITHMS .................................................................................................................................... 62

  4.1. OVERVIEW ........................................................................................................................................................................... 62

  4.2. BLASTOCYST PROCESSING ..................................................................................................................................................... 64

    4.2.1. Finding the blastocyst ...................................................................................................................................................... 64
8.3.3. Automatic Blastocyst Release ................................................................. 145
8.3.4. Automatic Blastocyst Orientation ......................................................... 146
8.3.5. Automatic Blastocyst Collection .......................................................... 147
8.3.6. Automatic Motion Control ................................................................. 148
  8.3.6.1. Visual Servoing .............................................................................. 149
  8.3.6.2. Automation Algorithm ................................................................. 151
8.3.7. ES Cells Delivery ................................................................................ 152
  8.3.7.1. Proposed Automation Algorithm .................................................. 152
  8.3.7.2. Proposed ES Cells Counter Algorithm .......................................... 153
8.3.8. Automatic Microinjections ................................................................. 155
8.3.9. Process Automation ............................................................................ 156
8.4. Conclusions ......................................................................................... 157

9. EXPERIMENTS WITH AUTOMATED MICROMANIPULATION ROUTINES .......... 160
  9.1. Automatic Blastocyst Delivery Trials ................................................... 160
  9.2. Automatic Blastocyst Capture Trials ................................................... 162
  9.3. Automatic Blastocyst Release and Collection Trials .............................. 163
  9.4. Automatic Blastocyst Orientation Trials ............................................. 164
  9.5. Automatic Motion Control Trials ....................................................... 164
  9.6. Automatic Microinjection Trials ......................................................... 164
  9.7. Conclusions ....................................................................................... 169

10. SUGGESTIONS FOR IMPROVEMENTS AND FUTURE RESEARCH ...................... 172
  10.1. Microinjection Equipment Upgrade .................................................... 172
  10.2. Vision System Improvements ............................................................. 173
  10.3. Fully Automated Blastocyst Microinjections ....................................... 175
  10.4. Intelligent System Control ................................................................. 176
  10.5. System Configuration Improvements ............................................... 177
  10.6. Related Areas for Future Research ..................................................... 178

11. CONCLUSIONS .................................................................................... 180

12. REFERENCES ....................................................................................... 183

APPENDICES ........................................................................................................ 192

APPENDIX 1. THE BLASTOWORKS SOFTWARE ................................................... 193
  A1.1. SOFTWARE STRUCTURE OVERVIEW ................................................. 193
  A1.2. THE USER INTERFACE MODULE ..................................................... 193
  A1.3. IMAGE MANAGEMENT ...................................................................... 194
    A1.3.1. Image Sources .............................................................................. 194
      A1.3.1.1. Live Video ............................................................................. 195
      A1.3.1.2. Simulated Video .................................................................. 196
      A1.3.1.3. Recorded Video .................................................................. 199
      A1.3.1.4. Pictures .............................................................................. 200
    A1.3.2. Image Adjustments ...................................................................... 200
      A1.3.2.1. Image Noise ....................................................................... 201
    A1.3.3. Image Processing ...................................................................... 201
    A1.3.4. Image Flow ............................................................................... 203
  A1.4. MOTION MANAGEMENT .................................................................... 203
    A1.4.1. Joystick Interface ..................................................................... 204
    A1.4.2. Robot Interface ....................................................................... 205
    A1.4.3. Syringes and Piezo Interface ..................................................... 206
    A1.4.4. XY Stage Interface .................................................................. 207
  A1.5. FILE MANAGEMENT ....................................................................... 208
LIST OF TABLES

TABLE 2.1. DESCRIPTION CHARACTERS .................................................................31
TABLE 2.2. COMMAND BYTES FOR THE MICROMETER SYRINGES AND PIEZO INJECTOR .................................................................32
TABLE 3.1. RESULTS FROM TELEOPERATED MICROINJECTIONS .................................................................................................................................45
TABLE 3.2. MANUAl MICROINJECTION DATA FROM EXPERT OPERATORS ......................................................................................................................48
TABLE 3.3. AMC’S MANUAL MICROINJECTION DATA FROM 2002 – 2006 ......................................................................................................................50
TABLE 5.1. SAMPLE RESULTS FROM THE ANALYSIS OF 1000 SIMULATED IMAGES FOR EACH NOISE CONDITION AND FOR SIMULATED BLASTOCYSTS WITH A 35 PIXELS RADIUS ......................................................................................87
TABLE 5.2. SAMPLE EXPERIMENTAL RESULTS FROM THE ANALYSIS OF 300 VIDEO IMAGES OF OOCYTES .........................................................91
TABLE 5.3. EXPERIMENTAL RESULTS FROM THE ANALYSIS OF 122 REAL VIDEO IMAGES OF BLASTOCYSTS .......................................................................92
TABLE 5.4. EVALUATION OF THE INJECTION SECTOR CLASSIFICATION ALGORITHM .....................................................................................................93
TABLE 6.1. SUMMARY OF THE TEMPLATE MATCHING SPEED-UP TECHNIQUES ........................................................................................................107
TABLE 9.1. RESULTS FROM AUTOMATIC AND MANUAL MICROINJECTIONS .......................................................................................................................168
TABLE A2.1. ICB SPECIFICATIONS ..................................................................................210
TABLE A2.2. ICB CONNECTOR DESCRIPTIONS ......................................................................211
TABLE A2.3. ICB COMMANDS ............................................................................................213
TABLE A2.4. ICB STEPPER COMMANDS .................................................................................214
LIST OF FIGURES

FIG. 1.1. MANUAL MICROMANIPULATOR MADE BY LEICA MICROSYSTEMS ..........................................................5
FIG. 1.2. CROSS SECTION OF THE ELECTROWETTING DEVICE DEVELOPED BY FAIR ET AL. ..........................8
FIG. 1.3. SET-UP USED FOR BLASTOCYST INJECTIONS ..................................................................................21
FIG. 1.4. BLASTOCYST MICROINJECTION ..................................................................................................22
FIG. 2.1. THE BlASTOCYST MICROINJECTION SYSTEM SETUP .........................................................................27
FIG. 2.2. THE MICROINJECTION SYSTEM CONFIGURATION ............................................................................29
FIG. 2.3. FRACTION OF AN EXPERIMENTAL DATA FILE .............................................................................31
FIG. 2.4. EXPONENTIAL FUNCTION APPLIED TO THE JOYSTICK COMMANDS ..................................................33
FIG. 2.5. TYPICAL PETRI DISH SETUP ........................................................................................................35
FIG. 2.6. JOYSTICK FUNCTION ASSIGNMENTS ..........................................................................................37
FIG. 2.7. THE INTERACTIVE USER INTERFACE ................................................................................................38
FIG. 3.1. PATH-FOLLOWING EXPERIMENTS ................................................................................................42
FIG. 3.2. MICROINJECTION SUCCESS RATES FROM EXPERT TRIALS ......................................................46
FIG. 3.3. MICROINJECTION SUCCESS RATES FROM THE AMATEURS TRIALS ........................................47
FIG. 3.4. BIRTH RATE FROM IMPLANTED BLASTOCYSTS ..............................................................................49
FIG. 3.5. YIELD OF CHIMERA FROM IMPLANTED BLASTOCYSTS ...............................................................49
FIG. 3.6. FAILED MICROINJECTIONS ...........................................................................................................52
FIG. 3.7. STEPS OF A SUCCESSFUL MICROINJECTION ..................................................................................56
FIG. 3.8. BLASTOCYST ROTATION METHOD BASED ON FLUID FLOW ..............................................................58
FIG. 4.1. OVERVIEW OF THE VIDEO PROCESSING ALGORITHM .................................................................63
FIG. 4.2. EXAMPLE OF THE APPLICATION OF HOUGH TRANSFORM TO LOCATE THE BLASTOCYST ..............65
FIG. 4.3. PSEUDO-CODE FOR THE CLASSIFICATION OF ACCUMULATOR PEAKS ...........................................66
FIG. 4.4. PSEUDO-CODE FOR TESTING THE PRESENCE OF A BLASTOCYST AT A TRUE PEAK LOCATION ..........67
FIG. 4.5. PSEUDO-CODE FOR FINDING THE BEST INJECTION AREA ON THE BLASTOCYST .............................68
FIG. 4.6. ORIGINAL AND PROCESSED IMAGE OF A REAL BLASTOCYST ..........................................................69
FIG. 4.7. PSEUDO-CODE FOR QUALIFYING THE BLASTOCYST ORIENTATION ................................................70
FIG. 4.8. ORIGINAL AND PROCESSED IMAGES OF BLASTOCYSTS ................................................................71
FIG. 4.9. TEMPLATE ACQUISITION ................................................................................................................75
FIG. 5.1. BLASTOCYST SIMULATION METHOD ...............................................................................................78
FIG. 5.2. SYNTHETIC IMAGES WITH DIFFERENT NOISE CHARACTERISTICS ................................................79
FIG. 5.3. IMPACT OF GAUSSIAN NOISE ON THE BLASTOCYST-FINDING ALGORITHM .................................81
FIG. 5.4. IMPACT OF IMPULSE NOISE ON THE BLASTOCYST-FINDING ALGORITHM .....................................83
FIG. 5.5. IMPACT OF REDUCING THE PERCENTAGE OF BLASTOCYST PIXELS TRANSFERRED TO THE TEST IMAGES ON THE BLASTOCYST-FINDING ALGORITHM ..........................................................84
FIG. 5.6. IMPACT OF GAUSSIAN BLUR ON THE BLASTOCYST-FINDING ALGORITHM .................................86
FIG. 5.7. IMPACT OF NOISE ON THE MEAN LOCALIZATION ERROR OF THE BLASTOCYST-FINDING ALGORITHM APPLIED TO REAL OOCYTE IMAGES .................................................................................90
FIG. 5.8. BLASTOCYST IMAGES AND THE PROCESSED IMAGES SHOWING THE RESULTS OF THE LOCALIZATION AND INJECTION SECTOR SELECTION ALGORITHMS ..................................................................................92
FIG. 6.1. TEMPLATE MATCHING SPEED-UP AS A FUNCTION OF THE REDUCTION IN THE SEARCH AREA ........ 99
FIG. 6.2. WINDOW-BASED TRACKING ........................................................................................................100
FIG. 6.3. TYPICAL PIPETTE TEMPLATES AND THEIR CORRESPONDING EDGE-IMAGES ................................ 105
FIG. 7.1. SIMULATED INJECTION NEEDLE TEMPLATE AND ITS EDGE-IMAGE USED FOR MATCHING ............110
FIG. 7.2. SYNTHETIC IMAGES WITH DIFFERENT NOISE CHARACTERISTICS ................................................111
FIG. 7.3. IMPACT OF GAUSSIAN NOISE ON THE CROSS-CORRELATION TEMPLATE MATCHING ..................113
FIG. 7.4. IMPACT OF IMPULSE NOISE ON THE CROSS-CORRELATION TEMPLATE MATCHING .....................115
FIG. A1.8. SIMULATED MOTIONS CONFIGURATION TOOL. .................................................................199
FIG. A1.9. VIDEO CONTROLLER TOOL WINDOW. ..............................................................................199
FIG. A1.10. LIVE VIDEO ADJUSTMENTS. ............................................................................................200
FIG. A1.11. IMAGE CORRUPTION BY CUSTOM NOISE LEVELS. .........................................................201
FIG. A1.12. IMAGE PROCESSING MENUS. ...........................................................................................202
FIG. A1.13. FLOW OF IMAGES WITHIN THE BLASTOWORKS SOFTWARE. ...........................................203
FIG. A1.14. OVERVIEW OF THE MOTION MANAGEMENT SYSTEM. .......................................................204
FIG. A1.15. JOYSTICK INTERFACE. ......................................................................................................204
FIG. A1.16. ROBOT INTERFACE. ..........................................................................................................205
FIG. A1.17. ICB INTERFACE. ..............................................................................................................206
FIG. A1.18. XY STAGE INTERFACE. .....................................................................................................207
FIG. A2.1. TOP VIEW OF THE ICB SHOWING CONNECTOR LOCATIONS. ...............................................212
FIG. A2.2. BOTTOM VIEW OF THE ICB. ...............................................................................................212
FIG. A2.3. CIRCUIT DIAGRAM FOR THE ICB (PART 1 OF 4). .............................................................216
FIG. A2.4. CIRCUIT DIAGRAM FOR THE ICB (PART 2 OF 4). .............................................................217
FIG. A2.5. CIRCUIT DIAGRAM FOR THE ICB (PART 3 OF 4). .............................................................218
FIG. A2.6. CIRCUIT DIAGRAM FOR THE ICB (PART 4 OF 4). .............................................................219
1. INTRODUCTION TO BIOMANIPULATION

Current research into genetics and associated biomedical areas relies on the use of genetically modified mice for discovering gene functions, and for the understanding of how gene defects lead to the development of diseases [1]. Gene-targeted mice, for example, are often used as models of a wide range of serious human afflictions, including diabetes, arteriosclerosis, hypertension, Alzheimer’s disease, and cancer [2].

Gene-targeted mice are typically created by injecting genetically altered mouse embryonic stem (ES) cells into early mouse embryos at the blastocyst stage. These manually controlled operations are called blastocyst microinjections, and their success depends, to a large extent, on meticulous manipulations of the delicate cells and embryos. Therefore, the skills of the person performing the operations significantly impact the results obtained from the microinjections.

According to published literature [3], operators need up to one full year of training to become proficient at injecting blastocysts. However, even with extensive training, low survival rates of microinjected cells are encountered, commonly between 40% and 70% [4].

The problems that affect the efficiency of blastocyst microinjections, as well as other biological micromanipulation tasks, are related to human errors and to the lack of repeatability of the operations [5, 6, 7, 8]. Thus, one possible way to address these problems is to automate the manipulations. Therefore, a major goal of this research was progress towards a fully automated microinjection system.
A pioneer in the area of embryo biomanipulation automation is Ogawa. Together with Takahashi, Mizuno, Kashiwazaki, Yamane and Narishige, Ogawa developed a computer-controlled system for the manipulation of eggs and early embryos in 1985 [5]. Their system depended heavily on operator inputs. The motion control of the manipulator worked in open-loop and was based on manually defined starting and ending positions. Nevertheless, the system successfully accomplished tasks such as bisection and microinjection. Ogawa and his fellow researchers continued to work on automation improvements, and in 1992 they reported on a new system that automated the subzonal insemination of mouse ova [6]. In this case they used computer vision techniques (i.e. template matching) to locate the holding and injecting pipettes at the beginning of each microinjection procedure, thereafter the microinjection operation was performed in open-loop.

In the past two decades intense efforts have been made to automate the positioning [9, 10, 11], holding [12, 13] and injection [14, 15] of biological cells. Recently, automatic visual tracking of cells have been reported [16, 17], and, in 2002, Sun and Nelson introduced visual servoing as a major technique to enable closed-loop control in an automated cell injection system [7]. Their paper describes an automated system for embryo pronuclear microinjection based on a single general-purpose microrobot and on a hybrid visual servoing scheme. Similarly, Zhao et al. [8] developed a system using two micromanipulators which, following user inputs, could also autonomously capture the target cells for injection.
Subsequent efforts to progress to full automation have encountered numerous difficulties. This has led to a re-evaluation of the basic premises of full automation and an apparent step back from that concept. Generally, there has been a move towards the development of augmented reality systems to aid and improve manually controlled microinjections [18, 19, 20]. Many recent studies have also been concerned with a better characterization of the cell membranes and of the forces involved in the microinjections [3, 21, 22, 23]. This may be due to the fact that, as stated by Arai, “micromanipulation tasks are versatile. So it is difficult to realize a full automation system” [24]. For this reason, much of the current focus is on classifying the basic operations.

1.1. Biomanipulation Techniques

Biomanipulation, as described by Sun and Nelson [7], “entails such operations as positioning, grasping, and injecting material into various locations in cells.” Consequently, the biomanipulation field is vast and includes a variety of ways in which cells can be manipulated manually or automatically. At the same time, all biomanipulation techniques can be classified as belonging to one of two categories based on whether the cell is physically touched, or not, during the procedures. The two categories are therefore referred to as contact techniques and non-contact techniques. This chapter briefly describes the major devices and techniques currently used (or under development) for biomanipulation.
1.1.1. Contact Techniques for Biomanipulation

Contact techniques use mechanical devices for cell manipulation. These give fine position control, typically with a resolution in the order of a few micrometers. They also typically use small application-specific end-effectors for precise manipulation. For example, the end-effector can be a sharp micro-blade for bisections, or the smooth tip of a holding pipette.

The range of devices used in contact biomanipulations covers standard manual micromanipulators for cell injections to computer controlled motorized platforms. An overview of these devices is presented below.

1.1.1.1. Manual Mechanical Micromanipulation

Manual mechanical micromanipulation is the current standard in both research and commercial laboratories working with cells or small microorganisms. In these systems, mechanical devices convert large movements of a human hand into fine end-effector movements.

Examples of procedures that typically use manual micromanipulators for biomanipulation include intra-cytoplasmic sperm injection (ICSI), pronuclei microinjection, and embryonic stem cells microinjection into blastocysts. These applications use glass pipettes with very fine tips as the end-effectors, and the operations require delicate motion control under a microscope.
A manual micromanipulator commonly used in the procedures described above is shown in Fig. 1.1. It has controls for both coarse and fine positioning, including a mechanical joystick that allows precise motions in the XY plane.

Although precise, micromanipulators like the one shown in Fig. 1.1 are highly susceptible to vibrations emanating from the operator’s hand, so motorized and oil-driven versions have been developed to decouple the joystick movements from the micromanipulators.

![Fig. 1.1. Manual micromanipulator made by Leica Microsystems. Picture reprinted with permission from [69].](image)

### 1.1.1.2. Atomic Force Microscopy/Manipulation

The atomic force microscope (AFM) was initially developed to “measure ultra small forces on particles as small as single atoms” [25]. Since its introduction in 1986 by
Binnig, Quate and Gerber, it has been widely used for surface imaging, for the study of surface properties, for surfaces modifications, and for manipulation of tiny structures like cells.

AFMs map the topography of samples by scanning the surfaces using a very sharp tip. AFM images are created during this scanning process by measuring the force between the scanning tip and the surface of the sample. This force is generally held constant and at a small value throughout the scanning process. This is accomplished by using a feedback control system and piezoelectric actuators that continuously adjust the height of the scanning tip. So, when the AFM is used for biomanipulations, it is classified as a contact type device.

According to Mokaberi and Requicha [26], “particles with diameters of approximately 10 nm are manipulated routinely with AFMs.” However, most manipulations are carried out using open loop control, i.e., after each modification is performed a new scan is required to visualize the altered state. For this reason, research is being carried out to further develop AFM technology to include calibration and compensation methods that update AFM images in real-time [27, 28].

The advantage of AFM’s over other types of scanning probe microscopes, e.g., scanning tunneling microscope (STM), is that the AFM can image electrically conductive and non-conductive samples alike, both under ambient conditions and in solution. This makes an AFM useful for biomanipulations and for studying the physical characteristics of biological specimens under physiological conditions [29]. However, problems arise in such cases because biological structures are often soft and compliant. Therefore, even
when very small forces are applied to the biological structures by the AFM tip, it has the potential to distort or damage cell membranes [29]. Research related to this problem is in progress and has already generated new types of microscopes, including AFMs with tapping modes, photonic force microscopes (PFM), and a 3D magnetic force microscope (3DMFM) [30].

1.1.2. Non-Contact Techniques for Biomanipulation

The non-contact techniques for biomanipulation are mainly used to translate or to rotate cells. Within this category there are five techniques that deserve special attention: 1) electrowetting, 2) dielectrophoresis, 3) laser trapping, 4) electro-rotation, and 5) thermal gelation. Of these five, only laser trapping has been shown to provide a strong enough holding force to enable further manipulation such as microinjection. However, it is not usually applied to cells because the strong laser light has the potential to damage or induce mutations on the genetic materials [7].

1.1.2.1. Electrowetting

The electrowetting method for fluid manipulation was first introduced in 1981 by Beni and Hackwood [31] as a novel technology to create displays. This method has recently found applications in biomanipulation and is considered the major technology in the field of digital microfluidics [32, 33].

Electrowetting is an electrostatic method that manipulates small quantities of fluids through direct electrical control of the fluid’s surface tension. It “requires no moving parts
or fixed channels, is self-contained, consumes little power, and imposes minimal constraints upon the fluid being pumped” [32]. According to the reference, this method only requires the fluid to be polarizable and conductive.

The electrowetting device developed by Pollack, Fair and Shenderov [32] uses two sets of planar electrodes placed close to each and with enough space for a droplet to move between them. One of the electrodes is a single continuous ground plane; the other is an array of independently addressable electrodes. When voltage is applied to the electrode directly below the droplet, it induces a charge on the surface of the droplet near that electrode. If part of that droplet is also above a neighboring grounded electrode, electrostatic forces alter the angle of contact between the droplet and the substrate, and the droplet is attracted to the charged electrode. A schematic drawing of their system is shown in Fig. 1.2.

![Fig. 1.2. Cross section of the electrowetting device developed by Fair et al. (reprinted with permission from [32]).](image-url)
Electrowetting for the manipulation of cells has one major drawback: only relatively large quantities of fluid have been manipulated using this method. Recently, droplets with volume as low as 0.4µL were manipulated using this method [63], but this still is a large volume when compared the volume of a blastocyst (0.5236 ηL). Nonetheless, it is a method worth noting since an automatic cell manipulation system will need to move samples around.

1.1.2.2. Dielectrophoresis

A concise description of the dielectrophoresis phenomenon was provided by Böhringer [34]: “In dielectrophoresis (DEP), neutrally charged objects are first polarized by an electric field, and then experience a net force due to the field. This force can only be non-zero if a field gradient exists, i.e., the positively and negatively polarized regions of the object occupy areas of different field strengths. If that object has stronger polarization than the surrounding medium, it is pulled towards the areas of higher field strength (this is called positive DEP). Alternatively, if the surrounding medium has a higher polarization, the object is pushed towards areas of lower field strength (negative DEP). DEP can be considered the electrostatic analogy of induced magnetism. One common example for DEP are charged clothes that attract (neutral) lint particles.”

Dielectrophoresis has been used in cell micromanipulation mainly as a technique to move groups of cells around. However, the Dielectrophoresis Group [35] has shown that the method can also be used to trap, focus, fractionate and isolate cells. Furthermore, they claim that this technology can be used to selectively manipulate individual cells within
their carrier medium, or hold cells in place while the carrier medium is replaced. Separately, Vykoukal and colleagues [36] have been working on applications of dielectrophoresis for Micro Total Analysis Systems (µTAS). Their reported work includes a flow cytometer with DEP for particle focusing and a system to detect malaria [36], [37].

One of the best examples of an application of dielectrophoresis for manipulating cells is the system developed by Arai et al. [38]. Their system combines both a laser trap and dielectrophoresis to effectively separate a single cell from a group of cells held in solution. The laser trap is used to precisely hold and move the desired cell to an extraction port, while dielectrophoresis is used to keep the other cells away from the laser focal point.

1.1.2.3. Laser Trapping

Ashkin [39], [40] demonstrated the use of laser trapping to manipulate cells in 1970. This technique is based on laser radiation pressure, a condition created by Fresnel reflection and refraction of laser light on the interface of an object and a medium of different index of refraction. Laser trapping uses a focused laser beam to create “an optical trap in which the lateral trapping force moves a cell in suspension towards the center of the beam. The longitudinal trapping force moves the cell in the direction of the focal point” [7].

As mentioned previously, laser trapping has not yet been proven to be safe for cell manipulation because laser light can damage the cell or induce mutations on the genetic materials. To overcome this problem, Arai and colleagues proposed and implemented a method for indirect cell manipulation based on laser trapped micro-tools [41], [42].
1.1.2.4. **Electro-Rotation**

Electro-rotation is the phenomenon that causes particle rotation when such particle is within a rotating electric field. This phenomenon is the basis of a non-contact micromanipulation technique of the same name, and which has been shown to be applicable to cell manipulation since 1960 [43], [10].

Electro-rotation of a particle, or cell, is realized by controlling the phase and magnitude of the electric field surrounding the particle. If this field is rotated “at a fast enough speed, a phase delay is generated in the polarized charge induced on the particle. This creates torque, which causes the particle to rotate” [44].

Applications of this technology have been used to study both the motor characteristics of bacteria [45] and the dielectric property of cells [46]. More recently, researchers have started to apply electro-rotation to orientate cells prior to microinjection procedures, as in the system developed by Toshiro [44].

1.1.2.5. **Thermal Gelation**

The use of photo-crosslinkable resin and thermal gelation is being investigated for capturing single cells. Arai [47] introduced the idea for this technology in 2003 and has since demonstrated and improved the method. Recently reported systems can fix a target cell in a gel and remove obstacles, or other cells, by circulating the carrier fluid [12].
1.2. Microinjection Technology

Cell injections were traditionally performed using manual micromanipulators, with the required injection force being provided by the operator. In this case, an injection pipette with a sharp tip was necessary for successful cell penetration, and the operator had to be very cautious to avoid damaging the embryo. Nevertheless, this traditional method worked. On the other hand, the success rates of the microinjections were often low because of difficulties penetrating the embryo membranes, which frequently caused lethal damage [14].

A more successful way to perform cell microinjections uses piezo-driven micropipettes. This technique was first used in 1990 by Koch and Büschges [48] to improve cell impalement. However, it was Kimura and Yanagimachi [14] who popularized the technique by applying it to intracytoplasmic sperm injection in mouse oocytes in 1995.

Nowadays, piezo-driven microinjectors are commercially available from several companies. In the case of blastocyst microinjections, the current standard is to use piezo-injectors that include vibration modes. A common example is the Piezo-Drill™ system by EXFO Life Sciences [70].

1.3. Biomanipulation Automation

In order to overcome some of the problems associated with manual micromanipulations, i.e., the need for extensive training of operators, human errors,
vibrations and sterility, automated biomanipulation systems have been proposed. However, one of the main difficulties in realizing such systems is the acquisition of reliable feedback information for automatic control. Depending on the application, the required feedback parameter may be a force, or a position, or both. Consequently, automated biomanipulation systems typically employ visual and force feedback [19, 22, 23].

Several techniques have been proposed for obtaining feedback in cell manipulation tasks. These include visual processing for the localization of cells and pipettes [49, 7, 17], and the use of piezoelectric sensors for force measurements [19, 22, 23, 50].

More recently, research towards automating biomanipulations have turned to developing systems that use visual and force feedback to create augmented reality environments. In these systems, the operator looks at a computer display and is able to manipulate cells, or other structures, using a computer mouse [44] or a haptic device [19, 28, 51]. With haptic feedback, the manipulation effectiveness is said to improve because the operator can feel the force being applied to the structures during manipulation.

1.3.1. Computer Vision for Biomanipulations

Visual feedback is a key element for ensuring successful cell manipulations. Traditionally, operators perform the entire manipulation process looking at the samples through the eyepieces of a microscope. However, this is changing with the advent of microscopes that incorporate high quality CCD cameras, thereby enabling biomanipulation based on video displayed on monitors.
The use of cameras for microscopy not only improves the working conditions for the operator but also facilitates the automation of the cell manipulation process. It provides invaluable sensory feedback that can be used to automatically determine the position of the biomanipulation pipettes, the position of the cells, and to control the motion of these objects.

In automated systems for biomanipulation, like the one developed by Sun and Nelson [7], a technique called visual servoing is often applied. In this case, visual information is used to control the position and speed of the injection needle using a mapping from the task space coordinates to the image space coordinates. This way, displacements and velocities calculated from visual information are transformed into displacements and velocities of the real objects.

The use of visual feedback to improve the control of robots was introduced by Shirai and Inoue [52] in 1973. In the 1980s the first visual servoing systems started to be reported in the literature [53]. Improvements from these early days include the development of systems with stereo vision for enhanced control [49] and the creation of innumerous algorithms for signal and image processing.

One of the main objectives of image processing algorithms for visual servoing is the identification and tracking of the objects of interest within the camera’s field of view. This constitutes a great challenge because such algorithms need to be fast and robust [53], and this is not easily accomplished because of limited processing power and because of the dynamics of real-time video. In addition, focusing the images is an issue in biomanipulation since focus is crucial to ensuring correct identification of objects and
boundaries. In microscope systems focus can even be used to check if objects are on the same Z-plane.

Auto-focusing algorithms can be implemented in association with visual servoing systems to ensure good focus during operation. On the other hand, focus adjustment is often viewed as an initial calibration procedure in biomanipulations because the focal level is typically fixed during the operations. Nonetheless, some applications require frequent changes to the Z-plane of the pipettes and consume significant amounts of time on this operation. In such cases, the focus of the images can be used to precisely return the pipettes to its desired manipulation height, as demonstrated by Sun and Nelson [7].

### 1.3.2. Foremost Efforts Toward Embryo Biomanipulation Automation

Research towards the automation of biomanipulation tasks has been very active in recent years, following and supporting the rapid growth of biomedical research. The increasing interest in this area is reflected by the increasing number of published scientific works. The following research papers deserve a special mention because of their key contributions toward the automation of embryo manipulations.

The research conducted by Sun and Nelson in 2002 [7] advanced the biomanipulation automation area by incorporating visual servoing as the major technique to enable autonomous cell injections. Their application task was embryo pronuclei DNA injection, which consists of injecting DNA material inside the nucleus of an embryo. The research resulted in the automation of only the insertion and extraction of the needle from the embryo. However, their control system design combined feedback from vision
processing and precision positioning and proved to be an effective way of controlling the injection pipette motions. This work opened new areas of research related to the application of image processing techniques in biomanipulation research, including the use of visual feedback in augmented reality systems [19, 20] and in cooperative systems [18, 4] for cell manipulations. In addition, this work proved that automated cell microinjection is possible and can improve the success rate of such operations.

On the other hand, the system developed by Sun and Nelson required the presence of an operator at the microscope stage to perform tasks such as selecting and collecting embryos for injection, and controlling the actual dispensing of DNA into the nucleus of the embryos. Consequently, the problems related to the need of extensive operator training were not completely addressed since their system depended on direct manual control by a human operator.

The development of a biomanipulation system without direct operator control was researched by Glasgow et al. [54]. Their work advanced biomanipulation automation through research into the use of microfluidics as the major technique of transport of embryos without direct human control. Their research produced a microfluidic system that was capable of transporting individual mouse embryos through a network of microfluidic channels. This development was important because it combined microfabrication and microfluidics research, and applied it to embryo biomanipulation. Furthermore, it proposed a method that can be used to increase the speed and repeatability of the biomanipulations.
The research of Glasgow et al. was advanced by Park et al. [55], who used microfluidics not only for transport, but also for the isolation and immobilization of embryos. Their research resulted in the development of a microfluidic system optimized for the automatic microinjection of mouse embryos, including microelectrodes for orientation adjustments through electrorotation. Therefore, despite the fact that actual microinjection trials were not reported, their research demonstrated that a lab-on-a-chip device was possible for embryo manipulations.

