Software Requirements Specification for ImgBeamer: SEM Image Formation

Joachim de Fourestier

January 30, 2023

Contents

Revision History

Date	Version	Notes
2023/02/02	1.0	First version
Date 2	1.1	Notes

1 Reference Material

This section records information for easy reference.

1.1 Table of Units

Throughout this document SI (Système International d'Unités) is employed as the unit system. In addition to the basic units, several derived units are used as described below. For each unit, the symbol is given followed by a description of the unit and the SI name.

symbol	unit	SI
m	length	metre
s	time	second

Additional units used:

symbol	unit	name
px	pixel	picture element ?
nm	length	nanometre (10^{-9} m)
µm	length	micrometre (10^{-6} m)

1.2 Table of Symbols

The table that follows summarizes the symbols used in this document along with their units. The choice of symbols was made to be consistent with the heat transfer literature and with existing documentation for solar water heating systems. The symbols are listed in alphabetical order.

symbol	unit	description
d_p	nm	probe diameter (size)
d_i	nm	probe step size in image space
d_o	nm	probe step size in object space
$I_{n \times m}$	8-bit gray level	Image intensity matrix that has n rows and m columns.
$M_{n \times m}$	boolean	Mask / stencil matrix that has n rows and m columns.
$R_{n \times m}$	8-bit gray level	Resulting image intensity matrix that has n rows and m columns.

1.3 Abbreviations and Acronyms

symbol	description
A	Assumption
DD	Data Definition
GD	General Definition
GS	Goal Statement
IM	Instance Model
LC	Likely Change
PS	Physical System Description
R	Requirement
SRS	Software Requirements Specification
TM	Theoretical Model
SEM	Scanning Electron Microscope
EM	Electron Microscopy
LM	Light Microscopy
BSE	Backscattered Electron
SE	Secondary Electron
ROI	Region of Interest
FOV	Field of View
CCD	Charge Coupled Device
ImgBeamer	SEM image formation demo tool

[The SRS is not generally written, or read, sequentially. The SRS is a reference document. It is generally read in an ad hoc order, as the need arises. For writing an SRS, and for reading one for the first time, the suggested order of sections is:

- Goal Statement

- Instance Models

- Requirements

- Introduction

- Specific System Description

—TPLT]
[The template description comments should be disabled before submitting this document for grading. —TPLT]

2   Introduction

Images formed by Scanning Electron Microscope (SEM) are created using a specific process where the image quality can be greatly influenced by the imaging parameters given by the user, aside from any inherent electron optics limitation of a given instrument. The motivation of this project is to be able visualize qualitatively the influence of the spot profile (shape and size) and the rastering parameters (such as pixel size). Since the quality of an image is very often evaluated subjectively by an experienced microscope user, the idea behind this software is provide a way to show that adjusting the imaging parameters could lead to a loss of information or misinterpretation. Is is also to prove or disprove if there is a "rule of thumb" for the optimal spot-to-pixel size ratio.

2.1   Purpose of Document

This document serves as the software specification. The details of the requirements, limitations, definitions, models to be used are laid explicitly within this document.

2.2   Scope of Requirements

The scope of the requirements of this software is restricted to only the image formation process used in SEMs, more specifically the image processing from signal to an image or visual representions. It does not cover election-sample interactions or CCD (charge coupled device) detector voltage to signal value conversion. See the assumptions section (Section 4.2.1) for further details.

2.3   Characteristics of Intended Reader

The intended reader of this document should have a basic understanding of electron-optics concepts used in an SEM. This can be equated to a standard level two university physics (optics, electricity, magnetism, thermodynamics and modern physics) along with a standard undergraduate course related to materials characterization (concepts such as backscattered electrons, mean free path, and electron density). The reader should have a basic understanding of two-dimensional image processing techniques.

2.4   Organization of Document

This document contains an introduction to the software and its goals. It is meant to be a reference document to the readers and is based on a SRS template from Smith and Lai (2005); Smith et al. (2007). The document include a description of the scope, a detailed list of the terminology, definitions, and models used, as well as a more specific and detailed description of the problem and its solutions. This document also defines the requirements, as well as the likely changes and traceability details.

3   General System Description

This section provides general information about the system. It identifies the interfaces between the system and its environment, describes the user characteristics and lists the system constraints.

