

**Image Processing Tool for**

**Leidenfrost-Ratchet Systems**

**Requirements Specificationfor Version 2.0 (First Draft)**

February 7, 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 3-6
   1. Target Environment 3
   2. Users 3
   3. Functional Requirements 3
      1. Issues 3
      2. Major Functions 4
      3. Major classes 4
      4. Minor system functions 4
   4. User Interface Specification 4
   5. Non-functional requirements 5
      1. Management 5
      2. Technical 5
      3. Performance 5
      4. Security 5
   6. System Evolution/Maintenance 6
3. Other Deliverables 6
4. Risks 6
5. System Architecture 7
6. Glossary 7

**1. Introduction/Overview**

A summary of the main goals of the software and the necessity of this document are given.

1.1 Purpose

The goal of this version of the Image Processing Tool is to make well-informed improvements to advance the processing capabilities of the software. The software's current capacity is to efficiently track a drop of liquid through images gathered from a high speed camera as it falls from an injection needle and travels along a ratchet surface. This document serves to outline the details and organization of the tool's development for ourselves and future developers.

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

1.3 Overview of Document

In summary, this document outlines and specifies the details of the tool in terms of planning, development, and function, 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 Target 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. 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. Since the data is outputs to a comma separated values file, a spreadsheet application is also necessary on the computing platform.

2.2 Users

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

2.3 Functional Requirements

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

2.3.1 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.

2.3.2 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

2.3.3 Major Classes

The major classes or divisions of functionality will include:

* Images
* Processing Form
* Results

2.3.4 Minor Functions

The minor function includes:

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

2.4 User Interface Specifications

The user interface will incorporate image folder uploading, image file listing, image removal, processing progress bar, andrun 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.

2.5 Non-functional requirements

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

2.5.1 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.

2.5.2 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.

2.5.3 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.

2.5.4 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.

2.6 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. 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.

**3. 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.

Additionally, a user manual detailing the final changes and additions will serve to update the previous manual and specify the new usage procedures.

**4. 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:

* Division of focus because of other responsibilities
* Poor communication between team members
* Poor time management of specific tasks
* Scheduling conflicts between team members and possibly the client
* Unforeseen emergencies involving health, family, etc.

**5. System Architecture**

****

Fig 1. This diagram portrays the flow of data from the experimental process to the output of the Image Processing software.

**6. 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. |