Another major contribution towards the development of automated biomanipulation systems was provided by Arai et al. [56]. Their research concentrated on the development and use of force sensors, force feedback and virtual reality environments to improve teleoperated biomanipulations. The result of their efforts was the development of the first system in which an operator could feel the force being applied to biological cells during manipulation. This work generated new research directions in the following areas: 1) force sensors for biological structures; 2) mechanical property characterization of biological structures; 3) force feedback for biomanipulation improvement; and 4) virtual and augmented reality environments for biomanipulations. All of these research areas have been generating knowledge that is applicable for the automation of embryo manipulations. For example, Sun et al. developed a MEMS-based cellular force sensor which was used to characterized the mechanical properties of mouse zona pellucida [21]. Ammi and Ferreira built upon Arai’s and Sun’s research and developed an augmented reality micromanipulation system using virtual haptic rendering [20]. Finally, Kim et al. revisited Arai’s study and developed a teleoperated microinjection system fitted with a
PVDF piezoelectric polymer sensor [19]. Kim et al. used this system to study the impact of injection velocity on the required forces for zebrafish egg cell penetration.

1.4. ES Cells and the Process of Gene Transfer into the Mouse*

The genetic revolution in medical science has spurred an increased demand for methods to model human diseases in living organisms. Typically, the common house mouse (Mus musculus) is used for genetic research for several important reasons. First, the sequences of both the human and the mouse genomes have now been completed. Because of this, genes known to cause human affliction can readily be identified in the mouse. Second, the physiology of the mouse appears to closely resemble that of humans as evidenced by the similarities demonstrated by many of the human disease models that have been established in the mouse. Third, and perhaps most importantly, the mouse is currently the only animal in which specific genetic modifications can be achieved through gene targeting. Because of these factors, there has been a surge in the demand for genetically engineered mice (GEMs) by the medical community.

The production of GEMs via gene targeting progresses through three well-defined steps: 1) construction of a gene targeting vector, 2) establishment of gene-targeted mouse embryonic stem (ES) cells, and 3) creating chimeras with the targeted ES cells that ultimately pass (transmit) the engineered gene to their offspring. Construction of the gene-

* The text in this section was reproduced from the journal paper [67], which was submitted for review to the IEEE Transactions on Robotics - Special Issue on Bio-Robotics. This section was reviewed and edited by Dr. Randy Thresher, who is the director of the Animal Models Core Facility at UNC Chapel Hill and one of the paper’s authors.
targeting vector is relatively straightforward, using cut-and-paste molecular biology techniques to assemble a plasmid with the appropriate features. The plasmid is then introduced into the ES cells through electroporation and, in a low frequency event, the plasmid undergoes homologous recombination with the chromosome and replaces the resident gene with that of the engineered gene. At this point, the plasmid is permanently engineered into the mouse chromosome. To create chimeras from the genetically modified ES cells, they are micro-injected into early-stage mouse embryos.

The standard method of injecting ES cells into a mouse embryo is performed using a procedure called blastocyst microinjection. In this procedure, 3.5-day-old mouse embryos are collected from a donor female. At this stage, the embryos are in the form of fluid-filled spheres called blastocysts. Using an inverted microscope fitted with specialized optics, micromanipulators and micro-volume syringes, the operator visualizes the blastocysts and captures one with a holding pipette. The operator then uses a second (injection) pipette to collect some of the genetically engineered ES cells. The injection pipette is then used to puncture the blastocyst and the ES cells are then expelled into the blastocyst’s cavity (blastocoel). The injected blastocyst is released and the process repeated many dozens of times in a single session. All steps are performed while peering through the binoculars of the microscope. In addition to the tedium, this is a very precise technique that requires many months for an individual to master.
1.4.1. Details of the Blastocyst Microinjection Process

The procedure for injecting embryonic stem cells into blastocysts, as described by Hogan et al. in [57], starts with the preparation or selection of appropriate injection and holding pipettes. These pipettes are typically made of borosilicate glass capillary tubing, which can be worked in microforges to provide the required tip diameters.

The injection pipette is used to collect the ES cells, so its tip diameter is normally in the order of 15 µm. This tip size is selected to allow the entry of a single ES cell without distortion. Fabrication of an injection pipette, or needle, is itself a laborious process requiring extensive training. To fabricate the needles, the capillary tube must be heated and pulled to a specific internal diameter, cut and then beveled on a diamond-coated grinding wheel. Finally, a fine point must be pulled on the tip of the needle using a microforge. The very recent application of piezo-driven microinjection systems has allowed for blunt-ended injection needles to be used, which are much easier to fabricate. However, the piezo systems are not inexpensive and many microinjection facilities continue to rely on hand-manufactured sharpened needles for blastocyst microinjection.

A holding pipette is used to hold the blastocyst in a stable position for the microinjection. Typical tip diameters of this type of pipettes are in the order of 75 µm, however the tip can be smaller to prevent the suction of the collapsing blastocyst into the pipette after the microinjection.

After fabricating the pipettes, the microinjection procedure continues with the assembly of the micromanipulators and the alignment of the pipettes in the injection area. The setup for these steps include two micromanipulators with instrument holders, an
inverted microscope with 200x magnification, and two micrometer syringes for the holding and injection pipettes.

When assembling the system for manual microinjections, the micromanipulators are positioned on the left and on the right side of the microscope. The left micromanipulator usually holds the holding pipette, and the right micromanipulator holds the injection pipette. Flexible polyethylene tubing filled with light paraffin oil connects the pipettes to their respective micrometer syringes, and the injection pipette is back-filled with low-viscosity silicone oil. The described blastocyst microinjection setup is schematically shown in Fig. 1.3.

Continuing the procedure, a Petri dish containing injection medium overlaid with light paraffin oil is placed on the microscope stage. This is followed by the alignment of the pipettes, which is performed to position the pipette tips parallel to the bottom of the Petri dish. Selected blastocysts and ES cells are then transferred into the injection area, starting the injection process per se.

![Diagram of blastocyst injection setup](image)

**Fig. 1.3.** (a) Set-up used for blastocyst injections. (b) Orientation of instruments and embryo in the injection area.
After the blastocysts and ES cells settle at the bottom of the Petri dish, ten to fifteen cells are selected and drawn into the injection pipette. The blastocyst to be injected is then immobilized by applying a vacuum pressure at the holding pipette. It is then carefully rotated (with the help of the injection pipette) until the inner cell mass (ICM) is directly opposite to the injection point, as shown in Fig. 1.4(a). Following the blastocyst immobilization, the loaded injection pipette is introduced into the blastocoel cavity, and the ES cells are slowly expelled, as depicted in Fig. 1.4(b). After delivering the ES cells, the injection pipette is then withdrawn and the process repeated up to 100 times in a given microinjection session. Finally, the injected embryos are implanted into the uterus of a pseudo-pregnant mouse for development to term.

![Fig. 1.4. Blastocyst microinjection: (a) the holding pipette immobilizes the embryo; and (b) the injection pipette deposits ES cells in the blastocoel cavity.](image)

### 1.5. Contributions Towards Biomanipulation Automation

Even though much of the focus of the bio-robotics research community is currently on the development of augmented reality systems for biomanipulation, there exists an ever-
growing need for improved consistency and efficiency in this area. This need can only be satisfied by full process automation; hence this research continued to push towards a fully automated system by concentrating on the development of a novel system setup for cell microinjections.

Here, a knowledge-based control system was used as the basis of producing an automated blastocyst microinjection system. The idea was to learn the task from skilled users, who have implicit knowledge about the microinjection process that they cannot articulate. To do this, a framework was created to allow a computer to observe and register all human actions during semi-automated microinjection experiments.

The developed framework consisted of a semi-automated system that was teleoperated from a graphical user interface (GUI) running on a desktop computer. The GUI provided the user with real-time video from the microscope, and received microinjection control commands through a joystick. This created a work environment with the feel of a computer game, one that was more ergonomic and comfortable for the operator. Furthermore, this system constructed a knowledge database by logging the cause and effect data recorded from the teleoperated microinjection trials.

The developed semi-automated system is described in detail in this work. This system was tested by a microinjection expert and by two novices, and the experimental data gathered was used to derive a set of preliminary knowledge-based rules for blastocyst microinjections. These rules were then encoded in an attempt to fully automate the microinjection process.
The examination of experimental results obtained with the teleoperated system showed that both the expert and the amateurs obtained microinjection success rates greater than 80%, surpassing the typical 40-70% range reported on the literature [4]. Furthermore, the implantation of the successfully injected blastocysts into surrogate mothers showed that the blastocysts injected on the semi-automated system were more likely to develop to term than those injected by experts through the traditional manual method. Therefore, the semi-automated system developed for blastocyst microinjections was proved effective and easy to use. Furthermore, the experiments demonstrated that this new system eliminated the need for extensive training of microinjection operators.

Contributions towards full process automation are also presented in this work, including the creation of a new system configuration more appropriate for computer controlled operations; the identification of basic microinjection tasks for automation; and the design of low and high-level automation algorithms. These developments were tested through preliminary experimentation and they successfully demonstrated automation by accomplishing automated blastocysts microinjections. Unfortunately, full process automation was not achieved in this work due to problems related to the automated delivery of ES cells. However, automatic microinjections performed with the assistance of an operator obtained a 75% success rate, showing that the developed system is close to achieving fully automated microinjection. Consequently, the research presented in this work shows that full automation of blastocyst microinjection is possible and that it increases the efficiency rate of this operation.
2. THE SEMI-AUTOMATED MICROINJECTION SYSTEM

The first step towards a full automation of blastocyst microinjections was the development of a semi-automated system capable of performing this task under the control of a human operator. The system presented in this chapter is a teleoperated cell injection system in which the user controls the entire microinjection process through a joystick and a graphical user interface.

This setup not only provides the operator with a more ergonomic working environment, but it also allows the controlling computer to capture and register all human actions during the microinjection experiments. The resulting reactions in the work area are also observed and registered by the system, and are based on information extracted from real-time video data. Therefore, this system allows the construction of a knowledge database from which future intelligent controllers can be developed.

This chapter describes the semi-automated system configuration, the equipment used, and the software algorithms developed for system control. A description of the graphical user interface is also provided, highlighting its features and capabilities that make whole the system easy to operate and capable of reliable experimental data collection.

2.1. Main Components

The developed semi-automated microinjection system was largely based on standard equipment used for manual blastocyst microinjections. Operations were
performed under a Zeiss IM35 inverted microscope, which provided the necessary optical magnification and illumination level for properly imaging the injection area. Images of that area were captured by a black-and-white CCD camera (LCL-902K, Watec America Corp.) that was installed on the microscope. The camera provided video feedback to the system’s controlling computer.

The other major components of the developed system include: 1) two Leitz manual micromanipulators for positioning of the holding and injection pipettes; 2) one computer controlled PMM piezo injector; 3) one Siskiyou MX7600R motorized micromanipulator and its respective driver for high-resolution motion control of the injection needle; and 4) two Narishige IM-6 micrometer syringes outfitted with motors for computer control of fluid motions. Later, a motorized XY stage was included as part of the system. This stage was used to move the petri dish around the microscope’s field of view, see Fig. 2.1.

A desktop computer with a Pentium4® CPU running at 3.2GHz was employed for processing, control, and for the teleoperated biomanipulations. The computer used a WinTV-PCI frame grabber card for capturing video, and a Saitek Cyborg Evo Force joystick as the control input for all motion commands.
Fig. 2.1. The blastocyst microinjection system setup.
2.2. System Configuration

The microinjection system configuration is shown in block diagram form in Fig. 2.2. The figure shows how the user controls the microinjection operation from the computer station, where he/she interacts with the system through a monitor and a joystick. Five devices are independently controlled from the joystick: 1) the micromanipulation robot for accurate positioning of the injection pipette; 2) the piezo injector used for blastocyst penetration; 3) the XY stage used to move the Petri dish around the microscope’s field of view; 4) the micrometer syringe used to produce suction force at the holding pipette; and 5) the micrometer syringe used to control the dispensing of ES cells into a blastocyst.

From Fig. 2.2 it can be seen that two devices are used to control the position of the injection pipette: 1) the manual micromanipulator; and 2) the computer controlled robot. The manual micromanipulator provides coarse positioning, i.e., large displacements, while the micromanipulation robot provides fine motion control, i.e., fine displacements. This robot is controlled using serial commands and provides 3-dimensional motions with 0.1µm resolution in each direction.

The holding pipette in this system is positioned manually through the left Leitz micromanipulator. Its position is set when a new Petri dish is placed on the microscope stage, and is kept fixed throughout the teleoperation procedures. This design was adopted because blastocyst manipulation was accomplished using a single moving pipette, in this case the injection pipette. Once a blastocyst was dropped into the injection area, it was rotated and translated by the user using the injection pipette alone. Then, by turning the
holding pipette suction on, the blastocyst was immobilized for the duration of the injection cycle. ES cells were injected into the blastocyst after the injection pipette was inside the blastocoel cavity. The injections were facilitated by the use of the piezo injector.

2.3. Software Units

The system configuration diagram presented in Fig. 2.2 shows that the processing software is composed of three main units: 1) the video processing unit; 2) the motion control unit; and 3) the experimental data recording unit.

![Fig. 2.2. The microinjection system configuration.](image-url)
The video processing software unit is responsible for the acquisition, processing and displaying of video frames. Video frames are acquired from the frame grabber card, and post-processed to locate the pipettes and any blastocysts. This software unit also estimates the optimum injection point through the blastocyst’s trophoblast cell layer, and it displays the original and processed video frames on the computer monitor. A description of the image processing algorithm used within this module will be given in Chapter 4.

The motion control software unit receives and manages motion commands from the joystick. Based on the inputs, this unit does all the necessary processing and signal generation to drive the motion devices via the computer’s serial ports.

The data recording unit observes and records all events taking place in an experiment into a data file. This unit keeps track of all inputs and outputs of the micromanipulator system. It also records the positions of the injection and holding pipettes, the blastocyst position, and a continually updated estimate of the optimum blastocyst injection point. The goal of this data collection system is to create a database from which a knowledge-based controller will be developed to automate the blastocyst microinjection process. This knowledge-based controller is desired for three reasons: 1) it can be encoded to see how it performs as an automated controller, 2) it can be used as a standard controller, against which other machine-learned or intelligent controllers can be tested, and 3) it can be used for the derivation of performance metrics.
A small section of an experimental data file is shown in Fig. 2.3. The figure shows that the data recording unit creates text files in which experimental information is encoded with description characters followed by the corresponding numerical values and time stamps. Using this format, everything that happens during an experiment can be recorded for later use. The meaning of the description characters are given in Table 2.1 and Table 2.2.

```
... 
  h 118 127   20:08:19.64 
m 0 450   20:08:19.73 
m 1 450   20:08:19.89 
b 85 119   20:08:19.92 
i 213 126   20:08:19.92 
p1 64   20:08:20.01 
p2 1   20:08:20.01 
...
```

*Fig. 2.3. Fraction of an experimental data file.*

<table>
<thead>
<tr>
<th>Character</th>
<th>Numerical Values</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>b</td>
<td>[bx] [by]</td>
<td>Blastocyst coordinates</td>
</tr>
<tr>
<td>s</td>
<td>[s]</td>
<td>Selected injection sector</td>
</tr>
<tr>
<td>h</td>
<td>[hx] [hy]</td>
<td>Holding pipette coordinates</td>
</tr>
<tr>
<td>i</td>
<td>[ix] [iy]</td>
<td>Injection pipette coordinates</td>
</tr>
<tr>
<td>m</td>
<td>[axis] [amount]</td>
<td>Motion command</td>
</tr>
<tr>
<td>p1</td>
<td>[byte]</td>
<td>Command byte 1: micrometer syringes</td>
</tr>
<tr>
<td>p2</td>
<td>[byte]</td>
<td>Command byte 2: micrometer syringes and piezo injector</td>
</tr>
</tbody>
</table>
The control of the micromanipulation robot is performed through the motion control software unit, which has subroutines solely dedicated to the processing of joystick commands related to this task. This ensures that user commands are formatted and submitted to the robot driver in real-time, thus guaranteeing precise motion control.

During blastocyst micromanipulation, operators often use fast motions to quickly adjust the position and orientation of blastocysts for injection. In contrast, small and precise motions are used during the actual injection phase. Therefore, the micromanipulator robot used was selected based on its speed and accuracy.

The Siskiyou MX7600R motorized micromanipulator was the robot selected to the developed microinjection system. It is able to move at “very fast” speeds, reaching up to 1.7 mm/s of linear velocity while still maintaining its accuracy thanks to built-in encoders. The robot’s minimum controllable displacement is 0.1 μm, which is enough for the blastocyst micromanipulation tasks since the smallest cells involved in the process (the ES cells) typically measure 10 μm in diameter.
The fast displacements and the precise motions required for the micromanipulations are achieved by applying an exponential function to the analog commands generated by the joystick, as shown in Fig. 2.4. This way the operator can command small or large displacements without changing any of the system settings. Furthermore, the joystick’s slider bar is used as a velocity gain control to allow for even faster motions, so the operator can easily increase the velocities directly from the joystick.

\[ dist = \frac{\text{sign}(J)}{10} \exp \left( \frac{|J| - 500}{592} \right) \]

**Fig. 2.4.** Exponential function applied to the joystick commands. Modifying the linear values generated by the joystick allows for fast displacement and for small and precise motions without changing any of the system’s settings.
2.5. XY Stage

The developed semi-automated blastocyst microinjection system incorporates an XY stage to move the petri dish around during the procedures. This is useful because it allows the collection of ES cells to be conducted from the computer station, thus permitting the operator to remotely carry out the entire microinjection procedure. The operations performed at the microscope stage are restricted to the initial setup of the working wells and the alignment of the pipettes. After that all operations are performed from the computer station using the joystick and keyboard.

The ability to remotely load the injection pipette and carry out entire procedures saves time when multiple microinjections are performed by avoiding trips to the microscope stage. This would be necessary without the XY stage because the ES cells are located separately from the injection site, and because they are typically loaded into the injection pipette prior to every three or four blastocyst injections.

Another advantage of having an XY stage comes from the fact that several working wells containing blastocyst and ES cells may be put on the same petri dish, as shown in Fig. 2.5. Consequently, there is a need to move each of those sites to the microscope’s field of view.

The installed XY stage presents a step resolution of 10 μm in each direction, and can move at speeds of up to 45 mm/s. A custom controller board drives this stage, and also provides storage area for the coordinates of 20 locations, allowing for instantaneous motion between different sites. This feature speeds-up multiple microinjection tasks by enabling fast and accurate motion between the blastocyst and the ES cells sites.
Fig. 2.5. Typical Petri dish setup. Each working well contains both blastocysts and ES cells for injection, and several working wells may be put on the same petri dish.

2.6. The Interactive User Interface

Because the blastocyst microinjection system is operated from a remote computer, a graphical user interface (GUI) was developed to facilitate interaction between the operator and the blastocyst’s micro-world. The GUI allows for easy control of the microinjection operation, and also converts a tedious and tiring manual microinjection task into a task with the look and feel of a computer game. The interactive GUI for microinjection puts the operator in front of a computer monitor instead of on the microscope, which is a: 1) more comfortable operating environment, and 2) teaching system for training new personnel about cell microinjection.

The developed interface also allows the user to monitor and tune the motion devices and the video processing algorithm, facilitating adjustments for peak performance and for the collection of reliable experimental data. This is achieved with the display of
both the original video (from the microscope’s CCD camera) and the processed video frames simultaneously. This provides the user with the means to visually inspect the functioning of the video processing algorithm instantaneously. If processing yields good tracking results throughout the microinjection process, the user has the option to save this experimental data into the knowledge database. This feature allows for the collection of reliable experimental data, which is essential for the development of machine-learned controllers planned for a future phase of this research.

To give the teleoperated microinjection process the feel of a computer game, a joystick was selected as the input device for the majority of the motion commands. As previously mentioned, using the joystick the user can control: 1) 3-dimensional motions of the injection pipette through the micromanipulation robot; 2) the activation of the piezo injector device; 3) fluid motion on the holding and injection pipettes through the motorized micrometer syringes; and 4) the position of the Petri dish through the motorized XY stage. These functions were assigned to the joystick and its buttons as shown in Fig. 2.6.

The GUI can be also used to adjust motion parameters such as: 1) the joystick sensitivities; 2) the speed of each axis of the motorized micromanipulator; 3) the conversion rate between joystick inputs and the size of the motion commanded; 4) the speed of the motors driving the micrometer syringes; and 5) the speed of the XY stage axes.

Another main feature of the developed graphical interface is the control given over the video processing algorithm. Using the GUI the user can customize several parameters that are used for finding and tracking the motion of the blastocyst and the pipettes. These
parameters include the approximate blastocyst size; the resolution of the search for the best injection point on the trophoblast; and the templates for the injection and holding pipettes, which are used for tracking via a matching algorithm. The next chapter will describe these parameters in more details.

A screenshot of the GUI is shown in Fig. 2.7. This screenshot shows that the GUI has two video panels: 1) original video frames are displayed in the left panel, and 2) processed video frames are displayed in the right panel. Fig. 2.7 also displays how the tracking algorithm superimposes object templates onto the processed video frames, giving instantaneous visual assessment of the video processing algorithm. The estimated

Fig. 2.6. Joystick function assignments: 1) translate the injection pipette along the X-, Y-, Z- and T-axis, 2) translate the XY stage along its X- and Y-axis, and 3) control the piezo injector and micrometer syringes. Joystick picture reprinted with permission from [68].
coordinates for the blastocyst, injection pipette, and holding pipette are also displayed within the interface.

A detailed description of the BlastoWorks software is given in the Appendix 1. In that section the reader can find functional diagrams and instructions to use this interface.

2.7. Conclusions

This chapter presented the design and implementation of a novel semi-automated system for microinjecting embryonic stem cells into blastocysts. Here, the implementation of real-time control of microinjection devices from a joystick was realized. This was combined to an interactive graphical user interface, resulting in the creation of a setup where blastocyst microinjections were treated like a computer game.
This new semi-automated microinjection system was developed as the first step towards full system automation. It was created to allow the evaluation of the system mechanisms and to develop control algorithms. In addition, the semi-automated system was designed to enable the acquisition of human knowledge about the microinjection process. This was realized through automatic registration of the operator’s actions and of their consequences in the work area.

This chapter provided detailed descriptions of the semi-automated system configuration and of the equipment employed to build it. The next chapter presents system evaluation experiments, including a comparative study between the microinjection performances achieved using the new system and the traditional setup.
3. **EXPERIMENTS AND ISSUES RELATED TO SEMI-AUTOMATED BLASTOCYST MICROINJECTION**

This chapter describes the experiments conducted with the developed semi-automated blastocyst microinjection system. The evaluations of the robot control system are presented first, followed by the experiments performed to compare teleoperated microinjections with the traditional manual blastocyst microinjections. In the latter case, the main goal was to evaluate the differences between the two methods in terms of microinjection survival rates and birth rates. Teleoperated microinjection experiments were also carried out under the control of novice operators. This was done to demonstrate the major improvements that the teleoperated system provides over the traditional method, including the dramatic reduction in operator training time and an increased percentage of birth rates.

In addition to presenting the conducted experimentation, this chapter provides an analysis of the observed blastocyst microinjection failures. This analysis includes the identification of the causes and possible solutions for the most common failure types. The gathered knowledge is then summarized by a set of rules compiled to guide teleoperated microinjections. These rules consist of the sequence of actions that are most likely to result in successful blastocyst microinjections according to the performed experiments.
3.1. Evaluation of the robot control system

The control of the injection pipette motions provided by the system was tested by conducting a series of path-following experiments. These experiments consisted of controlling the motion of the injection pipette tip around a circular path with 130 µm in diameter using four different approaches:

i. The operator used the joystick to generate motion commands for the robot

ii. The operator used the manual micromanipulator to directly control the injection pipette motions

iii. The system performed the task autonomously using an open-loop control strategy

iv. The system performed the task autonomously using visual servoing

The best results obtained from a series of 10 trials with each of these experiments are presented in Fig. 3.1. The figure shows time-lapsed pictures where the red circles represent the target path, and the black dots represent the position of the injection pipette’s tip at each processed video frame. The figures also show the duration of one full turn around the circular path and the mean square errors (MSE) computed from the deviations from the desired path.

It can be seen from the results in Fig. 3.1 that the operator’s ability to control the injection pipette position using the joystick was similar to his ability to control the pipette position using the manual micromanipulator. In both cases overshoots along the path can be seen, but the actual deviations are less than 20 µm. On the other hand, the experiments showed that direct manual control was faster than control through the joystick, i.e., the user
was able to complete a full turn around the circular path in a shorter period of time when using the micromanipulator.

Even although the intention of experiments was not to perform the task as fast as possible, but rather as precisely as possible, the results of the trials in Fig. 3.1 showed that the operator took 23.8 seconds to complete a full turn around the circle when using the joystick, and 8.6 seconds when using the manual micromanipulator. The observation is that the operator is very well trained in manual control of pipette motions, but not familiar
with the use of the joystick as yet. Therefore, faster and more precise motion control is expected as the operator becomes more familiar with the teleoperated system.

The results in Fig. 3.1 also show that autonomous pipette control provides smoother motion control of the injection pipette compared to its manual counterpart. The implemented open-loop automatic control scheme proved this fact, although it resulted in large tracking errors as Fig. 3.1(c) shows. The large MSE obtained in this case is a consequence of the system not being calibrated, i.e., the robot axes were not aligned with the image axes. Nonetheless, a qualitative analysis of the results shows that the obtained path following performance was reasonably good. This is important because it shows that position information obtained from the robot can be used to improve the injection pipette’s visual tracking algorithm. For example, the developed vision system uses this position information to adjust the location of the injection pipette’s search window.

In addition, Fig. 3.1(d) shows that autonomous pipette control improved by adopting a visual servoing system. In this case the controller continuously adjusted the pipette’s position along the desired path using a PID control scheme, resulting in a path following trial with smooth control and very small MSE. These results further motivate the goal of fully automating of the blastocyst microinjection process since better motion control translates into reduced chances of inflicting lethal damage to the embryos.

3.2. Semi-Automated Microinjection Experiments

The developed microinjection system was installed at the Animal Models Core (AMC) Facility at UNC Chapel Hill, where experimentation was carried out with real stem
cells and blastocysts. The teleoperated microinjections were performed by a microinjection expert and by two amateurs in the area (the graduate student that put the system together, here identified as amateur-1, and a young man who had never seen blastocysts microinjection before these trials, here called amateur-2). After each microinjection, the expert evaluated the blastocysts to define if they had survived or were killed by the operation. Subsequently, the surviving blastocysts were either incubated for 24 hours to verify survival, or implanted into surrogate mothers for development to term. The results obtained are summarized in Table 3.1.

An examination of the data in Table 3.1 shows that both the expert and the amateurs obtained an overall blastocyst survival rate greater than 80% for the these preliminary microinjection trials. This shows that the performance of the semi-automated system has already surpassed the typical 40-70% success rate range found for manual microinjections. Furthermore, since the success rates obtained by the operators were so close, one can conclude that the system is easy to use and that extensive operator training is no longer be required.

Additionally, both the expert and the amateur-1 operators obtained 100% survival rate for the injected blastocysts that were cultured for 24 hours. These results corroborate the previous observations and show that the developed system allows for the safe handling of embryos.
Table 3.1. Results from teleoperated microinjections

<table>
<thead>
<tr>
<th>Operator</th>
<th>Expert</th>
<th>Amateur-1</th>
<th>Amateur-2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total number of injected blastocysts</td>
<td>213</td>
<td>65</td>
<td>74</td>
</tr>
<tr>
<td># blastocysts that survived microinjection</td>
<td>189 (88.7%)</td>
<td>55 (84.6%)</td>
<td>61 (82.4%)</td>
</tr>
<tr>
<td># blastocysts cultured for 24 hours</td>
<td>25</td>
<td>16</td>
<td>0</td>
</tr>
<tr>
<td># blastocysts that survived for 24 hours</td>
<td>25 (100%)</td>
<td>16 (100%)</td>
<td>—</td>
</tr>
<tr>
<td># blastocysts implanted</td>
<td>172</td>
<td>39</td>
<td>64</td>
</tr>
<tr>
<td># mice born</td>
<td>69 (40.1%)</td>
<td>13 (33.3%)</td>
<td>11 (17.2%)</td>
</tr>
<tr>
<td># chimeras</td>
<td>36 (20.9%)</td>
<td>2 (5.13%)</td>
<td>4 (6.25%)</td>
</tr>
</tbody>
</table>

The plot in Fig. 3.2 presents the success rates of teleoperated blastocyst microinjections as a function of the trial number for the expert trials. In this case, each trial represents a different microinjection day, so the plot demonstrates the progression of the expert operator performance as she became more familiar with the system and with the joystick controls. The results show a demonstrable improvement over time and confirm that the success rates of the microinjections were consistently better than those expected from the traditional manual microinjections.
Fig. 3.2. Microinjection success rates from expert trials. Here, each trial represents a different microinjection day, and the success rate of the trial represents the percentage of good microinjections over the total number of operations performed on that day. The numbers above the bars represent the total number of microinjections for the trials.

A similar plot with data from microinjections performed by the amateur operators is shown in Fig. 3.3. The results in that figure demonstrate that both amateurs also performed consistently better than the expected rates from the traditional manual microinjections.

As mentioned above, some of the successfully injected blastocysts were implanted into surrogate mothers to develop to term. This process was performed by experts following standard protocols. Implantation involved the surgical insertion of 15 to 20 blastocysts into the uterus of each pseudo-pregnant recipient mouse. The obtained birth data is presented in Table 3.1, along with the yield of chimeras from those births. At a first glance, the data shows that the blastocysts injected by the expert were more likely to
develop into mice and to incorporate genetic modifications than the blastocysts injected by the amateurs. However, this comparison is not totally reliable because the microinjections performed by the expert and the ones performed by the amateurs were conducted on different dates.

A better assessment of the results is obtained through Table 3.2, which shows the results of manual microinjections performed by experts on the same days as the teleoperated microinjections. For the cases shown, the teleoperated and the manual microinjections used ES cells from the same batch and blastocysts collected from the same female donors.
Table 3.2. Manual microinjection data from expert operators

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td># blastocysts implanted</td>
<td>328</td>
<td>95</td>
<td>154</td>
</tr>
<tr>
<td># mice born</td>
<td>68 (20.73%)</td>
<td>27 (28.42%)</td>
<td>20 (12.99%)</td>
</tr>
<tr>
<td># chimeras</td>
<td>46 (14.02%)</td>
<td>3 (3.16%)</td>
<td>9 (5.84%)</td>
</tr>
</tbody>
</table>

A comparison between the data in Tables 3.1 and 3.2 shows that the blastocysts injected using the teleoperated system were more likely to result in births than the ones injected by experts using the traditional manual operations – even for the amateur operators. Furthermore, the yield of chimeras was also higher for the teleoperated microinjections than for the manual microinjections. These results can be clearly verified from the plots in the Fig. 3.4 and Fig. 3.5. They demonstrate that the teleoperated system improves the microinjection process, and that it is effective, easy to use, and clearly eliminates the need for extensive (and expensive) operator training.
Fig. 3.4. Birth rate from implanted blastocysts: Comparison between teleoperated and tradition manual microinjections. All manual microinjections were performed by experts.