[The purpose of this section is to provide general information about the system so the specific requirements in the next section will be easier to understand. The general system description section is designed to be changeable independent of changes to the functional requirements documented in the specific system description. The general system description provides a context for a family of related models. The general description can stay the same, while specific details are changed between family members. —TPLT]

3.1   System Context

[Your system context will include a figure that shows the abstract view of the software. Often in a scientific context, the program can be viewed abstractly following the design pattern of Inputs → Calculations → Outputs. The system context will therefore often follow this pattern. The user provides inputs, the system does the calculations, and then provides the outputs to the user. The figure should not show all of the inputs, just an abstract view of the main categories of inputs (like material properties, geometry, etc.). Likewise, the outputs should be presented from an abstract point of view. In some cases the diagram will show other external entities, besides the user. For instance, when the software product is a library, the user will be another software program, not an actual end user. If there are system constraints that the software must work with external libraries, these libraries can also be shown on the System Context diagram. They should only be named with a specific library name if this is required by the system constraint. —TPLT]

Figure 1: System Context

[For each of the entities in the system context diagram its responsibilities should be listed. Whenever possible the system should check for data quality, but for some cases the user will need to assume that responsibility. The list of responsibilites should be about the inputs and outputs only, and they should be abstract. Details should not be presented here. However, the information should not be so abstract as to just say "inputs" and "outputs". A summarizing phrase can be used to characterize the inputs. For instance, saying "material properties" provides some information, but it stays away from the detail of listing every required properties. —TPLT]

- User Responsibilities:

  –

- ImgBeamer Responsibilities:

  – Detect data type mismatch, such as a string of characters instead of a floating point number

  –

3.2 User Characteristics

[This section summarizes the knowledge/skills expected of the user. Measuring usability, which is often a required non-function requirement, requires knowledge of a typical user. As mentioned above, the user is a different role from the "intended reader," as given in Section 2.3. As in Section 2.3, the user characteristics should be specific an unambiguous. For instance, "The end user of ImgBeamer should have an understanding of undergraduate Level 1 Calculus and Physics." —TPLT]

3.3 System Constraints

[System constraints differ from other type of requirements because they limit the developers' options in the system design and they identify how the eventual system must fit into the world. This is the only place in the SRS where design decisions can be specified. That is, the quality requirement for abstraction is relaxed here. However, system constraints should only be included if they are truly required. —TPLT]

4 Specific System Description

This section first presents the problem description, which gives a high-level view of the problem to be solved. This is followed by the solution characteristics specification, which presents the assumptions, theories, definitions and finally the instance models.

4.1 Problem Description

ImgBeamer is intended to help visualize and understand the influence of the spot profile and pixel size. It should be able to help determine the optimal spot-to-pixel-size ratio.

4.1.1 Terminology and Definitions

This subsection provides a list of terms that are used in the subsequent sections and their meaning, with the purpose of reducing ambiguity and making it easier to correctly understand the requirements:

Terminology	Definition
Backscattered Electron	–
Beam	The continuous focused stream of electrons accelerated at high speeds down the column of an electron microscope.
Brightness	–
Contrast	–
Electron-Sample Interactions	–
Electron Microscopy	–
Exact-sampling	–
Field of View	–
Ground Truth	–
Image	A representation of visual information that can be represented by 2D matrix where each cell represents a pixel with a specific value (intensity or colour). –
Image Quality	The perceived clarity, amount of information or detail representative of reality that is contained within an image.
Image Resolution	The size dimensions (width and height) of an image. Typically, it is defined in pixels.
Interaction Volume	–
Light Microscopy	The practice of microscopes that specifically use light as the source of illumination to produce an image or signal.
Nyquist Limit	–
Over-sampling	This is a sampling rate or size that is too large where the specificity or origin of the information is lost. This generally results in increased confidence, but at a loss of precision.
Pixel Size	–
Probe	Generally interchangeable with "beam". Contextually within this document, it is the final part of the beam that comes into contact with the surface of the sample (being analysed within the microscope) forming a "spot" with a targeted size and shape.
Raster Grid	–
Region of Interest	–
Resolution	–
Secondary Electron	–

Signal-to-Noise Ratio	–
Spot Shape	The overall shape of the spot or probe, whether is it a perfect circle or elongated ellipse, or even a halo or ring.
Spot Size	The overall diameter or average feret/caliper diameter of the probe.
Step Size	The length or distance of a "pixel" by which the probe is moved or deflected to when raster scanning the surface of a sample.
Subregion	–
Under-sampling	This is a sampling rate or size that is too small where the specificity or origin of the information is more restricted. This generally results in increased precision, but at a loss of confidence, or even accuracy.

