

**The A Team**

**Image Processing Tool for**

**Leidenfrost-Ratchet Systems**

**Requirements Specification for Version 2.0 (Final Draft)**

February 16, 2014

**Authored By:**

Sanan Aamir

Romando Garcia

Anne Lam

James Rowe

Hieu Tran

Table of Contents

1. Introduction 2
   1. Purpose 2
   2. Scope 2
      1. Main Objective 2
      2. Specific goals 2
   3. Overview of Document 3
2. System 3
   1. Development Environment 3
   2. System Architecture 3-4
3. Users 4
   1. User Interface Specification 4
   2. Use Cases and Use Case Diagram 4-5
4. Functional Requirements 5
   1. Issues 5
   2. Major Functions 5
   3. Major classes 6
   4. Minor system functions 6
   5. Non-functional requirements 6

4.5.1 Management 6

* + 1. Technical 6
    2. Performance 7
    3. Security 7
  1. System Evolution/Maintenance 7

1. Other Deliverables 7-8
2. Risks 8
3. Glossary 8

**1. Introduction/Overview**

The purpose of this document is to provide the main goals and requirements for improving the Image Processing Tool for present and future developers.

1.1 Purpose

This document serves to outline the details and organization of the tool's continued development as we make well-informed improvements to advance the processing capabilities of the software.

1.2 Scope

The scope comprises what we intend to implement and nothing more.

. Systeml is pixels to real world distances tool tips/intuitive usesequencege in the sequencetion of the needle and surface lo1.2.1 Main Objective

The main objectives of this edition of the tool are to increase automation, improve the user interface, provide graphical data, and speed up processing.

1.2.2 Specific Goals

Specific modifications to be brought up in this version include:

* Automatic determination of needle and ratchet location
  + if camera position is constant, determine location once using first image in sequence
  + if camera position is altered, determine location for each image in sequence
* Removal of the base image calibration
* Alteration of drop image manipulation (remove white glare)
* Drop volume measurement for each image
* Graphing of various plots using the extracted data
* Improvements to the user interface, including tool tips/intuitive use
* Increased processing efficiency
* Conversion of pixels to real world distances

1.2.3 Overview of Document

This document specifies the minimum system requirements as well as the users, deliverables, risks, and term definitions of the tool, for all developers involved, present and future.

**2. System**

Technical and functional requirements are outlined in order to have a clear path for development.

2.1 Development Environment

The system shall be ran and developed on a PC with a CD-RW drive at the least. Minimum PC specifications will be adequate in order to run the software successfully. Since the data outputs to a comma separated values file, a spreadsheet application is also necessary on the computing platform.

2.2 System Architecture

****

Fig 1. System Architecture with research system and Image Processing Tool.

The setup of the research requiring the Image Processing Tool involves a high speed camera, an injection needle that releases a drop of liquid, and a ratchet surface. The experiment entails the droplet falling from the needle onto the surface and moving across the surface while the camera captures many frames of the motion. The pictures are saved in TIFF format. Figure 1 above shows the flow of data from the experimental process with the camera, needle, and ratchet surface system, to the output of the Image Processing software.

**3. Users**

The primary users of the software are students and mechanical engineering professors, Dr. Guo and Dr. Ok, studying Leidenfrost-Ratchet Systems at Midwestern State University.

3.1 User Interface Specifications

The user interface will incorporate image folder uploading, image file listing, image removal, processing progress bar, and run data action. In addition, the user will input the speed of the camera in terms of frames per second in a numeric up/down tool. Additional numeric up/down tools will allow the user to set the bounding range of the droplet and the real world width (i.e. in cm) of the image. Additionally, there will be some component that allows the user to fine tune the location of the needle and surface determined by the system.

3.2 Use Cases

The specific actions the user will take during interaction with the system are shown in Figure 2, the use case diagram. To begin, the user will load the data set, that is, the set of images from the experiment. Then, the user will input the speed at which the camera captured the images in frames per second. The user will also input the width, or distance, between the needle tip and the surface so the system has a feel for the distance in the image in relation to reality. The final interactions will be to hit run and then possibly fine tune the locations of the needle and surface that were determined by the system if deemed necessary.

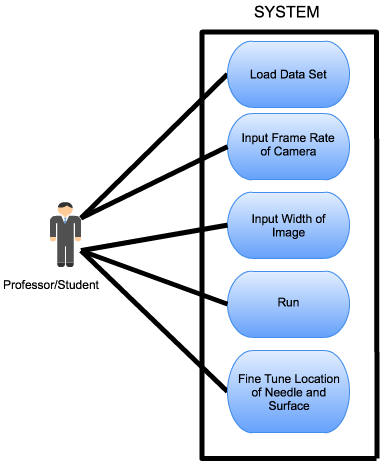


Fig 2. Use case diagram from system users, i.e. professors and students.