Fig. 3.5. Yield of Chimeras from Implanted Blastocysts: Comparison between teleoperated and tradition manual microinjections. All manual microinjections were performed by experts.

The results presented above are encouraging, but additional experimentation and biological research is still needed to confirm these observations. The problem is that many biological and physiological factors cannot be controlled and these affect both the birthing
and incorporation of genetic modifications into the developing embryo, causing variations in the data gathered. For example, Table 3.3 shows five years of data recorded from manual microinjections performed by experts. It shows that there is a large standard deviation for both the birth rate and yield of chimeras. This indicates that these results are difficult to predict due to the nature of the biological processes. Nevertheless, the fact that births and chimeras were obtained from teleoperated blastocyst microinjections provided encouragement for the development of an automated system. The automation efforts and results will be presented in Chapter 8 and Chapter 9.

<table>
<thead>
<tr>
<th></th>
<th>Mean</th>
<th>Standard deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Birth rate</td>
<td>0.1964</td>
<td>0.1612</td>
</tr>
<tr>
<td>Yield of chimeras</td>
<td>0.0754</td>
<td>0.0916</td>
</tr>
</tbody>
</table>

Statistics based on a total of 47,568 implanted blastocysts

3.3. Analysis of Failed Microinjections

The results from teleoperated blastocyst microinjections presented above showed that an overall 13.4% of the attempted microinjections resulted in failure. A review of the experiments revealed the existence of several factors that contributed to the observed failures. These included actions and situations that lead to the infliction of lethal damage to blastocysts, or a mere failure to deliver the ES cells inside the blastocoel cavity.
The most prominent type of microinjection failure observed was the inability to get the injection needle inside the blastocyst. Situations like this can occur due to different reasons, including:

a. **The blastocyst is too soft.** After being extracted from the mouse, the blastocyst starts to lose its rigidity if kept at room temperature. Consequently, microinjections attempted a few hours after collection sometimes have trouble penetrating blastocysts that are too compliant. An example of such situation is shown in Fig. 3.6(a). This problem was minimized by keeping the blastocysts in ice until their transfer into the Petri dish for injection. However, even taking this precaution a few problems with soft blastocysts were encountered, suggesting that the use of a cooling stage should be investigated.

b. **The injection pipette is not at the correct height for the operation.** This can prevent the penetration of the injection pipette because the blastocyst may be pushed away from the needle, as the example in Fig. 3.6(b) illustrates. Consequently, it is important to carefully check the height of the injection pipette before each microinjection. This is currently being done manually, but an automatic height-adjustment system similar to the one used by Sun and Nelson [7] may speedup and further improve the success rate of the blastocyst microinjections.
c. **The holding force is not strong enough to immobilize the blastocyst.** When this occurs, the blastocyst can rotate or be pushed away when the injection pipette tries to penetrate it, what often causes a microinjection failure.

When the injection pipette is forced towards the blastocyst but cannot penetrate it, the blastocyst often collapses back to the morula stage. This process typically occurs in a few seconds, so there is usually no time for a second injection attempt. As a result, these situations were counted as failed microinjections during the experiments.
Another cause of failure encountered during the microinjections was the stickiness of the ES cells. These cells are normally covered by proteins that make them very sticky, causing clogs in the injection pipette or difficulties during their release into the blastocyst. When clogs occur, we noticed that the operators tend to increase the pressure in the pipette to overcome it, but the outcome of this action is typically not good. Strong fluid jets are shot into the blastocyst when the pipette finally unclogs, frequently resulting in the injection of too many ES cells, or worst, in the injection of some of the oil used to control the fluid motions.

Even when they don’t cause clogs in the injection pipette, sticky ES cells can still jeopardize microinjections by attaching themselves to the tip of the pipette during delivery. An example of this situation is presented in Fig. 3.6(c). The example clearly demonstrates that just-delivered ES cells may be pulled out of the blastocyst if they get attached to the pipette.

Problems with sticky ES cells are currently avoided by the careful selection of “clean” cells for injection. However, research is still underway on the development of treatments and filtering techniques to eliminate this extra task. One promising protocol consists of filtering the cells using a 12-micron filter and adding 1% glycerol to their media.

In addition to the causes mentioned above, operator errors were also responsible for a few failed microinjections during the experiments. These errors arise primarily from the operator’s lack of familiarity with the joystick controls, and consisted of moving the injection pipette too far into the blastocysts. In those cases, the blastocysts were damaged.
because the injection pipette either hit the inner cell mass (ICM) or went completely through the embryo, as happened in the example in Fig. 3.6(d). Although errors like these were rare, we expect that their frequency will be further reduced as the operators become more familiar with the teleoperated microinjection system.

3.4. Rules for Blastocyst Microinjections

From the design of the system and from the observation of actions taken by the expert during the performed experiments, a sequence of actions was compiled to guide blastocyst microinjections. This sequence comprises the following steps:

1. Using the XY stage, bring a blastocyst to the injection area.

2. Capture the blastocyst by creating suction on the holding pipette. See Fig. 3.7(a).

3. Check and adjust the orientation of the trapped blastocyst. The ICM should preferably be on the holding pipette side, but definitively away from the injection point. See Fig. 3.7(b).

4. Firmly secure the blastocyst by increasing the holding force. A small part of the blastocyst’s membrane should be seen inside the holding pipette.

5. Verify the vertical position of the injection pipette by checking the focus of the image. The pipette and the blastocyst’s membrane next to it should be on the same focal level.
6. Bring the injection pipette in contact with the blastocyst such that it is applying a slight pressure to the membrane. This point should be directly across from the holding pipette to avoid blastocyst rotations. See Fig. 3.7(c).

7. Activate the piezo injector and slowly move the injection pipette towards the blastocyst until it gets inside the blastocoel cavity. See Fig. 3.7(d).

8. Once the pipette has penetrated the blastocyst, turn off the piezo injector and activate the injection pipette syringe to deposit 10 to 15 ES cells in the blastocoel cavity. See Fig. 3.7(e).

9. Activate piezo injector for a sort period of time to ensure the ES cells are completely released from the injection pipette.

10. Remove the injection pipette from the blastocyst using a smooth, strait line motion. See Fig. 3.7(f).

11. Release the pressure from the holding pipette to release the injected blastocyst.
Blastocyst Orientation Adjustments

One of the most important steps of the microinjection sequence presented in the previous section is the adjustment of the blastocyst orientation prior to injection (step number 3). This is not a trivial task, and is often the most time-consuming step of the microinjection process. Nevertheless, insight into this problem was obtained from the analysis of the preliminary experiments. Basically two techniques are commonly used by the operators to rotate the blastocyst using the semi-automated system. These techniques are presented below.

Fig. 3.7. Steps of a successful microinjection: (a) Capture a blastocyst; (b) Adjust its orientation; (c) Bring the injection pipette in contact with the blastocyst such that it is applying a small pressure to the membrane; (d) Activate the piezo injector and slowly move the injection pipette towards the blastocyst until it gets inside the blastocoel cavity; (e) Deliver 10 to 15 ES cells; (f) Remove injection pipette from the blastocyst and release it from the holding pipette.

3.5. Blastocyst Orientation Adjustments

One of the most important steps of the microinjection sequence presented in the previous section is the adjustment of the blastocyst orientation prior to injection (step number 3). This is not a trivial task, and is often the most time-consuming step of the microinjection process. Nevertheless, insight into this problem was obtained from the analysis of the preliminary experiments. Basically two techniques are commonly used by the operators to rotate the blastocyst using the semi-automated system. These techniques are presented below.
3.5.1. Blastocyst rotation method based on fluid flow

The first technique employed to adjust the blastocyst orientation consists of creating fluid flow around it. This effectively causes the blastocyst to rotate, enabling the positioning of ICM away from the injection site. This method is summarized by the following algorithm:

1. Set the suction force on the holding pipette to a small value. See Fig. 3.8(a).

2. Using the injection pipette, gently push the blastocyst away from the holder until the injection medium starts to flow into the holding pipette. A small fluid flow around the blastocyst will make it rotate slowly. See Figs. 3.8(b) and 3.8(c).

3. Increase the suction force once the blastocyst’s ICM is next to the holding pipette. This will immobilize the blastocyst at the correct orientation for injection. See Fig. 3.8(d).

The use of fluid flow to rotate the blastocysts was shown to be an effective method during the microinjection experiments. However, it is recognized that considerably more research is necessary to correctly characterize this process. This will most likely include modeling, simulations and extra instrumentation to obtain quantitative measures of the forces and flows involved in the rotation process.
3.5.2. Push-pull blastocyst rotation method

Another technique commonly used to adjust the orientation of the embryos prior to injection consists of releasing and capturing the blastocysts from the holding pipette, and repeating this operation as many times as necessary to get it in a good orientation. This method is essentially very simple; however it relies on random chances to get the blastocyst in a good orientation for injection.

The amount of time required for blastocyst orientation using this push-pull method is hard to predict due to the randomness of the process. Nevertheless, experimentation showed that this is also an effective method to accomplish the orientation task.
3.5.3. Alternative method for blastocyst rotation

An alternative method that can be used to adjust the orientation of the blastocysts may be based on electro-rotation [10]. This method involves the addition of electrodes around the injection area, what may require the construction of a system similar to the integrated cell processor developed by Park et al. [55]. Their system was claimed to safely rotate 2-cell embryos, but no statistical evidence was provided in the paper. Therefore, more research is also necessary in this area if one wants to use electro-rotation to orientate blastocysts.

3.6. Conclusions

This chapter presented the experiments conducted to evaluate the semi-automated microinjection system. Initially, the controllability of the micromanipulation robot was assessed through path-following experiments using the joystick and automatic control routines. The obtained results showed that teleoperated robot control is comparable to the traditional manual control in terms of the mean-squared-error (MSE) obtained from the path-following tasks. In addition, automatic visual servoing was shown to enable smoother and more precise motion control, resulting in the reduction of the measured path-following MSE by one order of magnitude.

The second phase of the system evaluations consisted of a series of blastocyst microinjection experiments, whereby the performance of a microinjection expert was tested against the performance of two amateurs. Overall, the results showed that all
operators obtained microinjection success rates greater than 80%, surpassing the typical 40-70% success range reported on the literature for this type of operations. Therefore, the semi-automated system was proven effective, easy to use and, more importantly, capable of eliminating the need for lengthy and expensive training of microinjection operators. Furthermore, the microinjection experiments indicated that the survival rates improved over time, so greater rates can be anticipated as the operators become more familiar with the joystick controls.

The performed experiments also included the successful implantation of injected blastocysts into surrogate mothers to create chimeras. The birth rates from those implants showed that the blastocysts injected on the new teleoperated system were more likely to develop to term than those injected by experts using the traditional manual method. For example, the birth rates obtained from blastocysts injected by experts doubled when the teleoperated system was used, i.e., the birth rate increased from 20.73% to 40.12%. In addition, the results showed that the yield of chimeras was also higher for the blastocysts injected on the teleoperated system. Moreover, these observations were shown to be true for teleoperated microinjections performed by both an expert and by amateurs. Thus, the semi-automated system proved to be a more efficient system for performing blastocyst microinjections.

This chapter also presented a comprehensive analysis of the performed teleoperated experiments. This analysis enabled the identification of the most common causes of microinjection failures, and the creation of rules to guide blastocyst microinjection procedures. Within the identified failure types, the most prominent one was the inability to
get the injection pipette inside the blastocoel cavity. Situations like this were found to occur for numerous reasons, the most frequent being inappropriate holding force, incorrect injection pipette alignment, and the rigidity of the blastocysts.

The set of rules created to guide microinjection procedures included the description of a sequence of actions that are likely to result in successful blastocyst microinjections. In addition, a method to adjust the blastocyst’s orientation based on fluid flow in the holding pipette was also provided.
4. VIDEO PROCESSING ALGORITHMS

The extraction of position information from the video images is a key element for the success of an automated blastocyst microinjection system. It is also essential for the construction of knowledge databases that link the cause and effect of micromanipulation actions performed by the users. Such systems can only be accomplished with information about the position of the objects involved in the task, so the vision algorithm is necessary to provide the required feedback. Consequently, major considerations during the vision algorithm design were accuracy, robustness, and processing speed. The latter is important because the algorithm needs to be fast enough to provide real-time data for an automated controller. A detailed description of the developed video processing algorithm is presented in the following sections.

4.1. Overview

The video processing algorithm developed for the blastocyst microinjection system is responsible for the detection and tracking of three objects within the video frames: 1) the blastocyst, 2) the injection pipette, and 3) the holding pipette. The algorithm detects and tracks the blastocyst using Hough transforms; searches for the holding pipette using cross-correlation template matching; and localizes the injection pipette using either sum-of-squared-differences template matching or cross-correlation template matching. The
implementations of these techniques are described in this chapter, accompanied by evaluation results based on simulate and real images.

An overview of the implemented algorithm is presented in Fig. 4.1. The initial phase of the algorithm consists of a series of basic steps which are common to the detection of all three features of interest. It starts with the acquisition of a frame image from the frame grabber card, followed by the conversion of this RGB image into a gray-scale equivalent. The gray-scale image is then processed to produce gradient information and is subsequently sent to a Canny edge detector algorithm [58] that generates an image with the edges found. After this initial processing, the algorithm splits in three parallel

![Fig. 4.1. Overview of the video processing algorithm. The highlighted blocks indicate outputs recorded during the experiments.](image-url)
paths that are concerned with the detection of the blastocyst and the pipettes. Finally, when the localization processes are completed, the coordinates of the objects are recorded and algorithm is restarted with the acquisition of a new video frame.

4.2. Blastocyst processing

The goals of the vision algorithm for blastocyst image processing include the localization and analysis of the embryo’s orientation. Descriptions of the algorithms designed to accomplish these goals are presented below.

4.2.1. Finding the blastocyst

A Hough transform [59] algorithm is used to search for a blastocyst within the video frames. This image processing technique uses a parametric description of the object of interest to map each point in the image space onto a curve in a parameter space, which is later searched until the most consistent location of the object is found. The Hough transform is implemented by treating each point in the image space as a constant and the parameters as variables. The idea is to have the points “vote” for the location of the feature that best suits them. This algorithm was selected to locate the blastocyst because it is robust, fast, and yields excellent results for this particular task.

Since the blastocysts are approximately circular, searching for a circle within the video frames can perform the task of finding one of them. Consequently, the implemented Hough transform uses the parametric equation for circles, which is given by
\[(x - a)^2 + (y - b)^2 = r^2\]. Here \(a\) and \(b\) are the coordinates of the center of the circle, and \(r\) is the radius of that circle.

From the parametric equation above it can be noticed that finding a circle requires finding the values of three parameters, i.e., \(a\), \(b\) and \(r\). This would normally require a 3D Hough transform. However, the computations are simplified assuming a constant diameter for the blastocysts. The typical value of this parameter is 100 µm, but deviations are accommodated by allowing the user to adjust this value through the developed GUI. This provides increased flexibility and accuracy to the algorithm.

An example of the results obtained from the application of the Hough transform in a typical microinjection image is shown in Fig. 4.2. The figure includes an image of the

![Fig. 4.2. Example of the application of Hough transform to locate the blastocyst: (a) Original image; (b) Processed image; (c) Hough accumulator matrix.](image)
Hough accumulator matrix, which is used to record the votes for the blastocyst locations.

After applying the Hough transform and obtaining estimates for possible blastocyst locations via the accumulator matrix, the developed software evaluates those locations to determine if a blastocyst is actually present there. This is necessary because the roundness of the holding pipette tends to be attractive to the search algorithm, and because random locations may be selected by the algorithm if no circular object is present in the image. Therefore, an error detection algorithm was implemented for this task.

The decision on whether a blastocyst is present on the location suggested by the Hough transform is based on two tests. First, the peaks in the accumulator matrix are ranked and classified as true peaks or false peaks. Second, the blastocyst locations pinpointed by test 1 are evaluated in terms of the ratio between the number of edge pixels and the area of the blastocyst.

The classification of the Hough accumulator peaks is performed according to the pseudo-code in Fig. 4.3. The decision on the presence of a blastocyst at a true peak follows the algorithm in Fig. 4.4.

- Compute the mean \((m)\) and the standard deviation \((std)\) of the values in the accumulator matrix
- Classify the accumulator peaks based on their value \((Pv)\)
  - if \(Pv > m + 4*std\), peak = true peak
  - else, peak = false peak
- Rank the true peaks according to \(Pv\)

**Fig. 4.3.** Pseudo-code for the classification of accumulator peaks.
4.2.2. Analyzing the blastocyst orientation

After a blastocyst is positively identified by the algorithms described above, it is analyzed to determine if its orientation is appropriate for injection. For safety reasons the injection point should be located away from the blastocyst's inner cell mass and preferably at the thinnest part of the trophoblast cell layer.

Two different algorithms were developed for this purpose. The first one locates the thinnest part of the blastocyst’s trophoblast and selects this area as the best injection point. The second algorithm divides the blastocyst’s image in ICM and cavity sectors, and determines if the blastocyst orientation is acceptable for injection. This decision on the blastocyst orientation is based on the classification of the injection sector (the one directly opposed to the holding pipette). These algorithms are described next.
4.2.2.1. Finding the best injection area on the blastocyst

This automated blastocyst analysis locates the thinnest part of the trophoblast cell layer and determines the best area for injection. The algorithm starts by dividing the blastocyst image into sectors, and continues by classifying each pixel inside the sectors as either belonging to the blastocyst or to the background. This pixel segmentation is based on a global threshold of the video frame image. After the segmentation, a count is made of the number of blastocyst pixels within each sector. The sector with the largest number of blastocyst pixels is assumed to contain the ICM, and the sector with the least number of blastocyst pixels is initially considered as the best sector for injection. This selection is confirmed only if the injection sector is not adjacent to the ICM sector. Otherwise, that sector is removed from the search pool and another candidate sector is considered as a possible injection site. The pseudo-code for this analysis algorithm is presented in Fig. 4.5 and an example of the results obtained when analyzing real blastocyst images is presented in Fig. 4.6.

- Find the blastocyst
- Threshold the image
- Classify pixels as blastocyst or background
- Divide blastocyst image in sectors
- Count the number of blastocyst pixels within each sector
- Select the sector with the largest number of pixels as the ICM sector
- Select the sector that has the smallest number of blastocyst pixels and is not adjacent to the ICM sector as the best injection sector

Fig. 4.5. Pseudo-code for finding the best injection area on the blastocyst.
The developed GUI allows the user to customize this blastocyst analysis by allowing changes to the number of sectors used to find the optimum injection area. Sector count can vary from 2 to 20. For the blastocyst microinjection system, experimentation determined that 8 sectors was an optimal number for the task. With 8 sectors the dimensions of the arc of each sector approximates the outer diameter of typical injection pipette tips.

4.2.2.2. Qualifying the blastocyst orientation

As mentioned above, this second algorithm was developed to determine if the blastocyst orientation is acceptable for injection. Similarly to the first algorithm, this one also starts by dividing the blastocyst image into sectors and by classifying each pixel as belonging to the blastocyst or to the background. However, in this case the number of blastocyst pixel in each sector is used to classify the sector as ICM or cavity. Then, if the sector directly opposed to the holding pipette is a cavity sector, the blastocyst orientation is specified as good for injection.
This algorithm was developed from the observation of teleoperated blastocyst microinjections performed by an expert. The experiments showed that injecting on the thinnest part of the trophoblast cell layer is not crucial for success. Instead, it was noted that the only major consideration was to never inject through the blastocyst’s inner cell mass. Therefore, this algorithm only classifies the blastocyst’s orientation as acceptable for injection if the injection sector is not part of the ICM.

The pseudo-code for this algorithm is provided in Fig. 4.7 and examples of real blastocyst analyses are presented in Fig. 4.8. Note that when the injection sector (sector 0) is classified as \textit{ICM}, the closest acceptable sector for injection is marked in yellow on the processed video frame.

\begin{itemize}
  \item Find the blastocyst
  \item Threshold the image
  \item Classify pixels as blastocyst or background
  \item Divide blastocyst image in sectors
  \item Count the number of blastocyst pixels within each sector ($iBP$)
  \item Compute the mean ($m$) and the standard deviation ($std$) of $iBP$
  \item Classify the sectors based on their $iBP$ values:
    \begin{itemize}
      \item if $iBP > m + 0.5std$, sector = \textit{ICM}
      \item else, sector = \textit{cavity}
    \end{itemize}
  \item Classify blastocyst orientation based on the injection sector:
    \begin{itemize}
      \item if sector(0) = \textit{cavity}, orientation is good for injection
      \item else, orientation is not good (do not inject)
    \end{itemize}
\end{itemize}

\textbf{Fig. 4.7.} Pseudo-code for qualifying the blastocyst orientation.
Fig. 4.8. Original and processed images of blastocysts. The algorithm locates the embryo and classifies its sectors as ICM or Cavity. If the injection sector (sector 0) is a Cavity sector, the blastocyst orientation is good for injection. The ICM sectors are marked in red and an acceptable injection sector is marked in green. If sector 0 is not good for injection, the closest acceptable sector is marked in yellow.

4.3. Finding and tracking the pipettes

After finding the blastocyst, the next major tasks of the vision processing algorithm are to find and track the motion of the injection and holding pipettes. Cross-correlation template matching was the technique selected to search for the holding pipette, and sum-
of-squared-differences template matching was chosen to search for the injection pipette. These techniques, and their benefits, are presented next. Subsequently, detailed explanations of speedup techniques for template matching are presented, including theoretical formulas and experimental validations. Lastly, the results from robustness evaluations based on simulated and real video images are discussed.

4.3.1. Cross-Correlation Template Matching Overview

This type of template matching is used to search for objects of interest based on their outline. It was selected to track the holding pipette because it is simple, flexible, and robust for this application. Furthermore, the holding pipette has several features that create unique edges in the images, making it an ideal target to be tracked by cross-correlation. The reader is referred to [60] for a good introduction to matching theory.

The implementation of a cross-correlation template matching can be described as follows: Given an image \( I \equiv \{I(x,y); 0 \leq x < X, 0 \leq y < Y\} \) and a template image \( T \equiv \{T(m,n); 0 \leq m < M, 0 \leq n < N\} \) containing the object of interest, the best matching position of the template over the image is given by the peak of the function \( CC(x,y) = \sum_{m,n} T(m,n)I(m+x,n+y) \) computed over all pixels in the image \( I \).

4.3.2. SSD Template Matching Overview

Sum-of-squared-differences (SSD) template matching [61] uses the entire image of the object of interest to search for it. This technique was selected to track the injection
pipette because it is, as the cross-correlation method, simple, flexible, and robust for this application. The SSD method is preferred for tracking the injection pipette because this pipette is small and does not present many edge features. Consequently, the use of all template pixels instead of just the edge pixels improves the robustness of the matching algorithm in this case. This is demonstrated by the evaluation experiments presented later in this chapter.

The implementation of the SSD template matching can be described as follows: Considering the same image $I$ and template $T$ defined above, the best matching position of the template over the image is given by the minimum of the function

$$SSD(x, y) = \sum_{m,n} \left[ I(x + m, y + n) - T(m, n) \right]^2$$

computed over all pixels in the image $I$.

### 4.3.3. Why Template Matching?

The reader may have noticed that these template matching techniques are computationally expensive. The cross-correlation computations, for instance, require the convolution of two images. Furthermore, template matching has the potential to produce unreliable results if: 1) the object’s shape changes; 2) the object’s size or orientation is unknown; 3) unforeseen shadows appear due to changes in lighting; or 4) the object is partially occluded.

Despite these potential problems, template matching is well suited for the task of finding the pipettes in the microscope images. During micromanipulation, the motions of the tip of the pipettes consist of translations only. Therefore, the template matching algorithms in this application do not need be concerned with instances of orientation.
Furthermore, the sizes of the pipettes on the images do not change during the operations, so there is no need to perform affine transformations on the templates.

The imaging system used for the biomanipulations also provides a controlled illumination of the microinjection area, which allows for further simplification of the template matching algorithms. Under the microscope the illumination is highly uniform and is kept constant throughout the microinjection process. This means that the matching algorithms do not need to accommodate for unexpected shadows or uneven illumination, meaning no image normalizations are needed.

4.3.4. Template Acquisition

A major action for ensuring the success of a template matching algorithm is the selection of a good template image. A template that truly represents the object being searched for simplifies the necessary computations, and also improves the success rate of the algorithm. With this in mind, the developed user interface allows for the acquisition of templates directly from the live video captured by the microscope’s CCD camera. This functionality allows for template acquisition at the beginning of each microinjection experiment, making the program more flexible to changes in pipette shape. An example of template acquisition is shown in Fig. 4.9, which also presents the TemplateViewer tool used to visualize acquired templates.
4.4. Conclusions

This chapter described the design and implementation of real-time image processing algorithms for an automated blastocyst microinjection system. The main goals of the developed vision system were to locate and track both the blastocyst and the micromanipulation pipettes within video frames captured from a microscope. Additionally, the system was required to analyze the blastocyst for the selection of an optimum injection point on its trophoblast cell layer. These goals were met using Hough transforms to find the blastocyst, cross-correlation template matching to search for the holding pipette, and sum-of-squared-differences (SSD) template matching to locate the
injection pipette. The injection sector selection was accomplished by a simple and effective algorithm based on image segmentations.

The evaluations of the implemented vision techniques were emphasized during this research and will be presented in the following chapters. This was important because the design of automatic controllers for blastocyst microinjection depended heavily on the developed vision system. This dependency was due to two main reasons: First, this vision system was used to gather human knowledge about the microinjection process by automatically extracting experimental data from manual microinjections; second, this vision system was the major source of feedback data for the automatic micromanipulation controllers. The analysis of the data collected during the teleoperated trials was described in Chapter 3, and the developed knowledge-based controllers for microinjection automation are described in Chapter 8.
5. Evaluation of the Blastocyst Processing Algorithms

The first series of experiments associated with semi-automated microinjection concentrated on finding and analyzing blastocysts. If this section of the system’s image processing software did not perform adequately, i.e., if the software gave performance metrics with a low rate of success, then it was reasoned that the performance of the complete system would be compromised.

This part of the video processing algorithm was evaluated based on synthetic images and pre-recorded video from manual blastocysts microinjections. For the initial tests, video frames containing a simulated blastocyst at random positions and orientations were created and analyzed by the video processing algorithm. The results obtained were then compared to the true blastocyst positions and orientations, providing error measures for each processed frame. The performance of the algorithm was also tested for cases where increasing levels of noise were introduced into the experiments. This was done to test both the accuracy and robustness of the algorithms.

After obtaining performance metrics from synthetic data, the algorithm was evaluated on real images. In this case, two sets of video frames were manually assembled for the experiments: One with 300 images of real oocytes, and the other with 122 real blastocyst images. For each image the position of the oocyte or blastocyst was manually determined. These sets were then used as the baseline to evaluate the algorithm in the same way it was done for the simulated video frames. The influence of noise was also examined for the real images case.
5.1. Experiments with Synthetic Images

The simulated blastocyst images used for the evaluation experiments were generated using two circles as shown in Fig. 5.1. Given the desired blastocyst radius, $R$, the radius of the internal circle is defined as $0.8R$, and its displacement from the center of the larger circle is defined as $d$. The magnitude of this displacement was also defined to be a fraction of the blastocyst radius, such that:

$$|d| = \frac{1}{2} (R - r) = \frac{1}{10} R$$  \hspace{1cm} \text{Eq. 5.1}

The orientation of the simulated blastocyst is defined by the angle $\theta$, which dictates the direction of the displacement $d$. In this case, the best injection point on the simulated blastocyst’s trophoblast is always known, and is given by the angle $\theta$.

During the creation of the simulated video frames, the position and orientation of the blastocysts within the images are randomly selected and then used to generate the images. These values constitute the ground truths that are later used to evaluate the video

Fig. 5.1. Blastocyst simulation method. The larger circle represents the blastocyst’s zona pellucida, and the inner circle simulates the blastocoel cavity.
processing algorithm.

Noise was also added to the simulated video frames to enable better characterization of the developed algorithm. This was accomplished by four different methods: 1) reducing the number of pixels used to represent the blastocyst; 2) blurring the simulated images; 3) adding Gaussian noise to the pixel intensities; and 4) adding impulse noise to the images. Examples of noisy synthetic images are presented in Fig. 5.2.

The algorithm characterization also included evaluation of the influence of the blastocyst size on the localization errors, so two sets of experiments were conducted to test this factor: one with the blastocysts’ radius set to 21 pixels, and the other with the radius
set to 35 pixels. The number of sectors used to find the best injection point was set to 8 on all cases.

The performance of the algorithm was evaluated in terms of the mean localization error and the mean injection sector selection error. For each noise condition, 1000 simulated video frames were analyzed. During these experiments the blastocyst was assumed to always be present at the location pinpointed by the highest peak in the Hough accumulator matrix, i.e., the decision algorithm used to verify the presence of the blastocyst was turned off. Also, the best injection sector selection algorithm was the only blastocyst analysis algorithm evaluated with synthetic images.

### 5.1.1. Impact of Gaussian Noise

The performed experiments showed that the introduction of additive Gaussian noise to the simulated images had negligible impact on the performance of the blastocyst-finding algorithm. The resulting mean localization errors, which are plotted in Fig. 5.3(a), presented values around two pixels for all levels of noise tested. In the case of the injection sector selection, Fig. 5.3(c) shows that the mean errors were practically zero across the board. The distributions of errors from a sample experiment using a simulated blastocyst with 35 pixels radius and additive Gaussian noise with variance 20 are shown in Fig. 5.3(b) and (d). Those figures show that, for the conditions tested, the measured localization error was less than 5.4 pixels 90% of the time, and the sector selection error was zero 72% of the time. Moreover, Fig. 5.3(d) shows that if an error of ±1 sector is allowed, the sector selection algorithm provided a success rate of 94%.
5.1.2. Impact of Impulse Noise

Similar experiments were conducted to measure the impact of impulse noise on the performance of the developed blastocyst localization algorithm. This type of noise can be related to real image distortions that result from dust on the optical system lenses, or even
from bad pixel readings on the camera’s CCD sensor. The obtained evaluation results showed that the localization is robust on images corrupted by impulse noise with probability up to 2%. For higher noise levels the localization performance decays rapidly, mainly due to the large increase in the number of false edges found by the edge detection algorithm. On the other hand, the sector selection performance was not significantly affected by the tested levels of impulse noise. The measured mean sector errors were virtually zero for all cases tested.

The results described above can be verified from the plots in Fig. 5.4, which also includes sample CDF plots of blastocyst localization errors and sector selection errors. From Fig. 5.4(b) it can be seen that when the impulse noise probability was set to 2% and the blastocyst had a 35 pixels radius, 90% of the images tested resulted in a localization error smaller than 4 pixels. For the same conditions, Fig. 5.4(d) shows that the sector selection error was zero for 80% of the analyzed images, and less than ±1 sector for over 98% of those images.