4.1.2 Physical System Description

[The purpose of this section is to clearly and unambiguously state the physical system that is to be modelled. Effective problem solving requires a logical and organized approach. The statements on the physical system to be studied should cover enough information to solve the problem. The physical description involves element identification, where elements are defined as independent and separable items of the physical system. Some example elements include acceleration due to gravity, the mass of an object, and the size and shape of an object. Each element should be identified and labelled, with their interesting properties specified clearly. The physical description can also include interactions of the elements, such as the following: i) the interactions between the elements and their physical environment; ii) the interactions between elements; and, iii) the initial or boundary conditions. —TPLT]

The physical system of ImgBeamer, as shown in Figure 2 and 3, includes the following elements:

PS1:

PS2: ...

[A figure here makes sense for most SRS documents —TPLT]

4.1.3 Goal Statements

Given an image, the goal statements are:

GS1: A user-friendly straight forward tool to load and reprocess an image with options following the SEM image formation process.

GS2: Provide a quantitative metric of relative image quality of the resulting image created from the given parameters and input image (as ground truth).

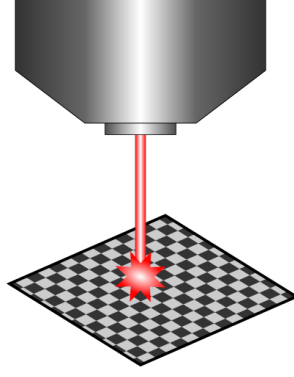GS3: The ability to visualize and export the reprocessed images.

Figure 2: Schematic drawing of an electron beam hitting the sample surface with an SEM chamber. (Image from: https://github.com/joedf/ImgBeamer)
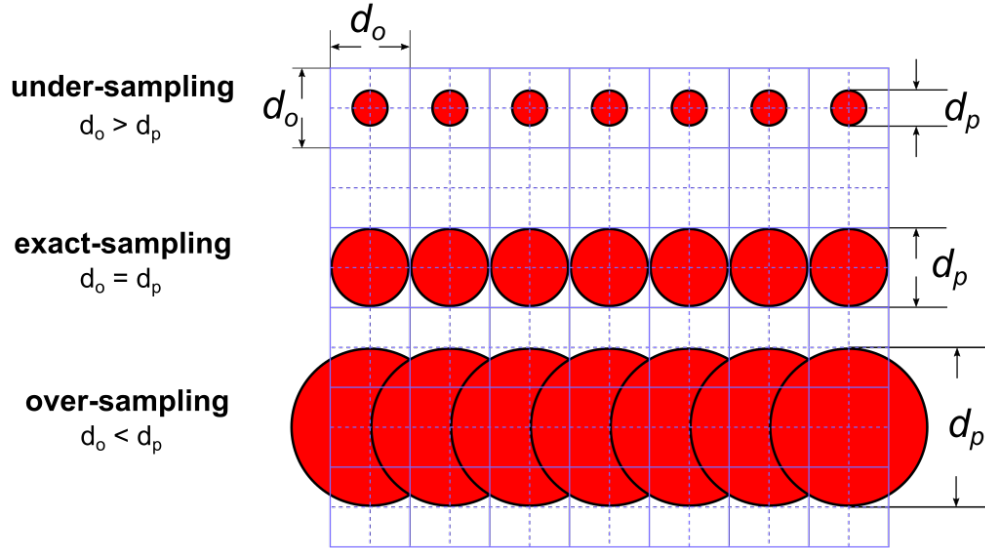

Figure 3: Schematic of the three sampling scenarios (under-sampling, exact-sampling, and over-sampling) and a visual representation of the pixel grid, d_p, and d_o. The dashed lines are to help visualize the center of each pixel cell. Adapted from Lifshin et al. (2014)

4.2   Solution Characteristics Specification

[This section specifies the information in the solution domain of the system to be developed. This section is intended to express what is required in such a way that analysts and stake-holders get a clear picture, and the latter will accept it. The purpose of this section is to reduce the problem into one expressed in mathematical terms. Mathematical expertise is used to extract the essentials from the underlying physical description of the problem, and to collect and substantiate all physical data pertinent to the problem. —TPLT]