**4. Functional Requirements**

This section describes the operations of the tool and how the system should behave.

4.1 Issues

Some issues that may arise include minor inaccuracies due to poor image quality and difficulty adjusting to inconsistent needle and surface locations due to changes in camera position.

4.2 Major Functions

The main functions of the tool are to:

* Input and process images
* Determine the location of the injection needle and surface
* Calculate the centroid, acceleration, velocity, and volume of the droplet at every frame
* Output results to an Excel file and graphically display the data as a function of time

4.3 Major Classes

The major classes or divisions of functionality will include:

* Images
* Processing Form
* Results

4.4 Minor Functions

The minor function includes:

* Ability to fine tune the resultant locations of the needle and surface

4.5 Non-functional requirements

Non-functional requirements describe multiple aspects of development and software quality.

4.5.1 Management

The continued development of this tool will be ongoing for the Spring 2015 semester with the hopes of completing a polished, efficient, and accurate data software program. All members involved in ongoing development are invested in learning and improving the needs of Leidenfrost-Ratchet Systems research. The cost of development will be free.

4.5.2 Technical

The technical requirements necessary to achieve our purpose involve continued object-oriented utilization of the Visual Studio Integrated Development Environment and the C# language. NUnit testing software will be explored among others to be researched. Many technical documents will be drawn up as required for the Software Engineering course.

4.5.3 Performance

The first version of the tool reported a 5x increase in performance over the Optimus software. The goal of this version is to improve upon that by usage of threading and optimized code.

4.5.4 Security

The tool is only to be accessed and utilized by members involved in research group(s) on Leidenfrost-Ratchet Systems. Since this system requires the experimental setup isolated to the Engineering Department and is not security intensive, the team has no concerns about access to the software.

4.6 System Evolution and Maintenance

The group anticipates the tool to eventually evolve as the client desires additional functionality. However, this edition will focus solely on the goals enumerated above in section 4.1 -4.5. The organization of the software will be established well and will require little maintenance (until the next edition) once the final product has undergone thorough testing. Some maintenance may be required in the interval to make manipulations to the graphical data should data need different representation. In addition, we intend to structure the classes in an object-oriented way to allow feasible enhancements with future objectives.

**5. Other Deliverables**

In developing and planning the software, a prototype of the design will be drawn up to improve the style and flow of the current software. The delivery of this artifact will serve as a guide and mechanism for feedback.

A group of documents including the planning document, design diagrams, test case document, testing plan, and the final report will be provided. Interim and final presentations will be given as well.

Additionally, a user manual detailing the final changes and additions will serve to update the previous manual and specify the new usage procedures. Of course the software's implementation will be given in the form of a CD.

**6. Risk**

As with any endeavor, there are risks associated that may impede or impact the quality of this product. Recognizing the following potential risks, the team intends to prepare and work as organized and proactive as possible.

We believe we may face these common obstacles and describe them in table 1 below.

**Table 1.** Potential obstacles that may risk the successful completion of development.

|  |  |  |
| --- | --- | --- |
| **Obstacle** | **Likelihood**  **(1-10 scale, 10 being most likely)** | **Mitigation Techniques** |
| Division of focus because of other responsibilities | 7 | Don’t procrastinate any assignments. Put software engineering first |
| Poor communication between team members | 3 | Use all available means of communication (email, text, face-to-face) |
| Poor time management of specific tasks | 5 | Be organized from the start, equipped with a plan. |
| Scheduling conflicts between team members and possibly the client | 2 | Warn team members of any conflicts that may arise |
| Unforeseen emergencies involving health, family, etc. | 1 | Do as much work while we can. |

**7. Glossary**

|  |  |
| --- | --- |
| **Term** | **Definition** |
| **Ratchet surface** | A surface that is asymmetrical and periodic. |
| **Leidenfrost-Ratchet System** | A system involving a ratchet surface heated to a fluid's Leidenfrost point will allow a droplet of that fluid to spontaneous accelerate along that surface, even if it means traveling up slope. |
| **Object**-**oriented programming** (**OOP**) | A programming language model organized around objects rather than "actions" and data rather than logic. |
| **Droplet** | A very small drop of a liquid. |
| **TIFF** | An image format using lossless compression (or none) that may be edited and re-saved without losing image quality |

**8. References**

This document was completed with the guidance of the Requirements Outline and notes written by Dr. Stringfellow.