5.1.3. Impact of Reducing the Number of Object Pixels

A third set of experiments was performed to evaluate the impact of “erasing” pixels that represent the blastocyst on the simulated images. The purpose of these experiments was to create situations that could be related to cases of edge detection failure or even to cases where partial occlusion occurs. Therefore, the set of simulated images for this experiment were generated from a randomly selected percentage of the blastocyst pixels.
The obtained results showed that the implemented algorithm is robust against reductions in the number of pixels representing the blastocyst. It can be seen from Fig. 5.5(a) that as the percentage of pixels decreases, the localization error increases. However, even when only 10% of the blastocyst pixels were used, the mean localization error was
smaller than 16% of the larger blastocyst radius, and less than 20% of the smaller blastocyst radius.

It was also noted from the experiments that reductions in the number of pixels representing the blastocyst did not affect the mean sector selection errors significantly. This can be seen from Fig. 5.5(c), which shows that the measured mean error was less than

![Graphs showing the impact of reducing the percentage of blastocyst pixels transferred to the test images on the blastocyst-finding algorithm.](image)

**Fig. 5.5.** Impact of reducing the percentage of blastocyst pixels transferred to the test images on the blastocyst-finding algorithm. (a) Mean localization error for simulated blastocysts with radius set to 21 and 35 pixels. (b) Sample CDF plot of the localization error obtained when using 50% of the blastocyst pixels and setting the radius to 35 pixels. (c) Mean injection sector selection error for the two blastocyst sizes. (d) Sample CDF plot of the sector selection error for the same conditions as in (b).
one sector for all cases tested.

The sample CDF plots in Fig. 5.5 show that when only 50% of the blastocysts pixels were used to create the test images, the localization error was smaller than 6.7 pixels 90% of the time, and the sector selection error was less or equal to 1 sector 93% of the time. From the same data it can be seen that the sector selection error was zero for 73% of the analyzed images.

5.1.4. Impact of Gaussian Blur

The last set of experiments with simulated blastocyst images studied the impact of Gaussian blur on the performance of the localization algorithm. This type of image distortion emulates the effect of changing the focus on real images, so these experiments basically evaluated how focusing affects the blastocyst localizations.

The obtained results, which are presented in Fig. 5.6(a) and 7.6(c), show that the algorithm is robust against blurring. It can be seen from the figures that the mean localization errors were less than 3 pixels for all of the blurring levels tested. Furthermore, the average sector selection was always correct.

Sample CDF plots exemplifying the performance of the algorithm for the case of a Gaussian blur with variance 2 are shown in Fig. 5.6(b) and 7.6(d). Those plots show that the localization error was less than 2.5 pixels 90% of the time, and that the sector selection error was zero 83.4% of the time. For this case, if an error of ±1 sector is considered acceptable, the sector selection was correct 99.7% of the time. Consequently, we can
expect the implemented algorithm to be robust against small changes in the focus of the imaging system.

Table 5.1 summarizes of the performed experiments with sample results that quantify the influence of each type of noise on the algorithm performance. Three levels of noise (low, median, and high) are represented in the table for each type of noise tested.

Fig. 5.6. Impact of Gaussian blur on the blastocyst-finding algorithm. (a) Mean localization error for simulated blastocysts with radius set to 21 and 35 pixels. (b) Sample CDF plot of the localization error from the case with Gaussian blur variance set to 2 and the radius set to 35 pixels. (c) Mean injection sector selection error for the two blastocyst sizes. (d) Sample CDF plot of the sector selection error for a Gaussian noise variance set to 2 and radius set to 35 pixels.
The percentage of correct blastocyst localizations and the percentage of correct injection sector selections are also shown in Table 5.1. Note that the sector selection errors shown in Table 5.1 were computed only for cases with correct blastocyst localizations because otherwise the measurements would have been meaningless.
5.2. Experiments with Real Images

The evaluation of the blastocyst-finding algorithm using real images was based on two sets of images: One with 122 blastocyst images and another with 300 oocyte images. Oocytes are unfertilized eggs which present the same size as the blastocysts. They were used in these experiments because they are easier (and cheaper) to obtain than blastocysts. The blastocyst images used here were obtained from pre-recorded microinjection videos that were generously provided by the Animal Models Core Facility at the UNC-Chapel Hill.

Each image in the assembled test sets was manually examined to determine the coordinates of the blastocyst or oocyte. This process created the baselines to evaluate the vision algorithm in the same way it was done for the simulated video frames. The experiments performed with these real images also included the analysis of the impact of noise on the algorithm’s localization errors.

5.2.1. Performance from Undistorted Images

As mentioned above, the algorithm performance was manually evaluated during the assemblage of the baseline sets. A total of 300 oocyte images were analyzed and it was determined that the algorithm did provide the correct localization for 99.7% of those images. The experiments with the 122 blastocyst images showed a 90% success rate for the localizations and a 94% success rate for the injection sector selection on those correctly localized blastocysts.
Following the baseline assemblage, the robustness of the developed algorithm was tested. The obtained results are described below.

5.2.2. Impact of Noise on the Blastocyst Localizations

As in the case of simulated video frames, the performance of the blastocyst localization algorithm was methodically tested by the addition of increasing levels of noise to the images prior to processing. Here, the localization errors introduced by Gaussian noise, impulse noise, and Gaussian blur were tested using the oocytes image set. The obtained results are summarized in Fig. 5.7.

The evaluation results demonstrated that the distortion of the real blastocyst images by Gaussian noise or by blurring did not significantly impact the algorithm performance. On the other hand, the introduction of impulse noise showed that it can cause major performance degradation. This is similar to what was observed during the evaluations based on simulated images, although the algorithm proved to be more robust on the real images. This can be noticed by comparing the Figures 5.4(a) and 5.7(b), which show that the localization algorithm is robust for impulse noise levels less than 2% for the simulated images, and less than 6% for the real images.

Similarly to the table presented in the previous section, Table 5.2 provides sample experimental results to summarize the impact of noise on the performance of the blastocyst-finding algorithm. Again, for each noise type tested, three noise levels are represented in the table. The results shown in Table 5.2 do not include measurements of injection sector selection error because these experiments were performed using images of
oocytes, which do not present a clear optimum injection point. The evaluation of the injection sector selection was performed using the real blastocysts images, and the obtained results are described in the next section.

Fig. 5.7. Impact of noise on the mean localization error of the blastocyst-finding algorithm applied to real oocyte images. The oocyte used for the experiments had a diameter of approximately 76 pixels, so the radius parameter was set to 38 pixels in all cases. (a) Impact of additive Gaussian noise. (b) Impact of additive impulse noise. (c) Impact of image blurring.
Impact of Noise on the Injection Sector Selection

This part of the vision algorithm was not methodically tested on real images because the optimum injection point selection is a subjective task (different sectors can be equally good for injection). Consequently, it is difficult to implement an automatic evaluation algorithm. Nevertheless, the manual evaluation of blastocyst localizations and injection sector selections once again showed a high success rate for the operations.

Table 5.3 summarizes the results obtained from tests with 122 real blastocyst images. It can be seen from the table that the worst measured performance for the injection sector selection presented a success rate near 87%, showing that the algorithm works well even if the images are distorted by Gaussian or impulse noise.
Examples of the processing of real blastocyst images are shown in Fig. 5.8. In the figure, a collection of blastocyst images is shown on the top row, and the corresponding processed images are shown on the bottom row. The circle surrounding the blastocyst indicates its estimated location, and the red mark indicates the section of the trophoblast chosen for injection.

Table 5.3. Experimental results from the analysis of 122 real video images of blastocysts.

<table>
<thead>
<tr>
<th>Exp. #</th>
<th>Gaussian Noise Variance</th>
<th>Impulse Noise Probability</th>
<th>Blastocyst Localization (% correct)</th>
<th>Sector Selection (% correct)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>90.16</td>
<td>94.55</td>
</tr>
<tr>
<td>2</td>
<td>50</td>
<td>0</td>
<td>82.14</td>
<td>86.96</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>5</td>
<td>70.0</td>
<td>91.43</td>
</tr>
<tr>
<td>4</td>
<td>50</td>
<td>5</td>
<td>60.26</td>
<td>93.62</td>
</tr>
</tbody>
</table>

Fig. 5.8. Blastocyst images and the processed images showing the results of the localization and injection sector selection algorithms. These images were cropped from the 320x240 video frames, and all blastocysts had an approximate diameter of 100µm.
5.2.4. Evaluation of the Blastocyst Orientation Check

The algorithm developed to classify the blastocyst sectors and decide if its orientation is acceptable for injection was manually evaluated using a set of 605 real images. The analysis of all images in this set resulted in correct blastocyst localizations, so the results presented here solely represent the performance of the decision algorithm.

During the experiments, each processed image was visually analyzed for the classification of the results in one of the following categories: 1) Success; 2) False positive; and 3) False negative. Success was used for the correct outcomes. False positive identified failures in which the algorithm wrongfully classified the blastocyst’s orientation as acceptable. Finally, false negatives identified failures in which the algorithm wrongfully classified the orientations as bad for injection. A summary of the obtained results is presented in Table 5.4.

Based on the evaluation results, we can say that the injection sector classification algorithm provided reliable decision on whether the blastocyst orientation is appropriate for injection. Table 5.4 shows that the probability of injecting a blastocyst through the

<table>
<thead>
<tr>
<th>Outcome classification</th>
<th>Number of occurrences</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Success</td>
<td>572</td>
<td>94.5 %</td>
</tr>
<tr>
<td>False positive</td>
<td>3</td>
<td>0.5 %</td>
</tr>
<tr>
<td>False negative</td>
<td>30</td>
<td>5 %</td>
</tr>
</tbody>
</table>
ICM was only 0.5% when this algorithm was used. Furthermore, the probability of a less harmful error was only 5%, i.e., the algorithm decided not inject the blastocyst on an appropriate spot only 5% of the time.

5.3. Processing Speed

The measured processing time for the localization and analysis of a blastocyst on a 320x240 pixels image was only 6ms/frame on the employed computer (Pentium4®, 3.2GHz). This measurement did not include pre-processing tasks such as image acquisition, gradient computations, and edges detection; however it shows that the employed techniques are well-suited for real-time video processing.

5.4. Conclusions

This chapter described the evaluation of the real-time image processing algorithms implemented for blastocyst localization and analysis. These evaluations were based on simulated and real images, and assessed both the performance and the robustness of the algorithms.

In the case of the blastocysts localizations, experimental results showed that the Hough transform algorithm is robust against Gaussian noise, partial occlusions and blurring. On the other hand, the algorithm performance was shown to be affected by the
presence of high levels of impulse noise, which caused the introduction of many false edges on the images. From the experiments with real images, a success rate of 99.7% was obtained for the localization of oocytes within video frames captured directly from the developed microinjection system. Tests with blastocyst images captured from a different microscope/camera setup showed 90% success rate for the localizations and a 94% success rate for the injection sector selections. In addition, these experiments demonstrated that the injection sector selection algorithm is robust against all of the noise types mentioned above, i.e., Gaussian noise, impulse noise, blurring and partial occlusions.

The experiments with simulated blastocysts also showed that the use of larger blastocyst images resulted in smaller relative localization errors, i.e., the ratio between the error and the blastocyst radius was smaller for big blastocysts than for the small ones. Thus, for best blastocyst-tracking performance, the microscope magnification should be kept as large as possible during the microinjection procedures.

Based on the evaluation results presented in this chapter and on the fact that the microinjection images captured under normal conditions present very low noise levels, the implemented vision processing system for blastocyst localization and analysis was deemed able to successfully accomplish its objectives. Furthermore, since the imaging conditions under the microscope present uniform illumination, good focus and a clean environment, the best success rates reported here represent the real system performance.
6. TEMPLATE MATCHING SPEEDUP

Based on experimental evaluation results, the basic implementation of the cross-correlation template matching algorithm described in Chapter 4 yielded good results for the localization of the injection pipette. Measured localization errors presented a mean of 0.377 pixels and a standard deviation of 1.266 pixels. However, the average processing time per video frame was unacceptably long (5.255 seconds per frame). Consequently, algorithm changes were made to address this issue, resulting in the development and implementation of four speedup techniques. These techniques are described in this chapter.

Here, reduction in the processing time required for template matching on each video frame is achieved by the application of simple search heuristics. The first heuristic comes from the fact that the positions of the pipettes in consecutive frames are highly correlated, thus suggesting the use of search windows centered at the previously known pipette location. Moreover, the same observation suggests that the search for the object should start exactly at its previous location and include its immediate neighborhood if the object is not readily found. This second heuristic rule is called expanding search area in this work.

Further speedup is achieved by reducing the resolution of the matching algorithm, what is done by skipping search positions during processing. In this case, the search algorithm does not consider every possible position of the object in the video frames.
Rather, a number of positions are skipped during the convolution of the template with the video frame by increasing the search step size.

The fourth and final search heuristic implemented to speedup cross-correlation template matching consists of using only the non-zero template pixels for the computations. This is advantageous because only edges pixels are used for the cross-correlation computations. All other template pixels are set zero. Thus, by skipping the zero-pixels, computation time is reduced.

6.1. Window-Based Search

As pointed out in [53], a speed-up in template matching is obtained if the search for the object of interest is restricted to a small area of the video frame. The area selected is based on a priori estimate of the object’s location, and the size of this search area is a function of the object’s rate of motion.

Reducing the search area drastically decreases the computation time because only a sub-part of the video frame is used for the convolution with the template. The computational savings are described mathematically by considering the image $I$ and the template $T$ defined in section A. If the search for the desired object is performed over the entire image, the number of elementary operations required for the computations is given by:

$$O_I = 2MN(X - M)(Y - N)$$

Eq. 6.1
This equation comes from the multiplication of the total number of points to check for a match, \((X - M)(Y - N)\), by the number of pixels in the template, \(MN\). The coefficient 2 comes from the additions performed during the cross-correlation computations.

Now, let \(J\) be defined as a sub-image of \(I\) and assume it has an area \(r\) times smaller than the area of \(I\). Considering the \(X\) and \(Y\) dimensions of the image \(I\) respectively reduced by the factors \(r_x\) and \(r_y\), the area reduction factor becomes \(r \equiv r_x r_y\) and the sub-image \(J\) can be defined as \(J(x, y) = I(x, y); 0 \leq x < \frac{X}{r_x}, 0 \leq y < \frac{Y}{r_y}\).

Considering this image and the template \(T\), the number of performed operations for template matching is now given by:

\[
O_J = 2MN\left(\frac{X}{r_x} - M\right)\left(\frac{Y}{r_y} - N\right)
\]

\(\text{Eq. 6.2}\)

Consequently, defining computational speed-up based on the number of elementary operations required for template matching, the following gain is obtained from Eq. 6.1 and Eq. 6.2:

\[
G_{p}(r_x, r_y) = \frac{(X - M)(Y - N)}{\left(\frac{X}{r_x} - M\right)\left(\frac{Y}{r_y} - N\right)}
\]

\(\text{Eq. 6.3}\)

The impact of the window-based searched on the number of template matching operations, i.e., on the computational speed-up, is illustrated in Fig. 6.1. The figure shows
a plot of $G_i(r)$ assuming the template size to be 30x30 pixels and the frame image to be 320x240 pixels.

The results obtained from the application of this technique to track the injection pipette in the blastocyst microinjection application showed a reduction in the average template matching processing time from 2.572s/frame to 88ms/frame when the search area was reduced from full-frame (320x240 pixels) to a 110x70 pixels window. Furthermore, the mean localization error was only increased from 0.337 pixels with a standard deviation of 1.266 to 0.365 pixels with a standard deviation of 1.287. These results represent a speed-up of 29.23 times, which is very close to the theoretical gain of 30.29 obtained from Eq. 6.3.

The window-based tracking of the pipettes is illustrated in Fig. 6.2. The figure shows a processed image displaying the selected tracking area for each object.
6.2. Expanding Search Area

The second template matching speed-up method applied to the vision algorithm consisted of implementing a search strategy based on *a priori* location information. Since small changes in object location are expected to occur between consecutive video frames, it is appropriate to start the search for the new object’s position at its previously known location. Then, the search area is increased from this known location until the object is found.

In the case of the microinjection system, the objects of interest are the tips of the pipettes; therefore the search processes are started at previous pipette locations, and continue at expanding neighborhoods of those locations. This search is terminated, and a pipette is considered found, when the compute cross-correlation value is larger than 95%.
of its maximum possible value. The maximum value is obtained by the cross-correlation of the template with itself.

Note that, from the point of view of processing time, this technique can only improve the performance of the template matching algorithm since in the worst case scenario the total number of operations is the same as it would be if this technique was not implemented. However, this technique has the potential to introduce localization errors since the search is stopped early if the breaking condition is met, and this can happen at a location where the matching is not optimal.

If we consider the previously defined image $I$ and template $T$, the use of the expanding search area technique can reduce the total number of elementary operations to $2MN$ on the best-case scenario, resulting on the following speed-up for the template matching algorithm:

$$G_E = (X - M)(Y - N)$$  \hspace{1cm} \text{Eq. 6.4}

Worst performance: $G_E = 1$

In our system, the use of this technique alone was able to decrease the average template matching processing time from 2.572s/frame to 0.268s/frame when the same injection pipette template mentioned earlier was used (68x28 pixels template). This represents an average speed-up of 9.6 times. Measurements of the localization errors introduced by this technique resulted in a mean error of only 1.955 pixels and a standard deviation of 2.277 pixels.
This expanding search area technique was implemented in conjunction with the previously described window-based method, providing an extra speed-up for computations. For the mentioned template, the average template matching processing time was reduced from 2.572s to 13ms, demonstrating a combined average speed-up of 197.85. The measured localization errors for this case presented a mean of 2.013 pixels and a standard deviation of 2.353 pixels.

6.3. Controlling the Search Step Size

Additional speed-up was obtained using this third method, which consists of skipping pixels during the search for the object of interest. This method is effectively the same as further reducing the search area. The trade-off here is that it decreases the spatial resolution of the template matching, introducing extra errors to the object localizations.

The mathematical formulation to calculate the speed-up provided by this method is very similar to the one developed earlier for the window-based search. In this case, consider \( n \) to be the search step size and the images \( I \) and \( T \) as previously defined. Based on the size of the images and on the value of \( n \), the total number of search points, \( K \), can be computed to be:

\[
K = \left\lfloor \frac{X - M}{n} \right\rfloor \left\lfloor \frac{Y - N}{n} \right\rfloor \quad \text{Eq. 6.5}
\]

Therefore, the total number of elementary operations required for the template matching of the template \( T \) on the image \( I \) is now given by:
\[ O_s = \frac{2MN}{n^2} (X - M)(Y - N) \]  \hspace{1cm} \text{Eq. 6.6}

This result can be compared to Eq. 2 to provide the computational speed-up of this technique, \( G_s(n) \):

\[ G_s(n) = n^2 \]  \hspace{1cm} \text{Eq. 6.7}

For the blastocyst microinjection system a decrease in spatial resolution is not critical because mechanical vibrations constantly cause the position of the pipettes to randomly change by two or three pixels. For this reason, the value of the search step size was set to 2. It can be noted from Eq. 6.7 that this value decreases the number of template matching operations by 75% while the decrease in spatial resolution is kept at a minimum.

Experiments showed that, with the search step size set to two, this technique decreased the average template matching processing time from 2.572s to 0.648s when a template with 68x28 pixels was used. This represents a speedup of 3.97 times. The measured localization errors presented a mean of 2.749 pixels and a standard deviation of 2.036 pixels. When this technique was used in conjunction with the previously described ones, the average processing time was reduced to 11ms, resulting in a combined average speed-up of 233.82 times.

### 6.4. Using only Non-Zero Template Pixels

Cross-correlation template matching is typically based on the edge information extracted from the images because edges can often faithfully represent the shape of objects
of interest. As mentioned in [62], the use of edge-images for matching improves the cross-correlation results because it restricts the computations to the most important pixels of the images — the edges pixels themselves. This produces sharper peaks in the correlation matrix, facilitating the work of the peak detection algorithm used to pinpoint the location of the desired object. However, edges can also be used to speed-up computation.

The speed-up method described here is based on the observation that only a small percentage of the template pixels are edge pixels. The remaining pixels can be ignored during the cross-correlation computations because they are set to zero by the edge detection algorithm. Therefore, this speed-up method consists of: (1) pre-processing the template image to find the location of all edge pixels, and (2) using only these pixels to perform the template matching computations. The computational savings of this method arises from eliminating the need to scan the entire template image during the cross-correlation computations. Speed-up is obtained by simply ignoring pixels that are not edges.

Considering the previously defined image $I$ and template $T$, if we assume that the fraction of edge pixels in the template image is $p$, the total number of operations required for the template matching can be expressed as:

$$O_{NZ} = 2pMN(X - M)(Y - N)$$

Eq. 6.8

Therefore, the computational savings of using only non-zero template pixels is given by:

$$G_{NZ}(p) = \frac{1}{p}$$

Eq. 6.9
In the case of our application, the typical value for \( p \) is 0.07 for the holding pipette and 0.15 for the injection pipette. These ratios can be observed in Fig. 6.3, which shows typical templates acquired for the injection and holding pipettes.

The use of this speed-up method alone reduced the average template matching processing time from 2.572s to 0.488s per frame when a 68x28 injection pipette template with \( p = 0.1597 \) was used. The localization errors in this case had a mean of 0.446 pixels with a standard deviation of 2.05 pixels.

6.5. Combining the Speed-up Techniques

When all four techniques were applied, the average template matching processing time of each frame was reduced to 3ms, demonstrating a total speed-up of 857.33 times. The localization errors in this case presented a mean of 2.843 pixels and a standard deviation of 1.964 pixels. The analysis of the results also showed that the pipettes were correctly localized 99.8% of the time.

Consequently, the video processing was able to work in real-time, updating the location of the blastocyst and pipettes 10 times per second. This rate has proven to be
adequate for the control of the microinjection system, a fact that has also been verified in [8].

A summary of the speed-up provided by the described methods is presented in Table 6.1. The theoretical values and the achieved speed-ups are shown, together with the measured localization errors introduced by each technique. The presented measurements are based on the analysis of 1000 simulated video frames for each case, which were created using a 68x28 pixels injection pipette template. Each of the frames was blurred with a Gaussian mask with variance 1 and corrupted by Gaussian noise with variance 100. Impulse noise with 5% probability was also added to all video frames.

6.6. Speeding up the SSD Template Matching Algorithm

Two of the techniques used to speedup the cross-correlation template matching algorithm were also implemented on the SSD algorithm. These included the window-based search method and the use of increased search steps. In this case, the localization processing time for a 24x20 pixels template was reduced from 3.871s/frame to 24ms/frame by trimming down the search area from 320x240 to 60x60 pixels, and by setting the search step to 2.
Table 6.1. Summary of the Template Matching Speed-up Techniques

<table>
<thead>
<tr>
<th>Method</th>
<th>Number of Elementary Operations</th>
<th>Computational Gain</th>
<th>Theoretical Speed-up</th>
<th>Experimental Speed-up</th>
<th>Localization Error (pixels)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Standard Cross-Correlation</td>
<td>(2MN(X-M)(Y-N))</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>0.337, 1.266</td>
</tr>
<tr>
<td>Window-based Search</td>
<td>(2MN\left(\frac{X}{r_x} - M\right)\left(\frac{Y}{r_y} - N\right))</td>
<td>(\frac{(X-M)(Y-N)}{\left(\frac{X}{r_x} - M\right)\left(\frac{Y}{r_y} - N\right)})</td>
<td>30.29</td>
<td>29.23</td>
<td>0.365, 1.287</td>
</tr>
<tr>
<td>Expanding Search Area</td>
<td>Min (2MN)</td>
<td>((X-M)(Y-N))</td>
<td>Max 53420</td>
<td>197.85</td>
<td>1.955, 2.277</td>
</tr>
<tr>
<td></td>
<td>Max (2MN(X-M)(Y-N))</td>
<td>1</td>
<td>Min 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Controlling the Search Step Size</td>
<td>(\frac{2MN}{n^2}(X-M)(Y-N))</td>
<td>(n^2)</td>
<td>4</td>
<td>3.97</td>
<td>2.749, 2.036</td>
</tr>
<tr>
<td>Using only Non-Zero Template Pixels</td>
<td>(2pMN(X-M)(Y-N))</td>
<td>(1/p)</td>
<td>6.26</td>
<td>5.27</td>
<td>0.446, 1.432</td>
</tr>
<tr>
<td>Applying all four techniques</td>
<td></td>
<td></td>
<td>758.46</td>
<td>857.33</td>
<td>2.843, 1.964</td>
</tr>
</tbody>
</table>

Frame Image: \(X=320, Y=240\)  Template: \(M=68, N=28, p=0.1597\)  \(r_x=2.909, r_y=3.429, n=2\)  Number of Analyzed Images: \(1000/\text{measurement}\)
6.7. Conclusions

This chapter described the implementation and the evaluation of four speedup techniques to enable real-time cross-correlation template matching on video images. These techniques were necessary because pipette localizations using the basic cross-correlation template matching algorithm resulted in unacceptably long processing time per video frame. Here, reduction in the processing time was achieved by the application of simple search heuristics. These included the use of search windows, expanding search areas, increased search step size, and the use of only non-zero template pixels for matching. The combined use of these techniques was shown to provide an average template matching speed-up of over 857 times. Furthermore, the speedup techniques did not significantly affect the localization errors on images presenting low noise levels, as will be shown by the template matching evaluation experiments presented in the next chapter.
7. **Evaluation of the Template Matching Algorithms**

The second series of evaluation experiments associated with the developed vision system was designed to assess the performance and the robustness of the template matching algorithms used for pipette localizations. Similarly to the evaluation of the blastocyst processing algorithms, this part of video system was evaluated using both synthetic and real images obtained directly from the microinjection system.

During the experiments with synthetic images, video frames containing a pipette at random positions were created and analyzed by the video processing algorithms. The results obtained were then compared to the true pipette locations, providing error measures for each processed frame. This process was employed to investigate the accuracy and the robustness of the template matching algorithms, which were realized by the introduction of increasing levels of noise into the experiments. The performed experiments also included the evaluation of the impact of each template matching speedup technique on the accuracy and robustness of the localizations.

After obtaining performance metrics from synthetic data, the template matching algorithms were evaluated on real images. In this case, two sets of video frames were manually assembled for the experiments: One with holding pipette images, and the other with injection pipette images. These sets were used as the baseline to evaluate the implemented matching algorithms in the same way it was done for the simulated video frames. As before, the experiments performed here evaluated the impact of noise and speedup techniques on the accuracy and robustness of the template matching algorithms.
7.1. Experiments with Synthetic Images

The synthetic images used for the evaluation experiments were created by placing a simulated injection pipette tip at random positions within blank video images. These video frames were later analyzed by the developed template matching algorithm, and estimates of the pipette locations were obtained. Such estimates were then compared to the true pipette locations, providing error measurements for each processed video frame. This method was used to examine the error introduced by the template matching speedup techniques and the robustness of the algorithm against additive Gaussian, impulse noise, missing pixels and Gaussian blur.

7.1.1. Video Simulation

The pipette image used to create the synthetic video frames was simulated using straight lines, as shown in Fig. 7.1. It measured 68 by 28 pixels. After being created, a percentage of its pixels was randomly selected and transferred as a group to a random location on a blank video frame, thus creating a noisy pipette image at a random position. Further image distortion was obtained by the addition of user-adjustable levels of Gaussian noise, impulse noise and Gaussian blur. Examples of noisy video frame images generated by this method are shown in Fig. 7.2.

![Fig. 7.1. Simulated injection needle template and its edge-image used for matching.](image-url)
The implemented cross-correlation template matching algorithm worked perfectly on clean simulated images; however such images do not approximate the real working conditions of the algorithm. Real images are typically noisy and somewhat blurred, thus better represented by distorted simulated images. Therefore, the performance of the algorithm was studied by incrementally changing each noise level and measuring the resulting average localization error. Each noise condition was tested based on the analysis of 1000 simulated images. The obtained results are presented below.
7.1.2. Impact of Gaussian Noise

The performed experiments demonstrated that the introduction of additive Gaussian noise to the simulated images had negligible effect on the performance of the template matching algorithm. Tests of the localization error as a function of the introduced Gaussian noise level resulted in values greater than zero only when using the increased search step speedup technique. However, the measured errors were independent of the noise level, and consequently were only due to the speedup technique. This result can be seen in Fig. 7.3(a).

When all four speedup techniques were active during the template matchings on simulated images corrupted by Gaussian noise, the results shown in Fig. 7.3(b) were obtained. Once again it was seen that the measured localization error was a function of the search step alone. The error noted to be independent of the Gaussian noise.

Figures 7.3(c) and (d) show sample results of the localization error distribution for the experiments performed with the Gaussian noise variance set to 100 and all four speedup techniques active. From those plots it can be seen that the localization was perfect for 30% of the cases, and presented an error less than 3.5 pixels for 90% of the analyzed images.
7.1.3. Impact of Impulse Noise

Similar experiments were conducted to measure the impact of impulse noise on the performance of the template matching algorithm. The obtained results demonstrated once again that the speedup techniques based on reduced search areas, non-zero template pixels and expanding search areas do not introduce localization errors to the matching results. On the other hand, the introduction of impulse noise was shown to have a major impact on the
edge-detection algorithm, causing it to find numerous false edges. This increase in the number of false edges caused, in turn, a great increase in the measured localization errors. This problem was alleviated by the use of Gaussian filters (low-pass filters), as it was logically expected.

Further experimentation showed that, even when filtering the corrupted images prior to processing, the presence of impulse noise caused the performance of the algorithm to deteriorate when the search step size was set to a value greater than 2. In contrast, the impulse noise had little effect when the search step size was set to 1 or 2. In those cases the localization errors were observed to be independent of the impulse noise levels tested. Figure 7.4(a) presents results that illustrate these observations, and Fig. 7.4(b) presents the results obtained with all speedup techniques active and the search step set to 2. From the plot in Fig. 7.4(b) it can be seen that the mean localization error was almost solely due to the search step being set to 2.

Sample PDF and CDF plots are also shown in Fig. 7.4. These were obtained from experiments with the impulse noise probability set to 10% and with all four speedup techniques active. It can be noticed that the plots presented here are very similar to the ones presented in Fig. 7.3, demonstrating that the major source of errors is the increased search step size rather than the additive noise. From Fig. 7.4(d) it can be seen that the measured localization error for the tested conditions was less than 3.5 pixels for 90% of the images.
7.1.4. Impact of Reduced Number of Object Pixels

A set of experiments was performed to evaluate the impact of “erasing” pixels that represent the object on the simulated images. The idea here, as in the case of the simulated blastocysts in Chapter 5, was to create situations related to real cases of edge detection failure, i.e., when a percentage of the edge pixels is not found or when object occlusions...
occur. Therefore, the images simulated for this set of experiments were generated from a randomly selected percentage of the injection needle pixels.

The experimental results showed that the implemented cross-correlation template matching algorithm is robust against reductions in the number of pixels representing the object of interest. It was noted that the performance of the algorithm with search step size set to 1 only started to degrade when the percentage of object pixels was down to 20%. On the cases with search step size set to 2 or 3, reducing the percentage of pixels actually improved the algorithm performance. These unexpected outcomes are believed to result from the reduction in the number of false matches induced by the reduction in the number of edge pixels in the images.