[This section presents the solution characteristics by successively refining models. It starts with the abstract/general Theoretical Models (TMs) and refines them to the concrete/specific Instance Models (IMs). If necessary there are intermediate refinements to General Definitions (GDs). All of these refinements can potentially use Assumptions (A) and Data Definitions (DD). TMs are refined to create new models, that are called GMs or IMs. DDs are not refined; they are just used. GDs and IMs are derived, or refined, from other models. DDs are not derived; they are just given. TMs are also just given, but they are refined, not used. If a potential DD includes a derivation, then that means it is refining other models, which would make it a GD or an IM. —TPLT]

[The above makes a distinction between "refined" and "used." A model is refined to another model if it is changed by the refinement. When we change a general 3D equation to a 2D equation, we are making a refinement, by applying the assumption that the third dimension does not matter. If we use a definition, like the definition of density, we aren't refining, or changing that definition, we are just using it. —TPLT]

[The same information can be a TM in one problem and a DD in another. It is about how the information is used. In one problem the definition of acceleration can be a TM, in another it would be a DD. —TPLT]

[There is repetition between the information given in the different chunks (TM, GDs etc) with other information in the document. For instance, the meaning of the symbols, the units etc are repeated. This is so that the chunks can stand on their own when being read by a reviewer/user. It also facilitates reuse of the models in a different context. —TPLT]

[The relationships between the parts of the document are show in the following figure. In this diagram "may ref" has the same role as "uses" above. The figure adds "Likely Changes," which are able to reference (use) Assumptions. —TPLT]
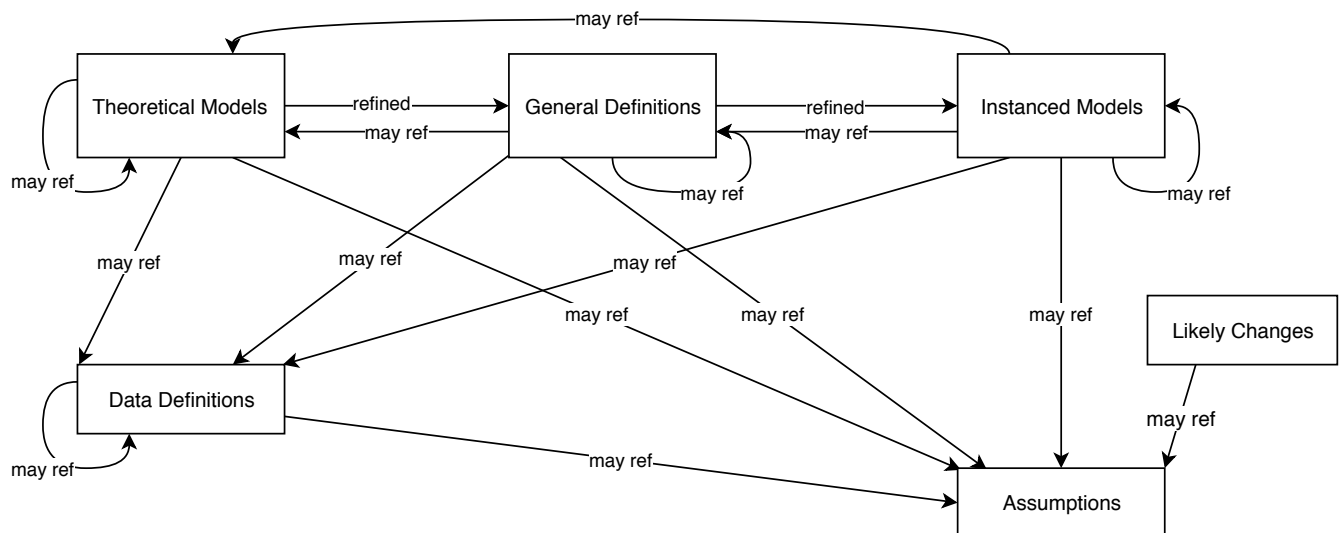
The instance models that govern ImgBeamer are presented in Subsection 4.2.6. The information to understand the meaning of the instance models and their derivation is also presented, so that the instance models can be verified.

4.2.1 Assumptions

[The assumptions are a refinement of the scope. The scope is general, where the assumptions are specific. All assumptions should be listed, even those that domain experts know so well that they are rarely (if ever) written down. —TPLT] [The document should not take for granted that the reader knows which assumptions have been made. In the case of unusual assumptions, it is recommended that the documentation either include, or point to, an explanation and justification for the assumption. —TPLT]

This section simplifies the original problem and helps in developing the theoretical model by filling in the missing information for the physical system. The numbers given in the square brackets refer to the theoretical model [T], general definition [GD], data definition [DD], instance model [IM], or likely change [LC], in which the respective assumption is used.

A1: [Short description of each assumption. Each assumption should have a meaningful label. Use cross-references to identify the appropriate traceability to T, GD, DD etc., using commands like dref, ddref etc. Each assumption should be atomic - that is, there should not be an explicit (or implicit) "and" in the text of an assumption. —TPLT]

4.2.2 Theoretical Models

[Theoretical models are sets of abstract mathematical equations or axioms for solving the problem described in Section "Physical System Description" (Section 4.1.2). Examples of theoretical models are physical laws, constitutive equations, relevant conversion factors, etc. —TPLT]

This section focuses on the general equations and laws that ImgBeamer is based on. [Modify the examples below for your problem, and add additional models as appropriate. —TPLT]

RefName: **T:COE**

Label: Conservation of thermal energy

Equation: $-\nabla \cdot \mathbf{q} + g = \rho C \frac{\partial T}{\partial t}$

Description: The above equation gives the conservation of energy for transient heat transfer in a material of specific heat capacity C ($\mathrm{J\,kg^{-1}\,^\circ C^{-1}}$) and density ρ ($\mathrm{kg\,m^{-3}}$), where $\mathbf{q}$ is the thermal flux vector ($\mathrm{W\,m^{-2}}$), g is the volumetric heat generation ($\mathrm{W\,m^{-3}}$), T is the temperature ($^\circ$C), t is time (s), and ∇ is the gradient operator. For this equation to apply, other forms of energy, such as mechanical energy, are assumed to be negligible in the system (A**??**). In general, the material properties (ρ and C) depend on temperature.

Notes: None.

Source: http://www.efunda.com/formulae/heat_transfer/conduction/overview_cond.cfm

Ref. By: GD**??**

Preconditions for T:COE: None

Derivation for T:COE: Not Applicable

["Ref. By" is used repeatedly with the different types of information. This stands for Referenced By. It means that the models, definitions and assumptions listed reference the current model, definition or assumption. This information is given for traceability. Ref. By provides a pointer in the opposite direction to what we commonly do. You still need to have a reference in the other direction pointing to the current model, definition or assumption. As an example, if T1 is referenced by G2, that means that G2 will explicitly include a reference to T1. —TPLT]

4.2.3   General Definitions

[General Definitions (GDs) are a refinement of one or more TMs, and/or of other GDs. The GDs are less abstract than the TMs. Generally the reduction in abstraction is possible through invoking (using/referencing) Assumptions. For instance, the TM could be Newton's

This section collects the laws and equations that will be used in building the instance models.

Number	GD1
Label	**Newton's law of cooling**
SI Units	$\mathrm{W\,m^{-2}}$
Equation	$q(t) = h\Delta T(t)$
Description	Newton's law of cooling describes convective cooling from a surface. The law is stated as: the rate of heat loss from a body is proportional to the difference in temperatures between the body and its surroundings. $q(t)$ is the thermal flux ($\mathrm{W\,m^{-2}}$). h is the heat transfer coefficient, assumed independent of T (A??) ($\mathrm{W\,m^{-2}\,^{\circ}C^{-1}}$). $\Delta T(t) = T(t) - T_{\mathrm{env}}(t)$ is the time-dependent thermal gradient between the environment and the object ($^{\circ}$C).
Source	Citation here
Ref. By	DD1, DD??

Detailed derivation of simplified rate of change of temperature

4.2.4 Data Definitions

This section collects and defines all the data needed to build the instance models. The dimension of each quantity is also given. [Modify the examples below for your problem, and add additional definitions as appropriate. —TPLT]

Number	DD1
Label	**Heat flux out of coil**
Symbol	q_C
SI Units	$\mathrm{W\,m^{-2}}$
Equation	$q_C(t) = h_C(T_C - T_W(t))$, over area A_C
Description	T_C is the temperature of the coil (°C). T_W is the temperature of the water (°C). The heat flux out of the coil, q_C ($\mathrm{W\,m^{-2}}$), is found by assuming that Newton's Law of Cooling applies (A**??**). This law (GD1) is used on the surface of the coil, which has area A_C ($\mathrm{m^2}$) and heat transfer coefficient h_C ($\mathrm{W\,m^{-2}\,{}^\circ C^{-1}}$). This equation assumes that the temperature of the coil is constant over time (A**??**) and that it does not vary along the length of the coil (A**??**).
Sources	Citation here
Ref. By	IM**??**