The experiments conducted here also reconfirmed that the other speedup techniques (expanding search area and use of only non-zero template pixels) do not introduce localization errors to the matching algorithm.

Results that illustrate the former observations are shown in Fig. 7.5(a). In those experiments a 3x3 Gaussian mask filter was used to pre-process the generated video frames, and the percentage of object pixels used to create those images was incrementally reduced from 100% to 10%. Figure 7.5(b) presents the results obtained when all four speedup techniques were used and the search step was set to 2. As before, the obtained mean localization error was noted to be almost solely due to the increased search step size. This can be verified by noticing the similarities between the obtained PDF and the CDF plots shown in the Figures 7.3, 7.4, and 7.5.
Fig. 7.5. Impact of reducing the percentage of object pixels transferred to the test images on the cross-correlation template matching. (a) Mean localization error for search step sizes set to 1, 2 and 3. (b) Mean localization error with all four speedup techniques active and search step set to 2. (c) PDF plot of the experimental data obtained when the percentage of object pixels used was set to 30 in (b). (d) CDF plot of the same data as in (c).

7.1.5. Impact of Gaussian Blur

The last set of experiments performed with simulated images dealt with the impact of blur on the localization errors. As explained before, this type of image distortion occurs frequently on real images, mainly due to out-of-focus images. Consequently, this series of
experiments was important to evaluate how much out-of-focus the images can be before the matching algorithm fails.

During these experiments, the Gaussian blur variance was increased from 0 to 20 and different configurations of the cross-correlation template matching algorithm were tested. The obtained results showed that the algorithm is not affected by blur with variance up to 2. However, the magnitude of the errors starts to increase with the blur for larger variance values. Figure 7.6(a) illustrates this observation.

Similarly to the results obtained in the previous experiments, the results obtained here showed no change in the mean localization errors when the expanding search area method was used. The same was observed when using only non-zero template pixels to perform the matchings. On the other hand, the experiments showed that a search step size larger than one is not the only cause for localization errors. Gaussian blur also affects the algorithm performance, as can be verified from Fig. 7.6(b).

The introduction of errors by Gaussian blur can also be noticed from the PDF and CDF plots for the cases when all speedup techniques were active and the search step was set to two. For images blurred with a Gaussian mask with variance less than 2, the PDF and CDF plots obtained are very similar to the ones presented in the Figures 7.3 through 7.5. However, when the images are blurred with larger variances the distribution of localization errors change. This change can be seen from Fig. 7.6(c) and Fig. 7.6(d), which were obtained from experiments using Gaussian blur with variance 4. The CDF plot for this case shows that a 90% confidence margin only guarantees the error to be less than 4.5 pixels.
7.2. Experiments with Real Images

After obtaining performance metrics from synthetic data, the template matching algorithms were evaluated on real images acquired directly from the microinjection system. In this case, two sets of images were assembled for the experiments: one with 300 video frames containing a holding pipette, and another with 893 video frames containing...
an injection pipette. The first set was used to evaluate the holding pipette localizations based on cross-correlation template matching algorithm. The second set was used to evaluate the application of both cross-correlation template matching and SSD template matching algorithm to locate the injection pipette.

After assemblage, both sets were manually analyzed for the determination of the correct pipette coordinates in each video frame. This resulted in the construction of the baselines used to evaluate the template matching algorithms in the same way it was done for the simulated video frames, i.e., these baselines were used to study the influence of noise and the impact of the speedup techniques on the localization errors.

7.2.1. Performance from Undistorted Images

The evaluations of the template matching algorithms based on undistorted real images were manually performed during the assemblage of the mentioned baselines. The method used consisted of selecting a group of images containing the object of interest, and then running the corresponding matching algorithm on each one of them. During this process, the matching outcome for each image was manually evaluated, so a measure of the algorithm performance was obtained. The baselines were subsequently built using only the images associated with correct localizations.

In the case of the baseline used for testing the cross-correlation template matching algorithm, the 300 images were processed to locate the holding pipette tip shown in Fig. 7.7(a). None of the speedup techniques were used and no image filtering was performed during these tests. The obtained results showed that the matching algorithm had a 100%
success rate, i.e., it provided the correct pipette localization in all 300 images. As an example, Fig. 7.7(b) presents one of the processed video frames obtained during this analysis.

The construction of the baseline used to evaluate the injection pipette localizations followed the same procedure described above. The employed template image is shown in Fig. 7.7(c) and a processed video frame is shown in Fig. 7.7(d). A total of 893 video frames containing the injection pipette were processed by the cross-correlation template matching algorithm, and manual analysis indicated that the localizations were correct for
887 of them. Therefore, the algorithm demonstrated a 99.33% success rate for undistorted real images, and allowed the construction of a baseline with 887 images. This baseline was then used to evaluate the SSD template matching algorithm, which provided correct localizations for all images. This means that the SSD algorithm presented a success rate that is at least 99.33% for the original images set.

After the baselines were built, the evaluations under noisy conditions were performed in the same methodical way as before, i.e., by corrupting each image with incremental levels of noise and measuring the resulting localization errors. The types of distortions used here were Gaussian noise, impulse noise and Gaussian blur. The obtained results are described next.

7.2.2. Impact of Gaussian Noise

The performed experiments showed that the cross-correlation template matching algorithm is affected by the addition of Gaussian noise to the real images, but the introduced errors are negligible for low noise levels. This can be seen from Fig. 7.8(a), which also demonstrates that the noise level at which the search for the holding pipette begins to be affected depends on the employed search step size. The plot also shows that the robustness of the algorithm against Gaussian noise decreases as the search step is increased. Similar results were obtained when all speedup techniques were active, as demonstrated by Fig. 7.8(b). In this case, the algorithm presented a mean localization error around 1 pixel for all Gaussian noise levels with variance less than 140. Therefore, the algorithm proved to be robust against low levels of Gaussian noise. Figures 7.8(c) and
7.8(d) reaffirm this conclusion by showing that of 90% of the localizations presented an error less than 2 pixels when the Gaussian noise variance was set to 100.

In the case of using cross-correlation to locate the injection pipette, Fig. 7.9(a) shows that bad results were obtained for all noise levels and search step sizes. These results proved that the injection pipette lacks good edge features to track, so the SSD algorithm was used for the task. Evaluation results for this case are presented by the plots in Fig. 7.10, which demonstrates that Gaussian noise had negligible impact on the SSD template matching algorithm. The plots also show that an increase in the search step results in increased localization errors. Nevertheless, for a search step of 2, the mean error was less than 0.9 pixels for all levels of noise tested. As an example, Fig. 7.10(c) shows that when using a search step of 2 and images distorted by Gaussian noise with variance 100, the localization error was less then 1.4 pixels 90% of the time.
Fig. 7.8. Impact of Gaussian noise on the cross-correlation template matching used to locate the holding pipette on real images. (a) Mean localization error for search step sizes set to 1, 2 and 3. (b) Mean localization error with all four speedup techniques active and search step set to 2. (c) PDF plot of the experimental data obtained when the Gaussian noise variance was set to 100 in (b). (d) CDF plot of the same data as in (c).
Fig. 7.9. Impact of Gaussian noise on the cross-correlation template matching used to locate the injection pipette on real images. (a) Mean localization error for search step sizes set to 1, 2 and 3. (b) Mean localization error with all four speedup techniques active and search step set to 2. (c) PDF plot of the experimental data obtained when the Gaussian noise variance was set to 100 in (b). (d) CDF plot of the same data as in (c).
Fig. 7.10. Impact of Gaussian noise on the SSD template matching used to locate the injection pipette on real images. (a) Mean localization error for search step sizes set to 1, 2 and 3. (b) PDF plot of the experimental data obtained when the Gaussian noise variance was set to 100 and the search step was set to 2. (c) CDF plot of the same data as in (b).
7.2.3. Impact of Impulse Noise

The experiments conducted to measure the impact of impulse noise on the search algorithms showed that this type of noise can cause significant performance degradation. The holding pipette localizations using the cross-correlation algorithm with search step set to 1 was only robust for impulse noise probabilities less than 2%. The same was observed when using a search step of 2 pixels, although the mean errors in this case were a little higher. For a larger search step, Fig. 7.11(a) shows that the algorithm presented no robustness against impulse noise. On the other hand, the use of the other two speedup techniques (non-zero template pixels and expanding search area) was shown to preserve the mean error level and the algorithm robustness for impulse noise levels up to of 2%. This is demonstrated by the plot in Fig. 7.11(b), and also by the PDF and CDF plots in Figures 7.11(c) and 7.11(d). These plots show that the localization error was less than 8 pixels for 80% of the analyzed images when the noise level was set to 2% and the search step to 2 pixels.

The use of the cross-correlation algorithm to locate the injection pipette once again presented bad results in the presence of noise. The plots in Fig. 7.12 show that the algorithm only presented reasonable performance when the search step size was set to 1 and the impulse noise probability was less than 2%. On the other hand, the search for the injection pipette using the SSD algorithm with search step less than or equal to 2 presented excellent results. Figure 7.13(a) shows that the algorithm was robust against impulse noise with probabilities up to 10% in that case. As an example, Fig. 7.13(c) shows that the
Localization errors were less than 1.3 pixels for 90% of the analyzed images when the impulse noise level was set to 2%.

Fig. 7.11. Impact of impulse noise on the cross-correlation template matching used to locate the holding pipette on real images. (a) Mean localization error for search step sizes set to 1, 2 and 3. (b) Mean localization error with all four speedup techniques active and search step set to 2. (c) PDF plot of the experimental data obtained when the impulse noise probability was set to 2% in (b). (d) CDF plot of the same data as in (c).
Fig. 7.12. Impact of impulse noise on the cross-correlation template matching used to locate the injection pipette on real images. (a) Mean localization error for search step sizes set to 1, 2 and 3. (b) Mean localization error with all four speedup techniques active and search step set to 2. (c) PDF plot of the experimental data obtained when the impulse noise probability was set to 2% in (b). (d) CDF plot of the same data as in (c).
Fig. 7.13. Impact of impulse noise on the SSD template matching used to locate the injection pipette on real images. (a) Mean localization error for search step sizes set to 1, 2 and 3. (b) PDF plot of the experimental data obtained when the impulse noise probability was set to 2% and the search step was set to 2. (c) CDF plot of the same data as in (b).
7.2.4. Impact of Gaussian Blur

The final set of experiments with real images evaluated the impact of blurring on the template matching localization errors. The obtained results showed that the only case significantly impacted by increasing levels of blur was the implementation of the cross-correlation algorithm to search for the injection pipette. In this case, the algorithm only presented acceptable error levels for unary search steps and blurring with variance less than 4. This can be seen from the plots in Fig. 7.15, which provide extra evidence that cross-correlation template matching is not appropriate to search for the injection pipette in this application.

Conversely, Gaussian blurring was shown to have little impact on the localization of the holding pipette through cross-correlation template matching, as Fig. 7.14(a) demonstrates. The same was observed for the injection pipette localizations using SSD template matching, which was shown to be even more robust against blurring. The plots in Fig. 7.14 and Fig. 7.16 also show that the use of speedup techniques did not significantly impacted the performance of these algorithms. For example, when the Gaussian blur variance was set to 4 and the search step to 2, 90% of the holding pipette localizations presented errors less than 1.75 pixels. Under the same test conditions, 90% of the injection pipette localizations using the SSD algorithm presented errors less than 1.3 pixels.
Fig. 7.14. Impact of Gaussian blur on the cross-correlation template matching used to locate the holding pipette on real images. (a) Mean localization error for search step sizes set to 1, 2 and 3. (b) Mean localization error with all four speedup techniques active and search step set to 2. (c) PDF plot of the experimental data obtained when the Gaussian blur variance was set to 4 in (b). (d) CDF plot of the same data as in (c).
Fig. 7.15. Impact of Gaussian blur on the cross-correlation template matching used to locate the injection pipette on real images. (a) Mean localization error for search step sizes set to 1, 2 and 3. (b) Mean localization error with all four speedup techniques active and search step set to 2. (c) PDF plot of the experimental data obtained when the Gaussian blur variance was set to 4 in (b). (d) CDF plot of the same data as in (c).
Fig. 7.16. Impact of Gaussian blur on the SSD template matching used to locate the injection pipette on real images. (a) Mean localization error for search step sizes set to 1, 2 and 3. (b) PDF plot of the experimental data obtained when the impulse noise probability was set to 2% and the search step was set to 2. (c) CDF plot of the same data as in (b).
7.3. Conclusions

This chapter described the evaluation of the template matching algorithms developed to locate and track the micromanipulation pipettes in the blastocyst microinjection system. As mentioned in Chapter 4, cross-correlation template matching was implemented to search for the holding pipette, and both cross-correlation and sum-of-squared-differences (SSD) template matching were tested for the injection pipette localizations. The accuracy and the robustness of these algorithms were evaluated here using simulated and real images.

In the case of the holding pipette localizations, experiments proved that the implemented cross-correlation template matching algorithm is effective for this task. The algorithm presented a 100% success rate for pipette localizations on real images, and demonstrated to be robust against blur and low levels of Gaussian noise, impulse noise and partial occlusions.

The application of cross-correlation template matching to locate the injection pipette within real images was also evaluated in this chapter. The obtained results demonstrated that the algorithm is not appropriate for this task because the injection pipette lacks good edge features to track. Therefore, the SSD template matching algorithm was selected as the alternative. In this case, evaluations based on images captured directly from the developed microinjection system showed that the SSD algorithm performed very well on clean images, obtaining a 100% localization success rate on the baseline set. Furthermore, this algorithm was shown to be robust against blurring, Gaussian noise, and impulse noise. On the other hand, the implemented SSD algorithm was shown to be
slower than the sped-up cross-correlation algorithm, affecting the processing time of the entire vision system. Nevertheless, reducing the template image size and the search window size solved this problem without affecting the localization performance.

Based on the evaluation results presented here and in Chapter 5, and on the fact that the microinjection images captured under normal conditions present very low noise levels, the implemented vision processing system was deemed able to successfully accomplish its objectives. Furthermore, since the imaging conditions under the microscope present uniform illumination, good focus and a clean environment, the best success rates obtained during the evaluations represent the real system performance.
8. **BLASTOCYST MICROINJECTION AUTOMATION**

From the recognition of an ever-growing need for improved consistency and efficiency of embryo micromanipulations, it is clear that full process automation is mostly desirable. An automated microinjection system can eliminate the randomness associated with human controlled operations, and possibly increase throughput, improve microinjection quality, and reduce costs through predictable and consistent actions. Furthermore, automation can free operators from this time consuming task, allowing them to work on more rewarding tasks such as research or data analysis. Therefore, major research efforts were invested towards the design and implementation of an automated blastocyst microinjection system during this work. These developments, and the obtained results, are described in this chapter.

8.1. **System Configuration for Automatic Microinjections**

A new microinjection system configuration was created to convert a setup convenient for human operators to one that is more appropriate for automatic control. This was accomplished by adding two pipettes and respective computer controlled micrometer syringes to the system. The new pipettes were used to feed and collect blastocysts from the injection area. As a consequence, their tip diameters were selected to be in the order of 120µm to allow for the entry of a single blastocyst without distortion. The design of this new system configuration is presented in Fig. 8.1.
Fig. 8.1. Four-pipettes system design. This configuration was created to facilitate the automation of the consecutive blastocyst microinjections.

The addition of a feeder and a collection pipette to the microinjection setup simplifies the design and implementation of control algorithms for automation. Simplification occurs because feeding and collecting blastocysts directly to and from the injection area avoids the need to move the XY stage to search for them. Consequently, this setup reduces the number of control variables in the system by reducing the number of moving parts during microinjections.

The two extra pipettes also simplify the requirements for the video processing algorithm by keeping the injected and the non-injected blastocysts separated from each other. Therefore, the vision algorithm is not required to be able to differentiate between the two blastocyst states.

In addition, this new system setup is advantageous for the processing of several consecutive blastocyst microinjections because it has the potential to work as a production
line. This should reduce the amount of time required for the microinjections, thus contributing to improved productivity.

Pictures of the expanded microinjection system setup are shown in Fig. 8.2. A block diagram describing the system configuration is presented in Fig. 8.3.

Fig. 8.2. Microinjection system setup for automatic blastocyst microinjections.
Fig. 8.3. System configuration for automatic blastocyst microinjection. The blocks marked in red were added to the semi-automated system to make it more appropriate for automatic control.
The updated system configuration diagram in Fig. 8.3 shows that modifications to the user interface and control software were also required. These included the addition of inputs for user control of the feeder and collection micrometer syringes, which were realized by setting up control functions to keys on the computer’s keyboard. The use of the keyboard was necessary due to the limited number of joystick inputs, but resulted in a welcome division of tasks between the operator’s hands.

8.2. Teleoperated Trials with the New System Configuration

Once the new system configuration was implemented, it was evaluated by conducting microinjection experiments under teleoperated manual control. These trials were performed concurrently with the semi-automated microinjection experiments described in Chapter 3. Again, both expert and amateur operators tested this new system configuration.

The performed experiments showed that the new setup is appropriate for automated of blastocyst microinjections. First, it was found that applying suction to the holding pipette prior to the delivery of a blastocyst, the blastocyst was captured immediately once it was fed to the injection area, i.e., as soon as it exits the feeder pipette. A similar result was observed for blastocyst collection, i.e., the creation of suction on the collection pipette prior to the blastocyst’s release from the holder causes it to move directly from one pipette to the other once it is released. These observations were important and encouraging because they demonstrated the possibility of creating a simple controller for the automatic delivery and collection of blastocysts from the injection area.
In addition to the advantages described above, the four-pipette setup demonstrated potential to speed-up multiple blastocyst microinjections in two ways: 1) by avoiding the need to search for the embryos in the working well since they are kept at known locations, i.e., inside the pipettes; and 2) by avoiding the need to identify non-injected blastocysts with the embryo population since injected and non-injected blastocysts are kept separated from each other.

Some of the observed problems with the four-pipette system included longer setup time and difficulties controlling the fluid motions in the feeder and collection pipettes. The first problem arose from the fact that two extra pipettes had to be fabricated, installed and aligned during system setup. The second problem was mainly caused by the lack of fluid stability within the pipettes and by the quality of the employed micrometer syringes. In this case, random oscillations of fluid motion within the feeder and collection pipettes often caused the release of extra blastocysts to the injection area. This problem was minimized by keeping the blastocysts far away from the tip of the pipettes.

8.3. Automation Algorithms

The next step towards blastocyst microinjection automation consisted of the development of control routines for automating basic process tasks. These tasks were identified in Chapter 3, where they were presented as a sequence of rules likely to result in successful microinjections. Here, the automation of those tasks were based on the knowledge gathered during the teleoperated microinjections described in Chapter 3, and depended heavily on information from the vision system presented in Chapter 4.
The created control routines included algorithms for automatic delivery, capture, orientation, release and collection of blastocysts. Algorithms were also created for injection pipette motion control and for blastocyst microinjection. These algorithms are described below.

### 8.3.1. Automatic Blastocyst Delivery

The delivery of a blastocyst to the injection area is the first task that needs to be completed during the microinjection process. This task was automated by commanding the feeder micrometer syringe to slowly rotate clockwise until a blastocyst was delivered and positively identified in the injection area. The automation algorithm developed for this task is shown in Fig. 8.4.

### 8.3.2. Automatic Blastocyst Capture

As soon as a new blastocyst enters the injection area it should be captured by the holding pipette. Therefore, an automation algorithm was created to accomplish this task. This algorithm assumes that a blastocyst is present in or near the injection area, and commands the holder micrometer syringe to rotate counterclockwise until the blastocyst is captured by the holding pipette. The task is deemed completed when the blastocyst is located next to the holding pipette tip, i.e., when the following equations are true:
Fig. 8.4. Blastocyst delivery algorithm.

Fig. 8.5. Blastocyst capture algorithm.
In the equations above, \((X_{\text{holder}}, Y_{\text{holder}})\) are the coordinates of the holding pipette tip, \((X_{\text{blast}}, Y_{\text{blast}})\) are the coordinates of the blastocyst center, and \(R_{\text{blast}}\) is the blastocyst radius. All of these parameters were obtained in real-time from the vision processing system. The flow diagram for this automation algorithm is presented in Fig. 8.5.

8.3.3. Automatic Blastocyst Release

There are basically two situations in which the blastocyst must be released from the holding pipette: 1) during orientation adjustments prior to injection; and 2) during its collection after microinjection. Hence, an algorithm was created to automatically control the blastocyst release task.

The developed automation algorithm is summarized by the flow chart in Fig. 8.6. The figure shows that the algorithm is basically the inverse of the one developed for blastocyst capture. Here, the holder micrometer syringe is slowly rotated clockwise until the blastocyst is no longer next to the holding pipette tip. The decision equations for task completion are given by:

\[ (X_{\text{blast}} - R_{\text{blast}}) < X_{\text{holder}} \quad \text{Eq. 8.1} \]

\[ Y_{\text{holder}} - 0.25R_{\text{blast}} < Y_{\text{blast}} < Y_{\text{holder}} + 0.25R_{\text{blast}} \quad \text{Eq. 8.2} \]

\[ Y_{\text{blast}} > Y_{\text{holder}} + 0.25R_{\text{blast}} \quad \text{Eq. 8.4} \]

\[ Y_{\text{blast}} < Y_{\text{holder}} - 0.25R_{\text{blast}} \quad \text{Eq. 8.5} \]
Therefore, the blastocyst release task was deemed completed when any one of the equations above was true.

8.3.4. Automatic Blastocyst Orientation

During preparations for injection the blastocyst’s orientation must be verified and adjusted. This is necessary to avoid injecting through the ICM, as explained in previous chapters. Here, the push-pull method of blastocyst orientation described in Chapter 3 was used for automation. This method was selected for its simplicity. As a result, the algorithm for automatic blastocyst orientation consisted of releasing and capturing the
blastocyst until an acceptable orientation for injection was detected. This algorithm is presented in Fig. 8.7.

Fig. 8.7. Blastocyst orientation algorithm.

8.3.5. Automatic Blastocyst Collection

Blastocyst collection is performed after the injection pipette retracts from the blastocyst. This task completes the microinjection cycle, so it involves releasing the blastocyst from the holding pipette and collecting it through suction on the collection pipette. The automation algorithm for this task is shown in Fig. 8.8.
8.3.6. Automatic Motion Control

One of the main system characteristics with major impact on the success of blastocyst microinjections is the quality of the injection pipette motion control. The importance of this feature was discussed in Chapter 1, in Chapter 3, and in cited references [5, 7, 8, 19]. Consequently, an algorithm was designed and implemented to enable precise pipette motion control in the blastocyst microinjection system. This algorithm is presented below.
8.3.6.1. Visual Servoing

The motion control system developed for the micromanipulation robot was based on visual servoing techniques. This means that all motion commands were computed solely based on visual feedback, i.e., on the localization information provided by the video processing system. This was possible because robot motions caused changes to the captured video images, which resulted in changes to the vision system’s output parameters.

The implemented control structure consisted of a PID (proportional-integral-derivative) control loop in which the position errors were computed directly from the 2-D image parameter space. In this case, the target coordinates and the pipette tip coordinates were defined with respect to the image reference frame. Therefore, the position errors used for control were obtained from the difference between those two points, as defined by Eq. 8.6.

\[ E = T - P \]  
\[ \text{Eq. 8.6} \]

where \( T \in \mathbb{R}^2 \) is the target location, and \( P \in \mathbb{R}^2 \) is the pipette location, both defined in terms of image coordinates:

\[ T = \begin{bmatrix} x_T \\ y_T \end{bmatrix} \]  
\[ \text{Eq. 8.7} \]

\[ P = \begin{bmatrix} x_P \\ y_P \end{bmatrix} \]  
\[ \text{Eq. 8.8} \]

The block diagram for the implemented PID control system is presented in Fig. 8.9. The diagram illustrates the relationship between the system variables and shows that the
control parameters include proportional gain ($K_p$), integral gain ($K_i$) and derivative gain ($K_d$). These parameters were used for control system calibration and provided the appropriate mapping of commands generated in the image space to motion commands in the target space.

The PID control equations for the developed 2-D visual servoing system were given by:

$$E_i(k) = E_i(k-1) + \frac{\Delta t}{2} (E(k) + E(k-1))$$  \hspace{1cm} \text{Eq. 8.9}

$$C(k) = K_p E(k) + K_i \frac{E_i(k)}{\Delta t} + K_d \frac{(E(k) - E(k-1))}{\Delta t}$$  \hspace{1cm} \text{Eq. 8.10}

where $E_i$ is the integral of the error signal; $\Delta t$ is the sample period; and $C = [C_x \quad C_y]^T$ is the motion command vector sent to the robot driver.

The above control system configuration was feasible for this application because the task space was restricted to 2-D during the microinjection operations. This meant that the robot was constrained to move along the X and Y axes only, guaranteeing that the
vertical alignment of the holding and injection pipettes was not affected. Furthermore, the control system was able to compensate for small misalignments between the robot and the image axes, so it successfully avoided expensive system calibration.

8.3.6.2. Automation Algorithm

Based on the visual servoing system described above, the automatic control of the injection pipette motions followed the algorithm presented in Fig. 8.10. The algorithm shows that, once a target coordinate is commanded, the visual servoing system takes the pipette to the desired location. This positioning process is stopped, and the task is deemed completed, when the error between the target the actual pipette position is small (less than 2 pixels).

![Fig. 8.10. Automatic pipette positioning algorithm.](image-url)
8.3.7. ES Cells Delivery

The delivery of ES cells inside the blastocyst cavity was not automated in this research. Consequently, we required an operator to manually perform this task using the joystick.

The automated of this part of the system is deemed future work. Getting this task automated requires the development of extra video processing algorithms to detect and count the number of delivered ES cells. This is not a trivial task because the ES cells are very small and typically move very fast inside the injection pipette. Furthermore, controlling them proved difficult. The problems associated with this task were detailed in Chapter 3 and included problems with sticky cells and the formation of clogs in the injection pipette. Both of these difficulties were aggravated by the need to preload the pipette with hundreds of cells prior to beginning the microinjection process. Chapter 3 also mentioned that progress towards the minimization of these problems was achieved by keeping the cells in 1% glycerol. However, more research is still required to create an effective protocol and a reliable setup for handling these cells. Therefore, the automation of this task will only be viable on the current microinjection system setup once these other problems are resolved.

8.3.7.1. Proposed Automation Algorithm

Even though ES cells delivery was not automated, an automation algorithm was developed for it. The algorithm, which is presented in Fig. 8.11, explicitly shows the need for feedback on the number of delivered cells. This count provides the basis for deciding
upon the task completion, so it will be essential for automation. Once that count is
obtained, the proposed algorithm shows that automation will be achieved by simply
rotating the injection micrometer syringe clockwise until the desired number of ES cells is
delivered.

8.3.7.2. Proposed ES Cells Counter Algorithm

A possible method for counting the number of delivered ES cells is proposed by the
algorithm in Fig. 8.12. The main idea behind this algorithm is the monitoring of a small
section of the injection pipette for the detection of cells passing through it. This detection
may be possible by defining a small monitoring area which can only detect one cell at a
time. In this way, the ES cells count can be incremented every time a new cell is detected
in the defined area.

The cells counter algorithm proposed above is illustrated by Fig. 8.13. The figure
also shows that a possibly good location for the detection area may be the section of the
pipette just outside the blastocyst. This location is likely to be an ideal detection site
because it avoids background image noise from blastocyst inner structures, and because the
image focus in that part of the pipette is close to the focus on the pipette’s tip.
Fig. 8.11. Proposed ES cells delivery algorithm.

Fig. 8.12. Proposed ES cells counter algorithm for vision processing.
8.3.8. Automatic Microinjections

In this application, the basic microinjection task consists of: 1) injection pipette insertion into the blastocyst; 2) delivery of ES cells; and 3) pipette retraction. Consequently, the control algorithm presented in Fig. 8.14 was developed to autonomously accomplish this sequence of actions. In this case, the algorithm consisted of a high-level control structure that integrated basic control tasks according to the sequence of rules suggested in Chapter 3.

The algorithm presented in Fig. 8.14 was designed to completely automate the microinjection process. However, since ES cells delivery automation was not successful, the algorithm was modified to pause at that stage and requested user assistance.
8.3.9. Process Automation

The final automation algorithm created for the blastocyst microinjection system consisted of a high level control sequence designed to attempt full process automation. Similarly to the microinjection algorithm described above, this algorithm is basically a manager code that commands the execution of basic control tasks according to the rules presented in Chapter 3. However, in this case the goal was design an algorithm to enable the four-pipette systems to work as a production line. The resulting algorithm is presented in Fig. 8.15.
This chapter presented research and development efforts towards the achievement of a fully automated blastocyst microinjection system. Although this final goal was not completely met, progress towards achieving it was made by the design of a system more appropriate for computer-controlled operations, and by the development and implementation of automation algorithms based on rules learned from manual experimentations.

The new system configuration presented here included two extra pipettes to the standard blastocyst microinjection setup. These pipettes were used to deliver and collect...
blastocysts directly to and from the injection area, facilitating the work of automated controllers and of vision processing algorithms. The obtained simplifications arose from the fact that the four-pipette system avoided the need to search for blastocysts within the working space, and also eliminated the need for visual discrimination between injected and non-injected blastocysts by keeping them separated.

Another characteristic of the four-pipette system that proved advantageous for automation was that the collection pipette helped to keep the injection area clean from clutter that can potentially affect the vision processing system. For example, ES cells are often dropped in the injection area due to a reduction in the needle’s external pressure when it retracts from the blastocyst. This result in extra noise in the video images, causing increased chances of localization errors. Therefore, by cleaning the injection area from blastocysts and cells, the collection pipette also contributes to maintaining the vision system reliability.

The contributions of this chapter towards the design of a fully automated microinjection system included control routines for automatic delivery, capture, orientation, release and collection of blastocysts. In addition, algorithms were also developed for injection pipette motion control and for automatic blastocyst microinjections. These algorithms were all implemented and tested individually during preliminary trials with real blastocysts. The results of these trials are described in the next chapter.

A high-level control algorithm was also developed to guide fully automatic blastocysts microinjections. This algorithm integrated the lower level routines according
to the microinjection rules described in Chapter 3. However, microinjection trials under the control of this algorithm were not conducted due to the lack of time and because the task of delivering ES cells inside the blastocyst was not yet automated. Nevertheless, initial progress on that front was achieved by the design of possible algorithms to automate the ES cells delivery.
9. EXPERIMENTS WITH AUTOMATED MICROMANIPULATION ROUTINES

This chapter describes experimentation conducted with the automation algorithms developed for the blastocyst microinjection system. Here, the basic automation algorithms presented in the Chapter 8 were individually tested and tuned using real blastocysts. This was followed by trials with the high-level automation controller implemented for automatic blastocyst microinjections. However, the performed experiments consisted of only preliminary trials. These were conducted to simply assess the feasibility of the developed algorithms for automation. Much more experimentation is still needed to obtain reliability and success rate measures. Nevertheless, the experiments and the obtained preliminary results are described below.