4.2.5 Data Types

[This section is optional. In many scientific computing programs it isn't necessary, since the inputs and outpus are straightforward types, like reals, integers, and sequences of reals and integers. However, for some problems it is very helpful to capture the type information. —TPLT]

[The data types are not derived; they are simply stated and used by other models. —TPLT]

[All data types must be used by at least one of the models. —TPLT]

[For the mathematical notation for expressing types, the recommendation is to use the notation of Hoffman and Strooper (1995). —TPLT]

This section collects and defines all the data types needed to document the models. [Modify the examples below for your problem, and add additional definitions as appropriate. —TPLT]

Type Name	Name for Type
Type Def	mathematical definition of the type
Description	description here
Sources	Citation here, if the type is borrowed from another source

4.2.6 Instance Models

This section transforms the problem defined in Section 4.1 into one which is expressed in mathematical terms. It uses concrete symbols defined in Section 4.2.4 to replace the abstract symbols in the models identified in Sections 4.2.2 and 4.2.3.

The goal of reprocessing an image GS1 will be solved by IM1. The goal of providing an image quality metric will be solved by IM2.

Number	IM1
Label	**Bit blit (as known as Bit Block Transfer)**
Input	an image $I_{n \times m}$ and a mask $M_{n \times m}$
Output	reprocessed image $R_{n \times m}$
Description	Given an image and mask (or stencil), a boolean operation is performed iteratively on each cell of the image matrix, based on the corresponding cell value found in the mask matrix at row i and column j.
	If the mask cell value (at i, j) is true (or any non-zero value), the corresponding image cell value is transfered/kept in the resulting/destination matrix.
	Otherwise, if the corresponding matrix cell (at i, j) is false (zero), then no value is transfered, and zero will inserted at i, j in the destination image matrix.
Sources	https://en.wikipedia.org/wiki/Bit_blit
Ref. By	

Number	IM2
Label	**Image quality metric / evaluation**
Input	a image $I_{n \times m}$ serving as ground truth and the image $R_{n \times m}$ to compare
Output	a positive ratio value ranging from 0 to 1.00.
Description	Given the original image (ground truth) and a reprocessed image, a metric must be calculated to express image similarity where 1.00 means a perfect match and 0.00 means absolutely no correlation or similarity found.
	If the images do no match in size, the smaller image shall be scaled up to match.
	The values do not need to be absolutely quantitative. They only need to be relatively comparable when only changing d_p or the spot shape.
Sources	
Ref. By	R8, NFR2

4.2.7 Input Data Constraints

Table 2 shows the data constraints on the input output variables. The column for physical constraints gives the physical limitations on the range of values that can be taken by the variable. The column for software constraints restricts the range of inputs to reasonable values. The software constraints will be helpful in the design stage for picking suitable algorithms. The constraints are conservative, to give the user of the model the flexibility to experiment with unusual situations. The column of typical values is intended to provide a feel for a common scenario. The uncertainty column provides an estimate of the confidence with which the physical quantities can be measured. This information would be part of the input if one were performing an uncertainty quantification exercise.

The specification parameters in Table 2 are listed in Table 3.

Table 2: Input Variables

Var	Physical Constraints	Software Constraints	Typical Value	Uncertainty
L	$L > 0$	$L_{\min} \leq L \leq L_{\max}$	1.5 m	10%

(*) [you might need to add some notes or clarifications —TPLT]

Table 3: Specification Parameter Values

Var	Value
$L_{\min}$	0.1 m

4.2.8 Properties of a Correct Solution

A correct solution must exhibit [fill in the details —TPLT]. [These properties are in addition to the stated requirements. There is no need to repeat the requirements here. These additional properties may not exist for every problem. Examples include conservation laws (like conservation of energy or mass) and known constraints on outputs, which are usually summarized in tabular form. A sample table is shown in Table 4 —TPLT]

Table 4: Output Variables

Var	Physical Constraints
T_W	$T_{\text{init}} \leq T_W \leq T_C$ (by A**??**)

[This section is not for test cases or techniques for verification and validation. Those topics will be addressed in the Verification and Validation plan. —TPLT]

5 Requirements

This section provides the functional requirements, the business tasks that the software is expected to complete, and the nonfunctional requirements, the qualities that the software is expected to exhibit.