9.1. Automatic Blastocyst Delivery Trials

Experiments with the blastocyst delivery algorithm showed that automatic control of the feeder micrometer syringe was better than simply teleoperated manual control. The reason for the improved performance was the computer’s ability to command smaller steps on the micrometer syringe. This resulted in better rotation control for the syringe and, consequently, better control of blastocysts motion inside the feeder pipette.

Slow and controlled delivery of blastocysts was achieved under computer control after system calibrations. These included adjustments to the micrometer syringe rotation speed and duration of the rotation commands. Furthermore, calibration of the waiting time
between motion commands was also performed. This was necessary because of delays in the hydraulic system response.

An example of auto-delivery of a blastocyst into the injection area is presented in Fig. 9.1. The figure shows images capture from the live video and the corresponding processed frames. The last image in the presented sequence concluded the algorithm execution since the blastocyst was positively identified in the video.

![Fig. 9.1. Example of automatic blastocyst delivery. The top row shows images captured from the live video, and the bottom row shows their corresponding processed versions. Here, the blastocyst is slowly expelled from the feeder pipette until it is detected by the vision system.](image-url)
9.2. Automatic Blastocyst Capture Trials

The algorithm developed for this task proved to be effective when a blastocyst was in or near the injection area. The creation of suction on the holding pipette acted like a strong magnet for blastocysts, guaranteeing the eventual capture of a nearby embryo without pipette or XY stage motions. Nevertheless, this task required careful calibration of the waiting time between the commanded syringe rotations. This was important to avoid creating too much suction on the holder, risking lethal damage to the embryo.

Figure 9.2 presents an example of this task being accomplished under computer control. As before, the figure presents images captured from the live video and their corresponding processed versions.

![Images of automatic blastocyst capture](image)

(A)     (B)     (C)

Fig. 9.2. Example of automatic blastocyst capture. The computer assumes a blastocyst is nearby and commands suction on the holding pipette until it is captured. The top row shows images captured from the live video, and the bottom row shows their corresponding processed versions.
9.3. Automatic Blastocyst Release and Collection Trials

The release and collection tasks, although controlled by different algorithms, were always concurrently commanded to avoid losing blastocysts, i.e., to avoid situations where the blastocyst leaves the camera’s field of view without entering the collection pipette.

The automatic execution of these tasks proved to be successful after careful calibration of the holder’s micrometer syringe. Calibration ensured the slow release of the blastocyst, thus preventing the problem of losing it as mentioned above.

A sequence of pictures obtained from one of the trials is shown in Fig. 9.3. In that case, suction was created on the collection pipette at the same time the blastocyst started to be released from the holding pipette.

![Fig. 9.3. Example of automatic blastocyst release and collection. Both tasks were commanded at the same time to avoid losing the blastocyst. The processed image (B) shows a smaller blastocyst than the one in (A) because a new version of the vision algorithm capable of finding the correct blastocyst radius was also being tested during these experiments.](image)

163
9.4. Automatic Blastocyst Orientation Trials

The adjustment of blastocyst orientation proved to be a difficult task to automate because the results of the push-pull method are not predictable. When performed manually, this method requires patience from the operator; but when performed by the computer, the requirement is variable waiting time. Nevertheless, despite being based on trial and error, the implemented algorithm was able to adjust the blastocyst orientation when necessary. This is illustrated by the pictures in Fig. 9.4.

9.5. Automatic Motion Control Trials

The evaluation experiments with the automatic control of injection pipette motions were described in Chapter 3. The results presented in that chapter showed that the developed visual servoing algorithm was successfully implemented. The algorithm provided smooth motion control and very small path-following errors. Furthermore, it was able to compensate for misalignments between the robot’s and the camera’s axes without expensive calibrations.

9.6. Automatic Microinjection Trials

The auto-injection algorithm described in Chapter 8 was evaluated through 20 preliminary blastocyst microinjections. In all of these trials the operator was responsible for holding and orienting each blastocyst prior to starting the injection process. In addition, the operator was also responsible for the delivery of ES cells once the injection
Fig. 9.4. Example of automatic blastocyst orientation: Vision processing determined that the orientation was not good (A), so the blastocyst was expelled from the holder (B), hitting the needle and rotating (C). At the same time the controller was already commanding suction on the holder (D), causing the blastocyst to accelerate towards that pipette (E), being captured at a good orientation for injection.
pipette was inside the blastocoel cavity. All operator actions were performed from the computer station.

The obtained success rate for the automatic microinjections was 75%, i.e., 15 blastocysts microinjections were considered successful by an expert in the area. This was an amazingly high success rate considering that these were the first microinjection trials conducted under computer control. Pictures from one of the trials are presented in Fig. 9.5.

After the microinjections, all 15 successfully injected blastocysts were implanted into surrogate mothers to develop to term, resulting in the birth of 8 pups. Therefore, the birth rate obtained for the preliminary automatic microinjections was 53.3%. In addition, 3 pups were later identified as chimeras, so the yield of chimeras from the implanted blastocysts was 20%. These results are summarized in Table 9.1. The table also presents results from manual blastocyst microinjections performed concurrently with the automatic microinjections. Both manual and automatic microinjections used blastocysts collected from the same female mice and ES cells from the same batch. In addition, the manual microinjections were performed by experts.
Fig. 9.5. Example of automatic blastocyst microinjection: (A) Blastocyst’s orientation and holding force were adjusted by the operator, who then started the auto-injection process; (B) The computer took over the operation and aligned the holding and injection pipettes; (C) Then the needle was brought near the blastocyst’s edge, where the piezo injector was turned on; (D) Vision processing was turned off and robot was commanded to move to the center of the blastocyst using only feedback from its motor encoders; (E) Piezo injector was turned off and the automatic control was paused. (F) Operator delivered ES cells using the joystick; (G) Automatic control restarts and extract the injection pipette from the blastocyst; (H) Vision processing is restarted and the injection pipette is taken to its home position, finalizing the injection process.
Table 9.1. Results from automatic and manual microinjections

<table>
<thead>
<tr>
<th>Operator</th>
<th>Automatic (computer controlled)</th>
<th>Manual (expert controlled)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total number of injected blastocysts</td>
<td>20</td>
<td>90</td>
</tr>
<tr>
<td># blastocysts that survived microinjection</td>
<td>15 (75%)</td>
<td>90 (100%)</td>
</tr>
<tr>
<td># blastocysts implanted</td>
<td>15</td>
<td>90</td>
</tr>
<tr>
<td># mice born</td>
<td>8 (53.3%)</td>
<td>27 (30%)</td>
</tr>
<tr>
<td># chimeras</td>
<td>3 (20%)</td>
<td>20 (22.2%)</td>
</tr>
</tbody>
</table>

An analysis of the preliminary data in Table 9.1 shows that the blastocysts injected using the automatic system were more likely to result in births than the ones injected by experts using the traditional manual method. On the other hand, the yields of chimeras from implanted blastocysts were comparable for both methods. These results can be verified from the plots in Fig. 9.6. They demonstrate that the automatic microinjection algorithm was able to safely control the penetration and extraction of the injection pipette from the blastocysts, proving that its use can potentially improve the blastocyst microinjection process.
Fig. 9.6. Birth Rate and Yield of Chimeras from Implanted Blastocysts: Comparison between automatic and tradition manual microinjections. All manual microinjections were performed by experts.

9.7. Conclusions

This chapter presented the results of preliminary experimentation conducted with the automation algorithms developed for the blastocyst microinjection system. These results demonstrated the feasibility of the basic control routines introduced in Chapter 8, including the algorithms for automatic delivery, capture, orientation, release and collection of blastocysts.

Experimentation was also conducted to evaluate the high-level control system developed for automatic blastocyst microinjections. In this case, the trials involved the participation of a human operator due to the lack of an automatic controller for ES cells delivery. Nevertheless, the microinjection task presented a 75% success rate on the first set of preliminary experiments. Moreover, the successfully injected blastocysts were
implanted to develop to term, resulting in a 53.3% birth rate and in a 20% yield of chimeras. Therefore, the developed microinjection system proved to be capable of automatically penetrating and retracting the injection pipette from the blastocyst. This accomplishment demonstrated the success of a major step towards achieving full process automation.

The automatic microinjection trials also revealed that the vision system is a cornerstone for automation success. Although it performed well during the most part of the trials, vision system errors were the major cause of failure during the automatic blastocyst microinjections. The problems observed were mainly caused by sporadic localization errors, which caused the visual servoing system to generate erroneous motion commands. Consequently, improvements are still required to create a more robust visual servoing system. These improvements can be made in two ways: 1) By modifying the vision algorithms to make them perform flawlessly on every single video frame; or 2) By creating a more robust automation controller that can identify and adapt to noise and errors in the information provided by the vision system. Based on the vision system evaluation results presented in Chapter 5 and Chapter 7, the second option is the one that will likely yield the most improvements to system performance. This indicates that research should be conducted towards the development of more intelligent controllers, possibly generated from machine learning methods such as genetic algorithms or neural networks.

In summary, the results presented in this chapter showed that automatic blastocyst microinjections can be successfully accomplished with the developed system. Experimentation proved that full automation is close to being accomplished even though
further work is still necessary. Nonetheless, areas where extra research is still necessary where identified here and in Chapter 8. Developments in those areas will enable the achievement of good efficiency rates and eliminate the need for human assistance during the microinjections. Therefore, the research presented in this chapter provided the initial steps and building blocks for the accomplishment of a fully automated blastocyst microinjection system in the near future.
10. SUGGESTIONS FOR IMPROVEMENTS AND FUTURE RESEARCH

10.1. Microinjection Equipment Upgrade

The first microinjection system improvement with potential to cause major impact on the success rate of the operations is equipment upgrade. The current setup was mostly built with surplus equipment and other donated parts no longer in use for regular blastocyst microinjections, including parts that had to be retrofitted or modified. Examples include the micrometer syringes, the manual micromanipulators, and the XY stage. Consequently, some of these components were not only outdated, but also not on their best working condition. Therefore, the acquisition of new state-of-the-art equipment such as new micrometer syringes and micromanipulators is bond to increase the performance of the microinjection system. This increased performance will come from: 1) more precise and more stable control over fluid motions in the pipettes; and 2) easier and faster system setup.

The use of a high resolution camera to acquired images from the microscope may also bring improvements to the system performance by allowing better localizations. However, this should be investigated carefully because in the current vision system higher resolution translates in higher processing time.

Related to image quality, a definitely necessary improvement for completing system automation is the incorporation of motorized focus adjustment to the microscope. This will benefit the microinjection system by allowing the completion of tasks at different
focal levels, i.e., ES cells collection and the blastocysts microinjections, without operator visits to the microscope stage. Furthermore, motorized focus control may benefit the vision system by allowing automatic control of image sharpness, which is important for correct object localizations.

A final suggestion for microinjection equipment upgrade is the acquisition of an anti-vibration table for the system. The developed setup did not include such equipment, thus it was susceptible to mechanical vibrations in the environment, which transferred to the microinjection pipettes causing random oscillations. Typical vibrations levels in the system caused frequent pipette oscillations in the order of 1µm. However, when machines were in operation nearby, the pipette vibrations were noticeably amplified. Therefore, an anti-vibration table will be useful for reducing oscillations and for maintaining a consistent environment for blastocyst microinjections.

10.2. Vision System Improvements

The current vision system still needs improvement in a few areas, including: processing speed; blastocyst detection; automatic selection of target point during microinjection; and ES cells detection. The processing speed, which is currently only 10 frames per second, may be improved by synchronizing the frame acquisitions and processing tasks. Alternatively, faster algorithms may investigated for the image pre-processing tasks, which comprise RGB to gray-scale conversion, noise filtering, gradient computations, edge detections, and image segmentation.
The detection of blastocysts needs improvement because, contrary to our original belief, their sizes and shapes are not always consistent. Differences are observed because blastocysts collected for injection are not always at the same developmental stage. More developed blastocysts are a little larger, with more expanded cavities and clearly defined ICM. In contrast, less developed blastocysts tend to be smaller and present smaller cavities and thicker trophoblast. In addition, some blastocysts present an outline that is more oval than circular. These differences are exemplified by the pictures in Fig. 10.1. As a consequence of the differences, accurate blastocyst detection based on the described 2D Hough transform required constant manual adjustments of the radius parameter. This situation was certainly not ideal, so the Hough algorithm was modified to also estimate the blastocyst’s radius. However, this modification resulted in a large increase in processing time. Therefore, different algorithms for fast and accurate blastocyst localizations should be investigated.

The automatic selection of microinjection target point also needs improvement due to the variety of blastocyst’s sizes and shapes mentioned above. In this case, microinjection efficiency may be increased by an algorithm able to find the optimal location to stop the needle. Such algorithm may prevent moving the injection pipette too

![Fig. 10.1. Examples of the variability in blastocysts’ sizes and shapes.](image-url)
far into small blastocyst, or not far enough into large ones. Furthermore, finding the optimal microinjection target will enable better pipette alignment prior to injection, which can increase the system performance in situations where the ICM is not situated directly across from the needle insertion point, as the case in Fig. 10.2 shows.

Finally, a vision algorithm for detecting and counting ES cells is needed for the completion of the automated blastocyst microinjection system. This need was discussed in details in Chapter 8, where possible algorithms were proposed. In summary, this function is needed to enable automatically delivery of the appropriate number of ES cells into the blastocysts during the microinjection process.

10.3. Fully Automated Blastocyst Microinjections

After the automating ES cells delivery and possibly improving some of the automation routines, the microinjection system will be ready to perform fully automated blastocyst microinjections. Trials with the completed system should be performed then. These trails are likely to provide insight into further system improvements and help define new research areas related to this biomanipulation area.
10.4. Intelligent System Control

As mentioned in Chapter 9, the current microinjection system has shown the need for a more robust autonomous controller, especially one that can identify and adapt to errors in the information provided by the vision system. This can possibly be accomplished by the development of a more complex rule-based controller; however, the use of artificial intelligence is likely to yield better results here because of the natural inconsistencies present in the task, i.e., differences in shape, size, and rigidity of blastocysts, differences in pipette shapes, and differences in image focus from one day to the next. Therefore, research into the use of machine learning techniques for automatic blastocyst microinjections should be conducted in the future.

Possible A.I. methods that may be applicable for automating the developed microinjection system include learn-from-example methods such as self-organizing maps and other types of neural networks. These methods are attractive because they may benefit from data on human-controlled microinjections gathered from teleoperated trails with the current system, as described in Chapter 3.

Intelligent controllers may also be used for the automation of the basic tasks described in Chapter 8. For example, it would be interesting to investigate if the robotic system can learn to adjust the blastocysts’ orientation for injection, and also if this could results in efficiency gains over rule-based methods.
10.5. System Configuration Improvements

Experimentation with the current system showed that it is cumbersome and difficult to setup. In addition, it is based on a very expensive binocular microscope which may not be needed for teleoperated or automatic biomanipulations. Therefore, future simplifications and miniaturizations are desired to: 1) facilitate the system setup process; 2) reduce equipment costs; and 3) create a more flexible system that can be used for different biomanipulation tasks and robotic research.

Based on the goals described above, a possible future system configuration is shown in Fig. 10.3. The proposed configuration is based on four micromanipulation robots, an XY stage, a petri dish cooling stage, and four motorized micrometer syringes. Furthermore, the setup proposes to replace the inverted microscope with a single objective and a motorized set of precision zoom lenses that can be directly attached to the video camera. The necessary light level may be provided by a fiber optic illuminator, and all system components should be placed on the top of an anti-vibration table.
10.6. Related Areas for Future Research

There are several related research areas that can benefit from the designed microinjection system. These include machine learning and artificial intelligence research applied to biorobotic systems, vision systems, and behavior learning. In addition, research
in human-machine interface and research in teleoperation can also be accomplished based on the developed system.

The addition of extra sensors and capabilities to the micromanipulation system can also make it usable for research in many other areas. For example, if a cellular force sensor is added to the system, it can be used for research in augmented reality systems for biomanipulations; or for research into the mechanical characterization of biological cells and embryos. The design of such force sensor is in itself a major research area nowadays.

Teleoperation based on two robots and two joysticks is another interesting problem that can be investigated using the developed test bed. In traditional biomanipulations the operator uses both hands for the operations. Therefore, it is logical to ask if the use of two joysticks can improve micromanipulation performance.

Finally, a lot of research is yet to be done on the miniaturization of blastocyst microinjection systems. In this case, research areas include electro-rotation for blastocyst orientation; microfluidics for the efficient transport of cells and embryos; and MEMS research for the creation of micro-needles and injection systems. Research is also needed on the integration of these areas into a lab-on-a-chip type of system for blastocyst microinjections.
11. **Conclusions**

A novel biomanipulation system for blastocysts microinjections was designed, implemented and evaluated during this research. The goal of increasing the consistency and efficiency rates of the microinjections was pursued through automation, resulting in the creation of a new system design more appropriate for computer controlled operations and intelligent (machine learned) controllers. This was achieved through the development of a teleoperated system that treated embryo microinjections as a computer game. The created microinjection system was entirely operated from a computer station through a joystick or automated controllers, and included real-time video processing for the acquisition of experimental data and control.

Experimentation with the developed microinjection system under the control of both expert and amateur operators proved that it was effective, easy to use, and capable of eliminating the need for extensive training of microinjection operators. The obtained results demonstrated that all operators obtained a microinjection success rate over 80%, confirming that the developed system enabled efficiency improvements over tradition manual microinjections. Furthermore, blastocysts injected in this new system were more likely to develop to term and to yield chimeras than the ones manually injected by experts.

The analysis of the teleoperated experiments produced new knowledge about the process and allowed the identification of major problems that can occur during blastocyst microinjections. Within those problems, the most frequent one was the inability to get the injection pipette inside the blastocoel cavity. This was found to occur due to numerous
reasons, including inappropriate holding force, incorrect injection pipette alignment, and the rigidity of the blastocysts. Possible solutions for minimizing this and other common problems were identified in this work, resulting in the creation of a set of rules to guide teleoperated and automatic microinjections.

Efforts toward full system automation were also part of this research. These included the design of new system configuration optimized for computer controlled microinjections and contributions to the vision processing area. In the latter case, speed-up methods that enable real-time template matching in video were implemented and evaluated. In addition, new algorithms were created to identify and analyze the image of blastocysts.

Further research contributions towards system automation consisted of the identification of basic process tasks and the development of control routines for their automation. These included algorithms for injection pipette motion control and for automatic delivery, capture, orientation, release and collection of blastocysts. In addition, high-level control routines were designed for automatic blastocyst microinjections and for full process automation based on the rules learned from the teleoperated trials.

Preliminary experimentation demonstrated the feasibility of each automated task and resulted in a 75% success rate for the microinjection phase alone. This showed that the developed system was capable of automatically penetrating and retracting the injection pipette from the blastocyst, which was a major step towards achieving full process automation.
In summary, this research proved that the application of robotics and automation methods to embryo biomanipulations can increase the efficiency of blastocyst microinjections. Furthermore, it demonstrated that fully automated microinjections are possible and close to being successfully accomplished. The initial steps and building blocks for future developments in that direction were provided here, including the identification of areas where further research is still needed. Developments in those areas will enable full process automation, at which point research will determine if automation can also increase the consistency of the blastocyst microinjections.
12. References


APPENDICES
APPENDIX 1. THE BLASTOWORKS SOFTWARE

This chapter describes the graphical user interface and the underlying control modules developed for the blastocyst microinjection system.

A1.1. Software Structure Overview

A general overview of the developed software is presented in Fig. A1.1. The figure shows that the main user interface module is responsible for the configuration and management of all other software modules.

![Fig. A1.1. Software overview.](image)

A1.2. The User Interface Module

This module is composed of the graphical user interface and all underlying code necessary to interact with an operator. Therefore, this module contains handler routines for all primary events generated by the user. In addition, this module is also responsible for managing the follow of information between the other software modules, so it contains code for communication events.
A1.3. Image Management

The image management system is responsible for the acquisition, adjustments and processing of images. For convenience, image acquisitions and adjustments are performed directly within the User Interface Module; however, processing is performed by the Image Processor Object. This object is responsible for the blastocyst and pipettes localizations. An overview of the image management system is displayed in Fig. A1.2.

A1.3.1. Image Sources

The developed software accepts four types of image sources: Live video, simulated video, pre-recorded video files, and picture files. These options were implemented to facilitate the development and testing of the video processing algorithms. During normal system operation the image source should be set to Live Video. The image source is selected from the Source menu, which is shown in Fig. A1.3.
A1.3.1.1. Live Video

When the Live Video source is selected, the images used for processing are acquired from one of the video devices attached to the processing computer. The appropriate device should be selected from the menu AV Devices. If a video grabber card is selected as the input video device, it may be necessary to also select the proper video source, as shown in Fig. A1.4. Note that only 320x240 video frames are accepted by the developed software at this time, so the appropriate video frame size should be selected.
The video acquisition code implemented here was based on the “DirectX.Capture” code created by Brian Low. This code was downloaded from the online community The Code Project (www.codeproject.com).

A1.3.1.2. Simulated Video

The selection of Simulated Video as the image source starts the video simulation process, which creates video frames containing a blastocyst and an injection pipette. The simulated blastocyst image was described in Chapter 5. The simulated pipette consists of a rectangle that extends from the right image edge to the simulated pipette tip location. An example of a simulated video frame is presented in Fig. A1.5.
Once in video simulation mode, the user can choose the simulation type to one of the following: 1) joystick control; 2) random motions; or 3) auto-experiments. These options are shown in Fig. A1.6.

The first video simulation type uses joystick inputs to control the simulated pipette motions. The second type ignores joystick inputs and creates random motions for both the
blastocyst and the pipette. The third type, on the other hand, is strictly experimental. It was used to generate training data for machine learning investigations. In this case, random blastocyst sizes and random locations are initially picked for both objects. Then, the pipette is moved in a straight line to a random destination, and the initial and final states of the objects are recorded in a data file.

Simulation parameters can be configured from the Simulation Options menu, which is shown in Fig. A1.7. The configuration options include the selection of objects for simulation and the selection of the percentage of object pixels to be transferred to the final video frames. The latter is used to create noisy simulated images as the ones described in Chapter 5.

![Fig. A1.7. Video simulation options.](image)

The Simulation Options menu also provides access to motion parameters used by the Random Motions simulation type. This is done through the Object Motions menu item. Clicking on this item opens the tool window shown in Fig. A1.8. The figure shows that
the simulated motions can be completely random between consecutive frames, or follow a continuous random path.

![Simulated Video Motion Options](image)

**Fig. A1.8.** Simulated motions configuration tool.

### A1.3.1.3. Recorded Video

Video files can also be used as the source of images for processing. In this case, the option *Recorded Video* should be selected on the *Source* menu. This selection asks for the selection of a video file and then opens the *Video Controller* tool window, which is shown in Fig. A1.9. This controller lets the user play, pause and stop the movie, and also select video frames for processing. This is done through the *Process Frame* button.

![Video Controller](image)

**Fig. A1.9.** Video controller tool window.
A1.3.1.4. **Pictures**

The use of picture files as image source is possible through the selection of the option *Pictures* on the *Source* menu. In this case, the user can select specific picture files for processing or an entire pictures folder.

A1.3.2. **Image Adjustments**

Image adjustments are currently implemented only for the live video images. The controls for adjustments are accessed through the *Image* menu, as demonstrated in Fig. A1.10. The figure also shows the Image Adjustment tool window, which is used to control the image brightness, contrast and gamma parameters.

![Image of Image Adjustment window](image.png)

*Fig. A1.10. Live video adjustments.*
A1.3.2.1. **Image Noise**

The input images can be corrupted by custom levels of Gaussian noise, impulse noise, or blurring. These are set through the Noise menu, as shown in Fig. A1.11. Examples of corrupted images can be found in Chapter 5.

![Fig. A1.11. Image corruption by custom noise levels.](image)

A1.3.3. **Image Processing**

The image processing setup parameters can be accessed from the *Image Processing* menu, as demonstrated by Fig. A1.12. This menu is used to configure the image preprocessing routines and the localization parameters for the objects of interest. The *Templates* menu item provides access to the injection and holding pipettes options, including the acquisition of templates directly from live video. It also allows the selection of the template matching type and speed-up techniques. The *Blastocyst Properties* menu item is used to configure the radius and the number of sectors used during blastocyst
localizations and analyses. Finally, the *Process Selection* menu item provides control over the objects’ localizations and displayed outputs.

Fig. A1.12. Image processing menus.
A1.3.4. Image Flow

The flow of images within the developed software follows the diagram in Fig. A1.13. The diagram shows that all image sources deliver their products to a centralized image acquisition process, which forward them into the processing pipeline. The first processing step consists of distorting the original images with custom levels of noise. After that, the images are displayed in the user interface and forwarded to the processing routines. Once the processing is completed, the processed images are displayed and saved.

A1.4. Motion Management

The motion management system is responsible for receiving and processing motion commands generated by the user or by automatic control routines. Here, a main processing unit performs commands conditioning and generates appropriate outputs to interface modules. These modules, in turn, communicate directly with external device drivers,
submitting the motion commands and receiving feedback data. An overview of the motion management system is displayed in Fig. A1.14.

![Fig. A1.14. Overview of the motion management system.](image)

### A1.4.1. Joystick Interface

The main purpose of this interface is to capture joystick events and transmit them to the processing unit. However, the interface also allows for custom configuration of the joystick inputs, including the definition of command gains, dead-zone offsets, and command values associated to the digital inputs. The interface is presented in Fig. A1.15.

![Fig. A1.15. Joystick interface.](image)
A1.4.2. Robot Interface

This interface is used to control robot motions through continuous communications with the Siskiyou robot driver. Here, the velocity, the acceleration and the zero position of each robot axis can be set individually. Furthermore, the interface can be used to directly command relative or absolute axes motions, including commands to move them back to their zero positions.

The robot interface is also responsible for constantly obtaining position feedback from the robot’s encoders, which are made immediately available to other processes such as automatic motion control. The robot interface is presented in Fig. A1.16.
A1.4.3.  Syringes and Piezo Interface

This interface was designed to communicate and control the *Interface and Control Board* (ICB), which is responsible for driving the micrometer syringes and for controlling the activation of the piezo injector. Here, the speed of each micrometer syringe motor can be adjusted through slide bars, and motor rotations can be directly commanded through push-buttons. The ICB digital outputs can also be controlled directly from this interface, including Output 1, which is used to control the activation of the piezo injector. The developed ICB interface is presented in Fig. A1.17 and details of the control board are presented in Appendix 2.

![ICB Interface Diagram](image)

Fig. A1.17. ICB interface.
A1.4.4. XY Stage Interface

The XY stage interface is presented in Fig. A1.18. It is used to communicate with the *ICB Stepper* driver board and to control stage motions. Similarly to the robot interface, each stage axis can be independently configured through this interface. Here, the user can set the axes velocity, acceleration, and zero positions. In addition, absolute and relative motion commands can be directly realized through push-buttons, including the homing of the stage axes. An extra feature of this interface consists of a memory area able to record up to 20 XY stage coordinates. These are used to enable automatic stage motions to the marked positions, which are initiated by software commands or by clicking the “Goto Mark” push-button.

![XY Stage Interface](image.png)

*Fig. A1.18. XY stage interface.*
A1.5. File Management

The file management system is responsible for recording experimental data during microinjection trials. This system is activated by the selection of an experiment name through the *File* menu. Once active, two data files are created to record motion commands and objects coordinates. Every time an action is commanded by the user or by an automatic routine, the file management system records it in the motion commands data file along with a time stamp. Likewise, when object motions are detected by the vision system, their locations are recorded and time-stamped in the objects coordinates data file. In addition, the serial numbers associated to each processed image are also recorded in the data files to allow off-line evaluations of the image processing system.
APPENDIX 2. INTERFACE AND CONTROL BOARD

A2.1. Introduction

The Interface and Control Board (ICB) was designed to handle the low-level motion control and output functions. This board is able to control four digital outputs and up to six DC motors in open or closed-loop (with encoder feedback). It receives commands from the master computer through serial communications and drives the motors and outputs based on local computations. Both RS232 and USB are supported by this board.

The ICB also provides a direct-drive function which enables the master computer to directly drive the motors and outputs. This is performed through two standard parallel port connections, which are optically isolated from the local control circuitry. In this mode, the master computer controls the motors and outputs by flipping bits on its parallel ports. Closed-loop control is not possible in this case, but the speed of the motors can be adjusted through serial communications with the ICB.

The ICB is based on the microcontroller ATMEGA128 from Atmel, which is an 8-bit RISC processor with 128K bytes of programmable flash memory. This microcontroller unit is responsible for all computations and I/O management on the ICB.
A summary of the ICB specifications is presented in Table A2.1 below.

Table A2.1. ICB specifications

<table>
<thead>
<tr>
<th>Specification</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Motor Outputs</strong></td>
<td>6 DC motors</td>
</tr>
<tr>
<td><strong>Max motor current</strong></td>
<td>2A (DC operation)</td>
</tr>
<tr>
<td><strong>Digital Outputs</strong></td>
<td>4</td>
</tr>
<tr>
<td><strong>Encoder Inputs</strong></td>
<td>6</td>
</tr>
<tr>
<td><strong>Analog Input or Digital I/O</strong></td>
<td>8 (all featuring 12-bit ADC capability)</td>
</tr>
<tr>
<td><strong>Analog reference adjustment</strong></td>
<td>1 multi-turn potentiometer on board</td>
</tr>
<tr>
<td><strong>Communications</strong></td>
<td>RS232, USB, Parallel</td>
</tr>
<tr>
<td><strong>Supply input</strong></td>
<td>Logic circuit: 7V – 26V DC, 500mA</td>
</tr>
<tr>
<td></td>
<td>Motors: 5V – 46V DC, 2A</td>
</tr>
</tbody>
</table>
A2.3. ICB Connectors

A description of the ICB connectors is presented in Table A2.2 below. The connector locations are presented in the Figures A2.1 and A2.2.

<table>
<thead>
<tr>
<th>Connector name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM</td>
<td>Motor power supply: 5V – 46V DC, 2A</td>
</tr>
<tr>
<td>VIN</td>
<td>Logic power supply: 7V – 26V DC, 500mA</td>
</tr>
<tr>
<td>RS232</td>
<td>Serial port (115.2Kbps, 8 data bits, 2 stop bits, no parity)</td>
</tr>
<tr>
<td>USB</td>
<td>USB port</td>
</tr>
<tr>
<td>PWR</td>
<td>+5Vdc output</td>
</tr>
<tr>
<td>A_REF</td>
<td>Test point for the analog reference voltage</td>
</tr>
<tr>
<td>PORTF</td>
<td>Digital I/O and ADC inputs</td>
</tr>
<tr>
<td>OUT</td>
<td>Digital outputs</td>
</tr>
<tr>
<td>MOTOR1 – MOTOR6</td>
<td>Motor outputs</td>
</tr>
<tr>
<td>PP1, PP2</td>
<td>Parallel port inputs</td>
</tr>
<tr>
<td>ENC1 – ENC6</td>
<td>Encoder ports</td>
</tr>
</tbody>
</table>
Fig. A2.1. Top view of the ICB showing connector locations.

Fig. A2.2. Bottom view of the ICB.
A2.4. ICB Commands

Table A2.3 presents the set of serial commands accepted by the ICB. These commands are used to setup and control 6 DC motors and 4 digital outputs. The parameters in brackets represent 8-bit numbers, i.e., bytes with value between 0 and 255.

<table>
<thead>
<tr>
<th>Command format</th>
<th>Return value</th>
<th>Example</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>? (string)</td>
<td>?</td>
<td>ICB_V1.1</td>
<td>Query: Returns the identification string</td>
</tr>
<tr>
<td>M[m]P[p]</td>
<td>P</td>
<td>M[1]P[100]</td>
<td>Set the PWM value for the motor m to p</td>
</tr>
<tr>
<td>M[m]D[d]</td>
<td>D</td>
<td>M[1]D[0]</td>
<td>Set the direction of rotation for the motor m to d</td>
</tr>
<tr>
<td>M[m]? P[p]D[d]</td>
<td>M[1]? P[100]D[0]</td>
<td>Query motor m: Returns its PWM value and direction of rotation</td>
<td></td>
</tr>
<tr>
<td>S</td>
<td>S</td>
<td>S</td>
<td>Stop all motors</td>
</tr>
<tr>
<td>O[b0][b1]</td>
<td>O</td>
<td>O[255][1]</td>
<td>Set outputs</td>
</tr>
</tbody>
</table>

A2.5. ICB STEPPER Commands

The developed Interface and Control Board can also be used to drive stepper motors. This is enabled by a simple change on the ICB’s firmware. Here, the commands implemented for the control of two bipolar stepper motors are described. Table A2.4 presents the complete set those commands. Again, the parameters in brackets represent 8-bit numbers.
<table>
<thead>
<tr>
<th>Command format</th>
<th>Return value</th>
<th>Description</th>
<th>Command details</th>
</tr>
</thead>
<tbody>
<tr>
<td>?</td>
<td>(string)</td>
<td>Query: Returns the identification string</td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>E</td>
<td>E-Stop: Stop motors</td>
<td></td>
</tr>
<tr>
<td>M[m]</td>
<td>M</td>
<td>Set step mode</td>
<td>(m = 0) full step (m = 1) half step</td>
</tr>
<tr>
<td>S[m][d][b1][b0]</td>
<td>S</td>
<td>Command ((255*b1+b0)) steps on motor (m) steps in the direction (d)</td>
<td>(b1 =) MSB (b0 =) LSB</td>
</tr>
<tr>
<td>A[m][s]</td>
<td>A</td>
<td>Set motor (m) speed to (s)</td>
<td>(0 \leq s \leq 255)</td>
</tr>
<tr>
<td>B[b]</td>
<td>B</td>
<td>Set the speed mode</td>
<td>(b = 0) constant speed (b = 1) ramp (acc/dec)</td>
</tr>
<tr>
<td>T[m][t]</td>
<td>T</td>
<td>Set ramp time</td>
<td>(0 \leq t \leq 255)</td>
</tr>
<tr>
<td>C[c]</td>
<td>C</td>
<td>Command byte. Bits 0 to 3 are used to directly control the motors</td>
<td>(c0 =) motor 1 direction (c1 =) motor 1 on/off (c2 =) motor 2 direction (c3 =) motor 2 on/off</td>
</tr>
<tr>
<td>D[m][p1][p0]</td>
<td>D</td>
<td>Send motor (m) to the position ((255*p1+p0))</td>
<td>(p1 =) MSB (p0 =) LSB</td>
</tr>
<tr>
<td>j[m]</td>
<td>j</td>
<td>Send motor (m) to the zero position</td>
<td></td>
</tr>
<tr>
<td>k[m]</td>
<td>k</td>
<td>Send motor (m) to the mark position</td>
<td></td>
</tr>
<tr>
<td>J[m]</td>
<td>J</td>
<td>Home motor (m)</td>
<td>Move towards the negative direction until hitting the limit switch</td>
</tr>
<tr>
<td>K[m]</td>
<td>K</td>
<td>Mark motor (m)</td>
<td>Move towards the positive direction until hitting the limit switch</td>
</tr>
<tr>
<td>Q[m]</td>
<td>Q</td>
<td>Set motor (m) current position as zero</td>
<td></td>
</tr>
<tr>
<td>R[m]</td>
<td>R</td>
<td>Set motor (m) current position as mark</td>
<td></td>
</tr>
<tr>
<td>GA[m]</td>
<td>[s]</td>
<td>Get speed of motor (m)</td>
<td></td>
</tr>
<tr>
<td>GB</td>
<td>[b]</td>
<td>Get speed mode</td>
<td></td>
</tr>
<tr>
<td>GT[m]</td>
<td>[s]</td>
<td>Get speed ramp time of motor (m)</td>
<td></td>
</tr>
<tr>
<td>GL</td>
<td>[s]</td>
<td>Get limit switch status</td>
<td>(s0 =) motor 1 zero (s1 =) motor 1 end (s2 =) motor 2 zero (s3 =) motor 2 end</td>
</tr>
<tr>
<td>GM</td>
<td>[m]</td>
<td>Get step mode</td>
<td></td>
</tr>
<tr>
<td>GP</td>
<td>[p0]...[p3]</td>
<td>Get motor positions</td>
<td>(M1 \text{ pos} = (255<em>p0+p1)) (M2 \text{ pos} = (255</em>p2+p3))</td>
</tr>
<tr>
<td>GR</td>
<td>[p0]...[p3]</td>
<td>Get motor mark positions</td>
<td>(M1 \text{ mark} = (255<em>p0+p1)) (M2 \text{ mark} = (255</em>p2+p3))</td>
</tr>
</tbody>
</table>
A2.6. ICB Circuit Diagrams

The ICB was designed to be a multi-purpose motor driver and I/O board. Therefore, it includes hardware for the most common functions in these types of applications: Powerful H-bridges, encoder inputs with Schmitt-trigger, analog inputs with adjustable reference level, and general purpose I/O pins. In addition, the ICB also features hardware for USB and RS232 communications with its master controller. The following diagrams present all of the circuits implemented on the ICB.
Fig. A2.3. Circuit diagram for the ICB (part 1 of 4).
Fig. A2.4. Circuit diagram for the ICB (part 2 of 4).
Fig. A2.5. Circuit diagram for the ICB (part 3 of 4).
Fig. A2.6. Circuit diagram for the ICB (part 4 of 4).
APPENDIX 3. SEMI-AUTOMATED BLASTOCYST MICROINJECTION

L. Mattos, E. Grant, and R. Thresher, “Semi-automated blastocyst microinjection,”
Proceedings of the 2006 IEEE International Conference on Robotics and Automation,
Orlando, USA, May 2006.
Semi-Automated Blastocyst Microinjection

Leonardo Mattos and Edward Grant
Center for Robotics and Intelligent Machines
North Carolina State University
Raleigh, NC 27695-7911 USA
{lemattos, egrant}@ncsu.edu

Abstract - The focus of this paper is the design and development of a semi-automated system for microinjection of embryonic stem cells into blastocysts. Semi-automation is achieved through treating cell microinjection as a computer game. In this first phase, cell manipulation and microinjection is carried out using a joystick and an interactive graphical user interface (GUI). For this system to be developed further, to achieve full automation, this first phase had to succeed. The interactive GUI records and displays real-time video images of the cells; processed images of cells; and the microinjection strategies adopted by operators during cell manipulation through a real-time video processing algorithm. Experiments showed that this first phase of the research was successful. Future research will develop knowledge-based controllers using machine-learning techniques as the research drives towards its goal of fully automated cell microinjection.

Index Terms – Cell injection, biomanipulation, teleoperation, micro robotic system.

I. INTRODUCTION

In recent years, biomedical and environmental health research has relied on the use of genetically modified mice (both transgenic and gene-targeted) as unique tools for the discovery and development of new treatments and cures for many serious diseases [1]. These mice also offer new ways of investigating the interactions between individual genes and the environments that influence the development of a wide range of human afflictions including diabetes, arteriosclerosis, hypertension, Alzheimer’s disease, and cancer [2].

Gene-targeted mice are commonly created through blastocyst microinjection, i.e., the injection of genetically altered embryonic stem (ES) cells into blastocysts. Cell microinjection is a very delicate process, one that requires meticulous manipulation of the cells. Operators spend hours at microscopes manually carrying out blastocyst microinjection procedures. Still, the results are often less than satisfactory. According to published literature [3], operators need up to one full year of training to become proficient at this task. Even then, low microinjection success rates are encountered, often being as low as 20%

Because current blastocyst microinjection procedures are subjected to the inherent inconsistencies and lack of reproducibility of human actions, automating the process will increase the overall efficiency of bio-manipulations. Here, a semi-automated system, which is teleoperated through a graphical user interface running on a desktop computer, has been developed to improve microinjection. This interface was designed as a video-based teleoperation, receiving commands from a joystick and visual feedback from video camera mounted on the microscope. The interface also records all microinjection operations, through a real-time video processing algorithm based on cross-correlation template matching and Hough transforms.

The developed system is similar to the cell injection system created by Sun and Nelson [4], and has features related to the human-in-the-loop biomanipulation developed by Kim et al. [3] and Arau et al. [5]. Those systems were considered as models for the biomanipulation system presented here, although the target cells and research goals are different in this case. Our eventual goal is to use techniques from artificial intelligence to learn how to perform blastocyst microinjections by extracting human knowledge of the process. Data recorded and analyzed from the current set-up will be used to generate a knowledge-based controller, regarded here as the first step towards a fully automated microinjection system. This controller will be used as a base-line against which other intelligent and machine-learned controllers will be tested. The goal is to produce intelligent controllers capable of automating the process and provide greater consistency and higher efficiency rates for the microinjection process.

First, this paper starts with an overview of the developed semi-automated microinjection system. The equipment used, the system configuration, and the software algorithms for cell microinjection, monitoring, and control are discussed. Second, a description of the graphical user interface is given, highlighting its pertinent features, capabilities, and its means of collecting experimental data. Lastly, some initial experimental data obtained from the system are discussed.

II. THE SEMI-AUTOMATED MICROINJECTION SYSTEM

A. Main Components

The hardware for the semi-automated blastocyst microinjection system described in this paper was developed using standard microinjection equipment for manual microinjection. A major hardware component of the cell manipulation system is a Zeiss INFS5 inverted microscope, which provides the necessary optical magnification and illumination levels for imaging the injection area. Images of that area are captured by a black-and-white CCD camera (LCL-902K, Watco America Corp.) installed on the microscope, and are sent to a host desktop computer as video signals, see Fig. 1.
The other major components of the system are: Two Leitz manual micromanipulators for positioning the holding and injection pipettes, one computer controlled LN piezo injector, one NewFocus 8082 kinematic stage and its respective driver for high-resolution motion control of the injection needle, and two Narishige IM-6 micrometer syringes outfitted with motors for computer control of fluid motions, see Fig. 1.

A Dell Dimension 4600 with a Pentium4 CPU running at 3.2GHz is employed for all cell teleoperation, processing, and control. The computer uses a WinTV-PCI frame grabber for capturing video images, and a Saitek Cyborg Evo Force joystick as the control input for positioning and orienting the cells in the Petri dish.

B. System Configuration

The microinjection system configuration is shown in block diagram form in Fig. 2. The figure shows that the user controls the microinjection operation from the computer station, where he/she interacts with the system through a monitor and a joystick. Four devices are independently controlled from the joystick: (1) a high-resolution kinematic stage that accurately positions the injection pipette; (2) a piezo-injector used for blastocyst penetration; (3) a micrometer syringe that produces a suction force at the holding pipette; and (4) a micrometer syringe that controls ES cells dispensing into a blastocyst.

From Fig. 2, two devices are used to accurately control the position of the injection pipette: (1) the manual micromanipulator; and (2) the computer controlled kinematic stage. The manual micromanipulator provides coarse positioning, i.e., large displacements, while the multi-axis kinematic stage provides fine motion control, i.e., fine displacements. This kinematic stage is controlled using serial commands and provides five degrees of freedom with incremental motions of less than 30 nm per axis.

The holding pipette can be positioned manually, by adjusting one of the Leitz micromanipulators; thereby it is maintained at a fixed position throughout the whole teleoperation procedure. This design was adopted because blastocyst manipulation can be accomplished using a single moving pipette, in this case the injection pipette. The user can rotate and translate the blastocyst into the injection zone using the injection pipette alone. Once the blastocyst is near the holding pipette suction is turned on, fixing the blastocyst for the duration of the injection cycle. ES cells are injected into the blastocyst after the injection pipette is “fixed” into it.

C. Software Units

Figure 2 also shows that the processing software is composed of three main units: (1) the video processing unit, (2) the motion control unit, and (3) the data recording unit for the experiments.

The video processing software unit is responsible for the acquisition, processing and displaying of video frames. Video frames are acquired from the frame grabber card, and post-processed to locate the pipettes and any blastocysts. This software unit also estimates the optimum injection point on the blastocyst’s trophoblasts (the thinnest point away from the inner cell mass), and it displays the original and processed video frames on the computer monitor. A description of the image processing algorithm used within this module will be given in a later section.

The motion control software unit receives and manages motion commands from the joystick. Using these inputs, the unit does all the necessary processing and signal generation to drive the motion devices via the computer’s serial and parallel ports.

The data recording unit is an observer software that records all the events of an experiment and writes these into a data file. This unit keeps track of all the inputs and outputs of the micromanipulator system. It also records the positions of the injection and holding pipettes, the blastocyst position, and a continually updated estimate of the optimum blastocyst injection point. The goal of data collection is to create a database to be used in the next stage of this research, where a
knowledge-based controller will be developed to automate the blastocyst microinjection process. Such knowledge-based controller should perform three functions: (1) it can be encoded to see how it performs as an automated controller, (2) it can be used as a standard controller, against which other machine-learned or intelligent controllers can be tested, and (3) it can be used for the derivation of performance metrics.

III. THE Interactive User Interface

Because the blastocyst microinjection system is operated from a remote computer, an interactive graphical user interface (GUI) was developed to facilitate interaction between the operator and the blastocyst’s micro-world. The GUI allows the user easy control over the operation, and also converts a tedious and tiring manual microinjection task into a task with the feel of a computer game. The interactive GUI for microinjection puts the operator in front of a computer monitor instead of on the microscope; this is a: (1) more comfortable operating environment, and (2) a training tool for teaching new personnel cell microinjection.

The developed interface is also used to evaluate the video processing software and to collect experimental data for a knowledge database. This is achieved with the display of both the original video (from the microscope’s CCD camera) and the processed video frames simultaneously, providing the user with the means to visually inspect the functioning of the video processing algorithm instantaneously. If processing yields good tracking results throughout the microinjection process, the user can transfer and save this experimental data into the knowledge database. This feature allows reliable experimental data to be stored for use during the next phase of this research, which will be the development of a knowledge-based controller for fully automating the blastocyst microinjections.

To make the teleoperated microinjection process similar to a computer game, a joystick was selected as the input device for all motion commands. Using the joystick, and visual feedback from the GUI, the user can: (1) control up to five degrees of freedom of the injection pipette; (2) control both the holding and injection micrometer syringes; and (3) activate the piezo-injector device. These functions were assigned to the joystick and its buttons as shown in Fig. 3.

The GUI can be also used to adjust motion parameters: (1) the joystick sensitivities; (2) the frequency of the steps of each axis of the high-resolution kinematic stage; and (3) the conversion rate between joystick inputs and the number of steps to be commanded.

Another main feature of the developed graphical interface is the control given over the video processing software unit. Using the GUI the user can select several parameters that are used for finding and tracking the motion of the blastocyst and the pipettes. These parameters include the approximate blastocyst size; the resolution of the search for the best injection point on the trophoblast; and the templates for the injection and holding pipettes, which are used for tracking via a matching algorithm. The next section will describe these parameters in more details.

A screenshot of the GUI is shown in Fig. 4. This screenshot shows that the GUI has two video panels: (1) original video frames are displayed in the left panel, and (2)
processed video frames are displayed in the right panel. Figure 4 also displays how the tracking algorithm superimposes object templates onto the processed video frames, giving instantaneous visual assessment of the video processing algorithm. The estimated coordinates for the blastocyst, injection pipette, and holding pipette are also displayed within the interface.

IV. VIDEO PROCESSING ALGORITHMS

The video processing algorithm developed for the blastocyst microinjection system is responsible for the detection and tracking of three objects within the video frames: (1) the blastocyst, (2) the injection pipette, and (3) the holding pipette. The algorithm detects and tracks the blastocyst using Hough transforms; and searches for the injection and holding pipettes using cross-correlation template matching.

This video processing algorithm was developed with the next phase of this research in mind, i.e., the creation of a fully automated microinjection system. Thus, the choice of a robust algorithm capable of providing accurate position information about those three objects was a major consideration. Furthermore, since our intention is also to develop a knowledge-based controller for the full automation, it was crucial to choose an algorithm that would provide the reliable experiment data needed to build the knowledge database from which the controller will learn.

An overview of the implemented algorithm is presented in Fig. 5. The initial phase of the algorithm consists of a series of basic steps which are common to the detection of all three features of interest. It starts with the acquisition of a frame image from the frame grabber card, followed by the conversion of this RGB image to a gray-scale one. The gray-scale image is then processed to produce gradient information and is subsequently sent to a Canny edge detector algorithm that generates an image with the edges found. After this initial processing, the algorithm splits in three parallel paths that are concerned with the detection and tracking of the blastocyst and the pipettes.

A. Finding and Analyzing the Blastocyst

A Hough transform is used to search for a blastocyst within the video frame images. The Hough transform uses a parametric description of the object of interest to map each point in the image space onto a curve in the parameter space, which is later searched to find the most consistent location of the object. The transform is implemented by treating each point in the image space as a constant and the parameters as variables, thus the idea is to have the points “vote” for the location of the feature that best suits them. The Hough transform was used because it is robust, fast, and yields good results for this particular task. For an introduction to Hough transforms, see [6].

Within the frame images, finding a blastocyst is a task similar to that of finding a circle because blastocysts are approximately circular too, see Fig. 7. Consequently, the implementation of the Hough transform for our problem uses the parametric equation for circles, which is given by \((x - a)^2 + (y - b)^2 = r^2\). Here \(a\) and \(b\) are the coordinates of the centre of the circle, and \(r\) is the radius of that circle.

Solving this problem would normally require a 3D Hough transform. However, the computations are simplified
assuming the diameter of blastocysts to be 100 μm, the typical value of this parameter. For the purpose of increased flexibility and applicability, i.e., embryo identification, the GUI allows the user to adjust this radius parameter.

After a blastocyst is identified, it is analyzed to determine the optimum injection point through the trophoblast cell layer (a single cell layer) that lines the blastocoeel. Such point should be located away from the blastocyst's inner cell mass, and preferably on the thinnest region of the trophoblast. This analysis starts by dividing the blastocyst image into sectors; each pixel inside the sectors is classified as either belonging to the blastocyst or to the background. This pixel segmentation uses a histogram-based local threshold. After the segmentation, a count is made of the number of blastocyst pixels within each sector. The analysis concludes when the sector with the least number of pixels is identified and selected as the best sector for injection.

The developed GUI allows the user to customize the blastocyst analysis too, mainly by changing the number of sectors used to find the best injection point. Sector choice can vary from 2 to 20, but experimentation determined that 8 sectors was an optimum number for the task: With 8 sectors the dimensions of the arc of each sector approximate the diameter of the injection pipette tip, which is normally in the order of 15 μm.

E. Finding and Tracking the Pipettes

The next major development in the visual processing software was finding and tracking the motion of the injection and holding pipettes during the microinjection experiments. Here, cross-correlation template matching was used. This technique searches for the object of interest based on the known outline of that object [7, 8]. This technique was chosen for its simplicity, flexibility, and because it can be made fast and robust enough for our application, see Fig. 6.

The implementation of a cross-correlation template matching can be described as: Given an image I = \( \{I(x, y) | 0 \leq x \leq X, 0 \leq y \leq Y\} \) and a template image T = \( \{T(m, n) | 0 \leq m \leq M, 0 \leq n \leq N\} \) containing the object of interest, the best matching position of the template over the image is given by the peak of the function

\[
R(x, y) = \sum_{m=0}^{M} \sum_{n=0}^{N} T(m, n)I(m-x, n-y)
\]

computed over all pixels in the image I.

A template matching implementation can be computationally expensive, since it requires the convolution of two images. It can also provide unreliable results if: (1) the object’s shape changes; (2) the object’s size or orientation is unknown; (3) unforeseen shadows appear due to changes in lighting; or (4) the object is partially occluded.

Despite all the above uncertainties, cross-correlation template matching is well-suited to finding the pipettes in the microscope images. During micromanipulation, the motions of the tip of the pipettes is translational only, no rotation is observable because the field of view of the camera is restricted to a very small area. Even in instances when the injection pipette is rotated by the kinematic stage, the motion on the image is simply perceived as translational about the Y-axis. Therefore, the template matching algorithm in this application does not need to be concerned with instances of orientation. The same can be said for changes in the size of objects due to motions towards or away from the camera. Consequently, there is no need to perform affine transformations on the template.

Another feature of the imaging system that helps simplify the template matching algorithm for our application is the controlled illumination. Under the microscope the illumination is highly uniform and is kept constant throughout the microinjection process. This means that the system does not need to worry about unexpected shadows or uneven illumination, so the algorithm does not need to implement normalization techniques.

C. Algorithm Performance

The first series of experiments associated with semi-automated microinjection concentrated on finding and analyzing blastocysts. If this section of the system’s image processing software did not perform adequately, i.e., show
performance metrics with a low rate of success, then it was reasoned that the performance of the complete system would be compromised. Figure 7 shows a collection of blastocyst images (top row) and the results obtained from the application of the finding and analyzing algorithm to those images (bottom row). The circle surrounding the blastocyst indicates its estimated location, and the red mark indicates the arc chosen by the system as being the one where the microinjection should be made. The performance of the algorithm was also tested using real images that were corrupted with Gaussian and impulse noise. This was accomplished by adding the noise to the frame images prior to its processing. A summary of the experimental conditions and the results for these experiments can be seen in Table 1.

V. CONCLUSIONS AND FUTURE WORK
A semi-automated system for microinjecting embryonic stem cells into blastocysts was designed and implemented. An interactive GUI and real-time video processing were combined to create a microinjection test-bed where cell microinjection can be treated like a computer game. A Hough transform was successfully applied to detect and track blastocysts. An analysis of the experimental data showed that the blastocysts were correctly located 90% of the time, and that sector selection was correct over 94% of the time. Simulation experiments determined that the sector selection error is less than 2 sectors when only 50% or more of the stimulated blastocyst pixels are used to build a test image, even with the presence of impulse noise with 5% probability.

Because this semi-automation has proved successful, the next phase of the research will concentrate on the development of an autonomous controller based on experiments with human operators and the application of machine learning methods. The development of a knowledge-based controller will increase the consistency and efficiency rates of the blastocyst microinjection. It is the next step towards full automation.

REFERENCES
APPENDIX 4. SPEEDING-UP VIDEO PROCESSING FOR BLASTOCYST MICROINJECTION

Speeding Up Video Processing for Blastocyst Microinjection

Leonardo Mattos and Edward Grant
Center for Robotics and Intelligent Machines
North Carolina State University
Raleigh, NC, 27606-7911 USA
{lmattos, egrant}@ncsu.edu

Randy Thresher
Animal Models Core Facility
University of North Carolina at Chapel Hill
Chapel Hill, NC, 27599-7264 USA
threshe@email.unc.edu

Abstract — This paper describes machine vision techniques that provide fast visual feedback for an automatic blastocyst microinjection system. The goals of the vision processing were to locate and track both the blastocysts and manipulation pipettes within the images and throughout the microinjection process. This was successfully accomplished using Hough transforms and cross-correlation template matching. Emphasis here is placed on a detailed description of the techniques applied to the algorithm to speed up the matching and to enable real-time visual feedback. These techniques allowed the algorithm to be, on average, 857 times faster than the original algorithm. This converts into a 99.89% reduction in the template matching processing time.

Index Terms — Biomanipulation, blastocyst microinjection, fast template matching.

I. INTRODUCTION

Blastocysts are early embryos, typically three or four days old for mice. At this developmental stage the embryo is a hollow microscopic ball of cells that have not yet differentiated except for the formation of the trophoblast, which is a layer of cells that surrounds the blastocyst [1]. This is the ideal time for the transfer and incorporation of genetically altered embryonic stem cells into the developing embryos. Injection now allows for the creation of gene-targeted animals.

Transgenic mice created by blastocyst microinjections are widely used as models custom tailored to address specific biological questions [2], including investigations related to a wide range of human afflictions such as diabetes, arteriosclerosis, hypertension, Alzheimer’s disease, and cancer [3].

Currently the injection of stem cells into blastocysts is performed by highly skilled and highly trained operators. The operations are tiring and can produce a low yield of viable injected embryos, sometimes as low as 20%. Automating this process will provide an increase in the overall efficiency of biomanipulations and, consequently, help advance life science research.

To automate a blastocyst microinjection the embryo has to be identified and tracked, and the micromanipulation pipettes have to be tracked too. This is done using real-time video acquired from a microscope. This knowledge is necessary if the operation is being controlled via visual feedback only, as is the case of the system being developed here. In this case the control scheme is known as visual servoing, for which Hutchinson et al. [4] note that the greatest challenge is the development of a fast and robust image processing algorithm.

Fortunately, research in biomanipulation is an active area, and part of that research deals with visual tracking. Biological cells have been tracked using snakes algorithms [5, 6], morphology and Bayesian estimation [7], and Hough transforms [8]. The visual tracking of pipettes has been mainly performed using template matching techniques based on: the minimization of the sum of squared differences [8]; minimization of the sum of squared distances [9]; or maximization of cross-correlations [5,10]. The published techniques work, but little attention has been paid to describing and analysing the methods employed to obtain real time vision systems. This paper addresses this issue by describing four template matching techniques used to speed up the vision algorithm of an automatic blastocyst microinjection system.

Here Hough transforms are used to find target objects (i.e., blastocysts), and cross-correlation template matching is used to track the microinjection pipettes. This is similar to the DNA injection system developed by Sun and Nelson [8]. The template matching speed-up methods are described using both mathematical formulas and measured performance gains, showing that the simple techniques provide up to 99.88% reduction in the template matching processing time.

II. ALGORITHM OVERVIEW

The video processing algorithm is key to the success of the automated blastocyst microinjection system. This algorithm has been developed to be robust and capable of providing accurate position and orientation data for: the blastocyst, the injection pipette, and the holding pipette.

The vision system uses Hough transforms to detect and track the blastocyst, and cross-correlation template matching to search for both the injection and the holding pipette. It also analyses the blastocyst to determine the optimum injection point through the trophoblast cell layer. Determining the pre-injection blastocyst orientation is necessary to minimize the chance of inflicting lethal damage to the embryo.
An overview of the implemented algorithm is presented in Fig. 1. First, the algorithm acquires an image from the video camera attached to the microscope. Second, this image is processed to produce gradient information, and a Canny edge detector algorithm [11] is applied to generate an image with the detected edges. Following this initial processing, the algorithm splits into three parallel paths each of which is used to detect and track either blastocysts or pipettes.

III. FINDING AND ANALYZING THE BLASTOCYST

As mentioned previously, the search for a blastocyst is based on the Hough transform [12]. This technique uses a parametric description of the objects of interest to find objects of a known shape within an image. Simply put, this operation treats each point in the image space as a constant and the parameters as variables, so the points in the image space are mapped to curves in the parameter space. This effectively allows the points to “vote” for the object locations that best suit them. Therefore, the most consistent object location is “elected.”

Here, finding the blastocysts was viewed as finding circles in video images. This approximation worked well because blastocysts are approximately circular, making the Hough transform a robust choice for this task. The algorithm is based on the parametric equation of a circle: $(x - a)^2 + (y - b)^2 = r^2$, where $a$ and $b$ are the coordinates of the center of the circle, and $r$ is its radius. Typically a Hough transform would involve a search for all three parameters $a$, $b$, and $r$. However, the computations are simplified here knowing that a typical blastocyst diameter is in the order of 100 μm. This is similar to the simplification used in [8]. Through knowing $r$, the Hough transform can locate the blastocyst’s central coordinates $(x, y)$.

Once a blastocyst is found, it is analysed to estimate the best injection point in the trophoblast cell layer. For safety reasons this point should be located furthest from the blastocyst’s inner cell mass, and preferably at the thinnest part of the trophoblast. The analysis divides the blastocyst image into eight sectors, and then it classifies the pixels inside each sector as either belonging to the blastocyst or to the background. This is a basic segmentation problem, one that uses a histogram-based local threshold. The analysis counts the number of blastocyst pixels within each sector, and it selects the sector with the least number of blastocyst pixels as the best sector for injection. Because this analysis is based on counting only, it is very fast.

The algorithm described in this section was successfully tested using pre-recorded video from manual blastocysts microinjections. The results show that blastocysts were correctly located 90% of the time, and that sector selection was correct over 94% of the time. Measured processing time for the localization and analysis was only time/frame on the employed computer (Pentium 4, 3.2GHz). Fig. 2 shows test images and their corresponding processed versions. It also shows that the algorithm for finding blastocysts is robust in terms of occlusions, and that it is effective in pinpointing the location of the embryo. Fig. 2 also shows that the sector analysis for finding the best injection point works well for detecting the thinnest parts of the trophoblast.

IV. FINDING AND TRACKING THE PIPETES

The next major task of the vision processing algorithm is finding and tracking the motion of the injection and holding pipettes during the microinjection procedure. Here, cross-correlation template matching, a technique that searches for the object of interest based on the known outline of that object, was used. This technique was chosen because it is simple, flexible, and robust for this application. For an introduction to matching theory, see [13].

The implementation of a cross-correlation template matching can be described as follows: Given an image $I = \{I(x, y)\}_{0 \leq x \leq X, 0 \leq y \leq Y}$ and a template image $T = \{T(m, n)\}_{0 \leq m \leq M, 0 \leq n \leq N}$ containing the object of interest, the best matching position of the template over the
image is given by the peak of the function
\[ R(x,y) = \sum_{x',y'} T(m,n)(x+x',y+y'n) \]
computed over all pixels in the image \( I \).

Cross-correlation template matching can be computationally expensive because it requires the convolution of two images. It also has the potential to produce unreliable results if: (1) the object’s shape changes; (2) the object’s size or orientation is unknown; (3) unforeseen shadows appear due to changes in lighting; or (4) the object is partially occluded.

Despite these potential problems, cross-correlation template matching is well suited for finding the two pipettes in the microscope images. Fortunately, during micromanipulation the motions of the pipettes consist of translations only. There is no rotation because the field of view of the camera is restricted to a very small area. Therefore, the implemented template matching algorithm is not concerned with instances of orientation. Furthermore, changes in the size of objects due to motions towards or away from the camera do not occur. Consequently, there is no need to perform affine transformations on the template.

The biomanipulation imaging system provides uniform illumination of the microinjection area. Again, this simplifies the template matching algorithm. There are no unexpected shadows or uneven illumination, so the vision algorithm does not need normalization techniques.

Finally, a successful template matching algorithm requires a good template image to be selected. A template that truly represents the object being searched for simplifies computations, and also increases the success rate of the matching algorithm. The developed system allows for the direct acquisition of the pipette templates directly from the microscope’s video. Template acquisition takes place at the beginning of each microinjection experiment, meaning the program can adapt to later changes in the pipette shapes.

The developed template matching algorithm yielded good matching results for the blastocyst microinjection application. Measured localization errors for the injection pipette presented a mean of 0.377 pixels and a standard deviation of 1.266 pixels. However, computational times were long (2.572s per frame on average). In order to address the computational speed issue, four speed-up techniques were implemented.

V. TEMPLATE MATCHING SPEED-UP

It is possible to reduce the processing time required for template matching on each video frame by applying simple search heuristics. In the case of the cell injection problem, it was noted that the positions of the pipettes on consecutive frames were highly correlated, suggesting the use of reduced and/or expanding search areas for the template matching computations. Noticing that template matching basically consists of moving the template over the image to check for a match suggests the control over the size of search steps. Remembering that the matching score is obtained from a convolution between the image and the template suggests the use of only non-zero template pixels for the computations. These techniques and their implementations are described below.

A. Window-based search

As pointed out in [4], a speed-up in template matching is obtained if the search for the object of interest is restricted to a small area of the video frame. The area selected is based on an a priori estimate of the object’s location, and the size of this search area is a function of the object’s rate of motion. Reducing the search area drastically decreases the computation time because only a sub-part of the video frame is used for the convolution with the template. The computational savings are described mathematically by considering the image \( I \) and the template \( T \) defined in section IV. If the search for the desired object is performed over the entire image, the number of elementary operations required for the computations is given by:

\[ O_t = 2MN(X-M+Y-N) \]  \hspace{1cm} (1)

This equation comes from the multiplication of the total number of points to check for a match, \( (X-M)(Y-N) \), by the number of pixels in the template, \( MN \). The coefficient 2 comes from the additions performed during the cross-correlation computations.

Now, let \( J \) be defined as a sub-image of \( I \) and assume it has an area \( r \) times smaller than the area of \( I \). Considering the \( X \) and \( Y \) dimensions of the image \( I \) respectively reduced by the fractions \( r_x \) and \( r_y \), the area reduction factor becomes \( r = r_x r_y \) and the sub-image \( J \) can be defined as:

\[ J(x, y) = I(x, y), 0 \leq x \leq X, 0 \leq y \leq Y \]  \hspace{1cm} (2)

Considering this image and the template \( T \), the number of performed operations for template matching is now given by:

\[ O_{t}(r_x, r_y) = 2MN(X-r_xM+Y-r_yN) \]  \hspace{1cm} (3)

Consequently, defining computational speed-up based on the number of elementary operations required for template matching, the following gain is obtained from equations (1) and (2):

\[ O_{t}(r_x, r_y) = \frac{(X-M+Y-N)}{X-M+Y-N} \]  \hspace{1cm} (4)

The impact of the window-based searched on the number of template matching operations, i.e., on the computational speed-up, is illustrated in Fig. 3. The figure shows a plot of \( G(r) \) assuming the template size to be 30x30 pixels and the frame image to be 320x240 pixels.

The results obtained by applying this technique to track the injection pipette in the blastocyst microinjection application showed a reduction in the average template
matching processing time from 2.572s/frame to 88ms/frame when the search area was reduced from full-frame (320x240 pixels) to a 110x70 pixels window. Furthermore, the mean localization error was only increased from 0.337 pixels with standard deviation 1.295 to 0.365 pixels with standard deviation 1.287. These results represent a speed-up of 29.23 frames, which is very close to the theoretical gain of 30.20 obtained from (3).

The window-based tracking of the pipettes is illustrated in Fig. 4. The figure shows a processed image displaying the selected tracking area for each object.

Fig. 4 Window-based tracking: Reduced areas are used to track the injection pipette (yellow), the holding pipette (green), and a glass bead representing the microspheres (red).

B. Expanding Search Area

The second template matching speed-up method applied to the vision algorithm consisted of implementing a search strategy based on a priori location information. Since small changes in object location are expected to occur between consecutive video frames, it is appropriate to start the search for the new object's position at its previously known location. Then, the search area is increased from this known location until the object is found.

In the case of the microinjection system the objects of interest are the tips of the pipettes; therefore the search processes are started at previous pipette locations, and continue at expanding neighborhoods of those locations. This search is terminated, and a pipette is considered found, when the compute cross-correlation value is larger than 0.95% of its maximum possible value. The maximum value is obtained by the cross-correlation of the template with itself.

Note that, from the point of view of processing time, this technique can only improve the performance of the template matching algorithm since in the worst case scenario the total number of operations is the same as it would be if this technique was not implemented. However, this technique has the potential to introduce localization errors since the search is stopped early if the breaking condition is met, and this can happen at a location where the matching is not optimal.

If we consider the previously defined image \( I \) and template \( T \), the use of the expanding search area technique can reduce the total number of elementary operations to \( 2MN \) on the best-case scenario, resulting in the following speed-up for the template matching algorithm:

- Best performance: \( G_2 = (X-M)(Y-M) \)  \( (4) \)
- Worst performance: \( G_2 = 1 \)

In our system, the use of this technique alone was able to decrease the average template matching processing time from 2.572s/frame to 0.208s/frame when the same injection pipette template mentioned earlier was used (68x88 pixels template). This represents an average speed-up of 12.6 times. Measurements of the localization errors introduced by this technique resulted in a mean error of only 1.955 pixels and a standard deviation of 2.277 pixels.

This expanding search area technique was implemented in conjunction with the previously described window-based method, providing an extra speed-up for computations. For the mentioned template, the average template matching processing time was reduced from 2.572s to 11ms, demonstrating a combined average speed-up of 197.85. The measured localization errors for this case presented a mean of 2.013 pixels and a standard deviation of 2.983 pixels.

C. Controlling the Search Step Size

Additional speed-up was obtained using this third method, which consists of skipping pixels during the search for the object of interest. This method is effective at the same if not further reducing the search area. The trade-off here is that it decreases the spatial resolution of the template matching, introducing extra errors to the object localizations.

The mathematical formulation to calculate the speed-up is very similar to the one developed earlier for the window-based search. In this case, consider \( n \) to be the search step size and the images \( I \) and \( T \) as previously defined. Based on the size of the images and on the value of \( n \), the total number of search points, \( E \), can be computed to be:

\[
E = \left( \frac{X-M}{n} + 1 \right) \left( \frac{Y-M}{n} + 1 \right)
\]  \( (5) \)

Therefore, the total number of elementary operations required for the template matching of the template \( T \) on the image \( I \) is now given by:

\[
G_2 = \frac{2MN}{n^2} (X-M)(Y-M)
\]  \( (6) \)
This result can be compared to (1) to provide the computational speed-up of this technique, \( G_2(n) \):

\[
G_2(n) = n^2
\]  

(7)

For this blastocyte microinjection system a decrease in spatial resolution is not critical because mechanical vibrations constantly cause the position of the pipettes to randomly change by two or three pixels. For this reason, the value of the search step size was set to 2. It can be noted from (7) that this value decreases the number of template matching operations by 75% while the decrease in spatial resolution is kept at a minimum.

Experiments showed that, with the search step size set to two, this technique decreased the average template matching processing time from 2.572s to 0.644s when a template with 68x28 pixels was used. This represents a speedup of 3.97 times. The measured localization errors presented a mean of 2.749 pixels and standard deviation of 2.036 pixels. When this technique was used in conjunction with the previously described ones, the average processing time was reduced to 11ms, resulting in a combined average speed-up of 233.82 times.

D. Using only Non-Zero Template Pixels

Cross-correlation template matching is typically based on the edge information contained in the images because edges can often faithfully represent the shape of objects of interest. As mentioned in [14], the use of edge-images for matching improves the cross-correlation results because it restricts the computations to the most important pixels of the images — the edges pixels themselves. This produces sharper peaks in the correlation matrix, facilitating the work of the peak detection algorithm used to pinpoint the location of the desired object. However, edges can also be used to speed-up computation.

The speed-up method described here is based on the observation that only a small percentage of the template pixels are edge pixels. The remaining pixels can be ignored during the cross-correlation computations because they are set to zero by the edge detection algorithm. Therefore, this speed-up method consists of: (1) preprocessing the template image to find the location of all edge pixels; and (2) using only these pixels to perform the template matching computations. The computational savings of this method arises from eliminating the need to scan the entire template image during the cross-correlation computations. Speed-up is obtained by simply ignoring pixels that are not edges.

Considering the previously defined image \( I \) and template \( T \), if we assume that the fraction of edge pixels in the template image is \( p \), the total number of operations required for the template matching can be expressed as:

\[
G_2 = 2pMN(X - M)(Y - N)
\]  

(8)

Therefore, the computational savings of using only non-zero template pixels is given by:

\[
G_2^p = \frac{1}{p}
\]  

(9)

In the case of our application, the typical value for \( p \) is 0.07 for the holding pipette and 0.15 for the injection pipette. These ratios can be observed in Fig. 5, which shows typical templates acquired for the injection and holding pipettes. The use of this speed-up method alone reduced the average template matching processing time from 2.572s to 0.488s per frame when a 68x28 injection pipette template with \( p = 0.1597 \) was used. The localization errors in this case had a mean of 0.446 pixels with a standard deviation of 2.05 pixels.

Fig. 5 Typical pipette templates and their corresponding edge images. (A) Holding pipette. (B) Injection pipette.

E. Combining the Speed-up Techniques

When all four techniques were applied, the average template matching processing time of each frame was reduced to 3ms, demonstrating a total speed-up of 857.33 times. The localization errors in this case presented a mean of 2.843 pixels and a standard deviation of 1.964 pixels. The analysis of the results also showed that the pipettes were correctly localized 99.9% of the time.

Consequently, the video processing was able to work in real-time, updating the location of the blastocyst and pipettes 10 times per second. This rate has proven to be adequate for the control of the microinjection system, a fact that has also been verified in [10].

A summary of the speed-up provided by the described methods is presented in Table 1. The theoretical values and the achieved speed-ups are shown, together with the measured localization errors introduced by each technique. The presented measurements are based on the analysis of 1000 simulated video frames for each case, which were created using a 68x28 pixels injection pipette template. Each of the frames was blurred with a Gaussian mask with variance 1 and corrupted by Gaussian noise with variance 100. Impulse noise with 3% probability was also added to all video frames.

VI. CONCLUSIONS

This paper described a machine vision algorithm that was developed and implemented to provide visual feedback for an automatic blastocyst microinjection system. The objectives of the developed system were to locate and track both the blastocyst and the micromanipulation pipettes within video frames captured from a microscope, and to do this as fast as possible. These goals were met using Hough transforms to find the blastocyst and cross-correlation template matching to search for the pipettes. Real-time visual feedback was achieved using four template matching speed-up techniques.
which were described in details in this paper. These techniques provided an average speedup of 857 times in relation to the processing time of a basic cross-correlation template matching algorithm. Expressed in another way, the final algorithm reduced the template matching processing time by 99.88% while introducing little localization errors.

Acknowledgement

The authors would like to thank Kimberly Klauckman for her invaluable assistance in assembling the microinjection system, for preparing all of the manipulation pipettes, and for providing the microinjection videos used in this research.

References

APPENDIX 5. NEW DEVELOPMENTS TOWARDS AUTOMATED BLASTOCYST MICROINJECTIONS

New Developments Towards Automated Blastocyst Microinjections

Leonardo Mattos and Edward Grant
Center for Robotics and Intelligent Machines
North Carolina State University
Raleigh, NC 27695-7911 USA
{ismattos, grant}@ncsu.edu

Randy Thresher and Kim Kluckman
Animal Models Core Facility
University of North Carolina at Chapel Hill
Chapel Hill, NC, 27599-7264 USA
{thresher, kluckman}@email.unc.edu

Abstract—This paper presents results related to our latest semi-automated blastocyst microinjection system. Here, the improvements made to the microinjection system are described and evaluated. First, after replacing the original piezo-electric kinematic stage by a DC motor-based robot manipulator, experimentation showed that the speed and the precise motion control of pipettes were improved. Second, by introducing an X-Y stage into the system, to manipulate the Petri dish around the microscope’s field of view, multiple microinjection speed was improved. Third, by using SSD template matching to track the injection pipette, rather than the cross-correlation template matching algorithm used in the original system, improvements were made to pipette localization. Under human control, this new semi-automated system gives improved microinjection performance metrics compared to previously obtained results. The system is also providing implicit human knowledge of the microinjection process via the human-control interface. It is the encoding of this knowledge that will lead to the first fully automated system. The semi-automated microinjection system is being tested and evaluated in the AMC at UNC-Chapel Hill.

Index Terms – Cell injection, biomanipulation, teleoperation, microsurgical robotic system.

I. INTRODUCTION

Currently, research in genetics and associated biomedical areas rely a great deal on the use of genetically modified mice for the discovery of gene functions and for the understanding of how defects in genes lead to the development of diseases [1]. Gene-targeted mice, for example, are often used as models of a wide range of serious human afflictions, including diabetes, arteriosclerosis, hypertension, Alzheimer’s disease, and cancer [2].

Gene-targeted mice, also known as knocked-out mice, are commonly created by the injection of genetically altered embryonic stem (ES) cells into early embryos during the blastocyst stage. These operations are called blastocyst microinjections. The success of these microinjections depends, to a large extent, on meticulous manipulations of the delicate cells. Therefore, the skills of the person performing the operations have a significant impact on the results of the microinjections.

According to published literature [3], operators need up to one full year of training to become proficient at injecting blastocysts. However, even with all that training, low survival rates of microinjected cells are encountered, often being between 40% and 70% [4].

The problems that affect the efficiency of blastocyst microinjections, as well as other biological micromanipulation tasks, are related to human errors and to the lack of repeatability. Thus, one possible way to address these problems is to automate the manipulations. A fully automated microinjection system is the goal of our research.

A pioneer in the area of embryo biomplementation automation was Ogawa. Together with Takahashi, Mizuno, Kashiwazaki, Yamane and Narishige, Ogawa put together a computer-controlled system for the manipulation of eggs and early embryos in 1985 [5]. Their system depended heavily on operator inputs, and the motion control of the manipulator worked in open loop based on manually defined starting and ending positions. Nevertheless, the system proved to successfully accomplish tasks such as bi секtion and microinjection. Ogawa and fellow researchers continued to work on automation improvements, and in 1992 they reported on a new system that automated the subzonal insemination of mouse ova [6]. In this case they used computer vision techniques (i.e., template matching) to locate the holding and injecting pipettes at the beginning of each microinjection procedure, and from that point the microinjection operation was performed in open-loop.

Within the past two decades other researchers have also worked in areas related to biomplementation automation, studying and proposing solutions to problems such as the positioning [7], [8], [9], holding [10], [11] and injection [12], [13] of biological cells. By the beginning of the 21st century, research on the automatic visual tracking of cells started to be reported upon [14], [15], and in 2002 Sun and Nelson introduced visual servoing as a major technique to enable closed-loop control in an automated cell injection system [16]. Their paper, which has become a classical reference in the biomplementation automation area, describes an automated system for embryo pronuclei DNA injection based on a single general-purpose microrobot and a hybrid visual servoing scheme. On that same year, Zhao et al. [17] also reported on a similar system using two micromanipulators which, following user inputs, could also autonomously capture the target cells for injection.

Recent research published in this area has demonstrated an apparent step back from the full automation concept and a general move towards the development of augmented reality systems to aid and improve manually controlled microinjections [18], [19], [20]. Many recent studies have also been concerned with a better characterization of the
cell membranes and of the forces involved in the microinjections [3], [21], [22]. This phenomenon may be due to the fact that, as Arai put it, “micromanipulation tasks are versatile. So it is difficult to realize a full automation system” [23]. Researchers may be following Arai’s suggestion to start by classifying the basic operations. However, there is an ever-growing need for improved consistency and efficiency of biomacromanipulation operations which can only be satisfied by full process automation. Therefore, we continue the push towards a fully automated system even without the knowledge of all of the intricate details of cell microinjections.

The idea is to let the system learn the task from skilled users, who have vast but implicit knowledge about it. Therefore, we have created a framework that allows a computer to observe and register all human actions during microinjection experiments.

The developed framework consists of a semi-automated blastocyst microinjection system that is teleoperated via a graphical user interface running on a desktop computer. This configuration enables the construction of a knowledge database from which intelligent controllers will be developed to fully automate the microinjection process.

This paper presents the latest improvements to the developed semi-automated system, which is described in greater details in [24], and which is currently in use at the Animal Models Core (AMC) Facility at UNC-Chapel Hill. Here we introduce the use of a new robot manipulator, and present experimental results that show how this new manipulator improves microinjection performance. In addition, an XY stage was incorporated to speedup multiple microinjection tasks. This is also described here, along with improvements and evaluation results of the employed vision system. Finally, preliminary results of blastocyst microinjections are provided.

II. THE SEMI-AUTOMATED MICROINJECTION SYSTEM

The developed teleoperated blastocyst microinjection system is shown in Fig. 1. The figure shows that the microinjections are performed under the microscope, which provides the necessary optical magnification and illumination levels for proper imaging the injection area. Video of that area is acquired by a CCD camera, and is sent to the desktop computer for displaying and processing. Real-time analysis of the video images determines the exact position of the blastocysts and pipettes during the microinjections experiments, which are recorded as experimental data and which will be later used as feedback information for automated operations.

When using this system to perform the microinjections, the operator sits in front of the computer screen and controls the entire procedure using a joystick. The generated motion commands are recorded as experimental data, and then processed and sent to the motion devices. Using the joystick the user can control: (a) 3-dimensional motions of the injection pipette through the motorized micromanipulator; (b) the activation of the piezo injector; (c) fluid motion on the holding and injection pipettes

Fig. 1. Microscope stage setup for teleoperated blastocyst microinjections through motorized micrometer syringes; and (d) the position of the petri dish through the motorized XY stage.

The developed computer interface is a key system element for making teleoperated micromanipulations possible. It provides the user with a micromanipulation environment that resembles a computer game, and allows the operations to be performed from a more comfortable and ergonomic setting; one that imposes less strain to the user’s eyes and body. The developed interface also allows the user to monitor and tune the motion devices and the video processing algorithm, what facilitates adjustments for peak performance and for the collection of reliable experimental data.

Initial tests with the system described in [24] showed that it required a few updates to become useful. The main problem lay with the original robot manipulator used, a piezo-electric kinematic stage, which caused excessive vibrations while in motion and tended to damage the embryos during injection. Another problem was the lack of an XY stage, which is essential for multiple microinjection tasks if we want to avoid having the operator go to the microscope stage before each microinjection.

We have addressed the problems mentioned above with the acquisition of an XY stage and a new robot micromanipulator, which were tested and evaluated as described in the next section. The image processing algorithm was also updated from [24] to improve its localization performance and robustness. The description and evaluation of the algorithm updates are presented ahead, followed by a description of preliminary blastocyst microinjection experiments.

III. SYSTEM IMPROVEMENTS

A. New Robot Manipulator

The updated blastocyst microinjection system is based on a new micromanipulator robot that provides vibration-free motions and is faster than the previously employed piezo-electric kinematic stage. That stage, the NewFocus 8082, was found unfit to the task of injecting blastocysts for the following reasons:
i. Excessive vibration during motion: Although not visible by eye, vibrations caused by the piezo-electric motors create fluid motions that make the collection of ES cells for injection impossible. Furthermore, injection experiments showed a high risk of fatal damage to the blastocysts when the NewFocus stage was used (7 out of 12 blastocyst injections were considered a failure by a microinjection specialist because of damage inflicted on the embryos’ membrane by pipette vibrations).

ii. Slow motions: The speed of motion of the NewFocus stage was a problem for this application. Its maximum velocity of 1.2 mm/min (or 20 μm/s) was found to be too slow for blastocyst manipulation. Blastocysts typically measure 100 μm in diameter, and operators like fast motions to quickly grab and position them for injection.

iii. Limited lifetime: The NewFocus 8082 kinematic stage is not designed for constant motions since its target application is the alignment of optical devices. The manufacturer specifies a lifetime of 15,000 cycles for each picomotor actuator, and defines a cycle as 1 mm of travel range out and back pushing a 5 lb axial load. Therefore, this stage is not appropriate for the blastocyst microinjection system, which needs a robot that can sustain constant motions.

The new Siskiyou MX7600R motorized micromanipulator provides motions that are virtually vibration-free because it is based on DC motors instead of piezo-electric or stepper motors. Successful ES cells collection experiments using this robot confirmed that its motions are smooth enough for the intended application.

The new robot is also able to move at speeds up to 1.7 mm/s of linear velocity while still maintaining its accuracy thanks to built-in encoders. The minimum controllable displacement of this Siskiyou robot is 0.1 μm. This is a much larger value than the 30 μm achieved by the NewFocus stage, but is enough for the blastocyst micromanipulation tasks since the smallest cells involved in the process (the ES cells) typically measure 10 μm in diameter.

In the developed system, the fast displacements and the precise motions required for blastocyst manipulations are achieved by applying an exponential function to the analog commands generated by the joystick, as shown in Fig. 2.

![Fig. 2. Exponential function applied to the joystick commands. Modifying the linear joystick values allows for fast displacement and for small and precise motions without changing any of the system’s settings.](image)

This way the operator can command small or large displacements without changing any of the system’s settings.

Furthermore, the joystick’s slider bar was set as a velocity gain control to allow for even faster motions, so the operator can easily increase the velocities directly from the joystick. Fig. 3 shows the restructured joystick function assignments for control of the microinjection system.

As a performance comparison between the two robots (and also to evaluate the visual servoing system), experiments were conducted to check how well the system could control the injection pipette motions. The experiments consisted of commanding the robots to follow a circular path with 130 μm in diameter. For both robots, the same visual servoing scheme was used to control the motions, but the control gains were adjusted differently to accommodate for the mechanical differences between them. The obtained results are presented in Fig. 4, which shows time-lapsed pictures of the experiments. The paths followed by the robots are marked with black dots, which represent the position of the injection pipette’s tip at each processed video frame. The figures also show the duration of one full turn around the circular path and the mean square errors (MSE) computed from the deviations from the desired path.

The results presented in Fig. 4 show that the Siskiyou micromanipulator was able to complete the task almost twice as fast and with almost a quarter of the MSE achieved by the NewFocus kinematic stage. However, this was mainly due to different calibrations of the visual servoing system, which is based on a PID control strategy. The path-
following MSE results obtained with the NewFocus stage could have been better if its speed was reduced, but this was not desired since the stage is already too slow for this application. Consequently, the PID values were set to make the stage move as fast as possible, giving rise to the observed overshoots and errors during the path-following experiment.

A measure of the improvement provided by the employed visual control system was obtained by performing the same experiments under operator control. In this case, path following was tested for three conditions:

i. The operator used the joystick to generate motion commands for the NewFocus robot.

ii. The operator used the joystick to generate motion commands for the Siskiyou robot.

iii. The operator used the manual micromanipulator to directly control the injection pipette motion.

The best results from a series of 10 trials for each experiment are shown in Fig. 5. The results show that the MSE obtained with the Siskiyou robot was about half of the value obtained with the NewFocus stage, therefore indicating that the operator had better control of the injection pipette’s position when using the Siskiyou robot. On the other hand, the direct manual control of the pipette proved to be faster and more precise than using either of the two robots. The observation is that the operator is very well trained in manual control of the motions of the pipette, but not familiar with the use of the joystick as yet. Therefore, we expect faster and more precise motions as the operator becomes more familiar with the teleoperated system. In any case, when comparing the results in Fig. 5 with the ones in Fig. 4, it is clear that the visual servoing system provides a much finer control over the injection pipette motions. This further motivates the goal of fully automating the blastocyst microinjection process since better motion control translates into reduced chances of inflicting lethal damage to the embryos.

The last experiment performed evaluated the Siskiyou micromanipulator in an “open-loop” trial using the same path-following experiment. In this case the goal was to evaluate the position control provided by the robot controller unit (the Siskiyou MC2000), which is based on readings from the motor’s encoders. This experiment was labelled as open-loop because the visual servoing system was turned off. Here, only the initial pipette location was obtained using the template matching algorithm. All subsequent motion commands consisted of relative motions solely based on expected pipette locations.

The obtained results from 10 trials of this last experiment were all very similar. As an example, a time-lapsed picture of one of the trials is presented in Fig. 6. It shows that the open-loop path-following trial resulted in high MSE. In contrast, a qualitative analysis of the path followed shows that it was reasonably good. This is important because it shows that position information obtained from the robot can be used to improve the injection pipette’s visual tracking algorithm. In the developed system this position information is being used to adjust the location of the pipette’s search window.

![Fig. 6. Open-loop servoing using the Siskiyou MX7600 robot.](image)

B. XY Stage

The new and improved semi-automated blastocyst microinjection system incorporates an XY stage to move the petri dish around during the procedures. This is useful because it facilitates the collection of ES cells and because it allows the operator to remotely carry out multiple microinjections without the need to physically go to the microscope stage to move the petri dish.

With the XY stage, the operator only goes to the microscope stage to setup the working wells at the beginning of the microinjections. After that all operations are performed from the computer station using the joystick. This saves time when multiple microinjections are performed, especially because ES cells and blastocysts are typically placed at different locations within the same working well. Furthermore, several working wells may be put on the same petri dish, so there is also the need to be able to move each of those sites to the microscope’s field of view. An example of a typical petri dish setup is shown in Fig. 7.

The installed XY stage is directly controlled from the system’s joystick or, alternatively, from buttons placed on the graphical user interface. It presents a step resolution of 10 µm in each direction, and can move at speeds up to 45 mm/s. A custom controller board drives this stage, and also provides storage area for the coordinates of 10 locations, allowing for prompt motion between different sites.
C. Improved Vision System

On our previously reported microinjection system [24], the employed vision processing was introduced as consisting of pipette tracking algorithms based on cross-correlation template matching, and a blastocyst localization algorithm based on Hough transforms. These were (and continue to be) used as source of information to record user actions and their respective reactions during the microinjection experiments. Another major aim of the vision system is to locate those objects in real-time, what enables the application of visual servoing techniques on a future fully automated system.

The previously reported evaluation results (see [24] and [25]) have demonstrated good localization performance for all three objects of interest: 93.5% success rate for the blastocyst localizations, 68.9% success rate for the pipettes localizations, and 94.0% success rate for the automatic selection of the injection area on the blastocyst’s trophoblast. However, these results were obtained either from pre-recorded microinjection videos or from simulated images. Therefore, they reflect expected values and not true performance measurements.

A better vision algorithm evaluation was performed once the system was moved to the AMC’s facility. This time, images of blastocysts and pipettes collected with our own system’s camera were used for the experiments. The results are presented in Table 1, and reveal even better performances than previously reported. The exception was the success rate obtained for the injection pipette localizations using the original cross-correlation template matching algorithm. This rate was found to be lower than expected due to the fact that the injection pipette tip is very small and does not present many distinctive features. Consequently, the acquired template presented few pixels and poor features, causing the cross-correlation template matching algorithm to fail more frequently than it was expected from simulations. Improvement was obtained by changing the search algorithm to a Sum-Of-Squared-Differences (SSD) template matching, as described in [16]. In this case all template pixels are used for matching, so more robust localization results are obtained.

Further improvement to the injection pipette tracking algorithm was obtained by using the position information provided by the micromanipulation robot to adjust the position of the pipette’s search window. This provided an error-filtering action that prevents the template to drift off from the real pipette location in noise conditions; however, further experimentation is necessary to obtain a quantitative measure of its effectiveness.

A picture demonstrating the performance of the current vision system is presented in Fig. 8. It shows good algorithm performance even in the presence of occlusions and when edges of different objects merge.

IV. Preliminary Microinjection Results

As mentioned earlier, we have performed a few preliminary blastocyst microinjection experiments using the current system setup. These experiments were all carried out by a microinjection specialist, who tested the initial system when it was based on the NewFocus stage, and also the updated system based on the Siskiyou robot. The obtained results, which are presented in Table 2, demonstrate the large performance improvement provided by the new robot micromanipulator. Furthermore, despite the fact that these were just initial experiments, the obtained 81% blastocyst survival rate has surpassed the 40-70% range commonly found for manual microinjections [4].

<table>
<thead>
<tr>
<th>Object</th>
<th>Localization (% correct)</th>
<th>Sector selection (% correct)</th>
<th>Number of images</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blastocyst</td>
<td>93.5%</td>
<td>68.9%</td>
<td>790</td>
</tr>
<tr>
<td>Injection pipette</td>
<td>94.0% (cross-correlation)</td>
<td>—</td>
<td>893</td>
</tr>
<tr>
<td>Injection pipette</td>
<td>99.3% (SSD)</td>
<td>—</td>
<td>893</td>
</tr>
<tr>
<td>Holding pipette</td>
<td>100% (cross-correlation)</td>
<td>—</td>
<td>360</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Robot employed</th>
<th># blastocysts that survived microinjection</th>
<th># injected blastocysts that survived for 24 hours</th>
<th>Number of injected blastocysts</th>
</tr>
</thead>
<tbody>
<tr>
<td>NewFocus 8082</td>
<td>5</td>
<td>2</td>
<td>12</td>
</tr>
<tr>
<td>Siskiyou MDC7600K</td>
<td>13</td>
<td>13</td>
<td>16</td>
</tr>
</tbody>
</table>
V. CONCLUSIONS AND FUTURE WORK

This paper presents the latest improvements and updates incorporated into our semi-automated blastocyst microinjection system, including the replacement of the original piezo-electric-based kinematic stage by a new DC motor-based micromanipulator. This updated system was evaluated in this paper by the undertaking of a comparative study that measured the performance of this new micromanipulator robot for a blastocyst manipulation task. The results obtained demonstrated that transferred vibrations were reduced, manipulation speed was increased, and precision motion control was improved. Moreover, in early experimentation, teleoperated microinjections using the new robot have resulted in an 81% blastocyst survival rate, surpassing the 40-70% range typically found for manual microinjections. Therefore, even greater survival rates are expected once the operator becomes more familiar with the controls.

The incorporation of an XY stage to speedup multiple microinjection tasks has also justified its inclusion into the system. Lastly, the results from the vision processing algorithm were evaluated. The results showed a success rate of nearly 100% for localizing objects of interest. This means that the system performs reliable experimental data collection and provides reliable position feedback for visual servoing.

For the near future, regular microinjection experiments are planned, including the implantation of the injected blastocysts into surrogate mothers to confirm they can develop into chimeras. Furthermore, we plan to add artificial intelligence to this microinjection system to fully automate the process. Such intelligence will come in the form of a knowledge-based controller, which will be developed from the knowledge being gathered by the use of the current semi-automated system.

ACKNOWLEDGMENT

The authors would like to thank Dr. John Muth from the NCSU’s ECE Department, and Dr. Marzloff and Dr. Corvus from the UNC’s School of Medicine for funding and the use of motorized micromanipulators.

REFERENCES