5.1 Functional Requirements

R1: Accept an input image in the following formats:

  - PNG (Portable Net Graphics)
  - JPG (Joint Photographic Experts Group)
  - BMP (Bitmap)
  - preloaded example images

R2: Accept user input of the size and shape of the spot profile and ensure that all its defining values are positive real numbers.

R3: Accept user input of pixel size and ensure it is a positive real number.

R4: Accept user input to specify a subregion (FOV) for processing.

R5: Be easy to use with little to no setup required.

R6: Display a representative spot layout used for processing the image.

R7: Reprocess (using IM1) and display the resulting image to the user.

R8: Calculate and display an image quality metric (according to IM2) to the user.

5.2 Nonfunctional Requirements

The key nonfunctional requirements of this software are accuracy, usability, maintainability, and portability. These are listed below in detail:

NFR1: **Accuracy** The images produced should be not manipulated to produce results that subjectively represent the opinions of the author(s) or developer(s). Thus, the software should transparently follow the specifications and be verifiable by an expert in field. The trends in image quality metric are more important the metric values themselves.

NFR2: **Usability** The user should be able to intuitively use the software and quickly understand what is being displayed. Although the intended use is more of a qualitative nature, the user should be able to grasp any trends in the change of the image quality metric when the input parameters are changed. The software should have a simple user interface and be responsive when given a relatively small input image (ref to assumptions here?).

NFR3: **Maintainability** The code should follow a consistent style and be reasonably separated in multiple files were it makes sense. Function names should be no longer than 40 characters. Duplicate code should be avoided wherever possible. Comments should be plentiful, but short, and avoid any unnecessary use jargon or domain-specific terms.

NFR4: **Portability** The software should be cross-platorm (Windows, Linux, MacOS) with little to no setup required. This can be in the form of a web-application in an HTML5 compliant web-browser.

6 Likely Changes

LC1: [Give the likely changes, with a reference to the related assumption (aref), as appropriate. —TPLT]

7 Traceability Matrices and Graphs

The purpose of the traceability matrices is to provide easy references on what has to be additionally modified if a certain component is changed. Every time a component is changed, the items in the column of that component that are marked with an "X" may have to be modified as well. Table 5 shows the dependencies of theoretical models, general definitions, data definitions, and instance models with each other. Table 6 shows the dependencies of instance models, requirements, and data constraints on each other. Table 7 shows the dependencies of theoretical models, general definitions, data definitions, instance models, and likely changes on the assumptions.

[You will have to modify these tables for your problem. —TPLT]

[The traceability matrix is not generally symmetric. If GD1 uses A1, that means that GD1's derivation or presentation requires invocation of A1. A1 does not use GD1. A1 is "used by" GD1. —TPLT]

[The traceability matrix is challenging to maintain manually. Please do your best. In the future tools (like Drasil) will make this much easier. —TPLT]

The purpose of the traceability graphs is also to provide easy references on what has to be additionally modified if a certain component is changed. The arrows in the graphs represent dependencies. The component at the tail of an arrow is depended on by the component at the head of that arrow. Therefore, if a component is changed, the components that it points to should also be changed. Figure ?? shows the dependencies of theoretical models, general definitions, data definitions, instance models, likely changes, and assumptions on each other. Figure ?? shows the dependencies of instance models, requirements, and data constraints on each other.

	T??	T??	T??	GD1	GD??	DD1	DD??	DD??	DD??	IM??	IM??	IM??	IM??
T??													
T??			X										
T??													
GD1													
GD??	X												
DD1				X									
DD??				X									
DD??													
DD??								X					
IM??					X	X	X				X		
IM??					X		X		X	X			X
IM??		X											
IM??		X	X				X	X	X		X		

Table 5: Traceability Matrix Showing the Connections Between Items of Different Sections

	IM??	IM??	IM??	IM??	4.2.7	R??	R??
IM??		X				X	X
IM??	X			X		X	X
IM??						X	X
IM??		X				X	X
R??							
R??						X	
R??					X		
R??	X	X				X	X
R??	X						
R??		X					
R??			X				
R??				X			
R??			X	X			
R??		X					
R??		X					

Table 6: Traceability Matrix Showing the Connections Between Requirements and Instance Models

18

	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??
T??	X																		
T??																			
T??																			
GD1		X																	
GD??			X	X	X	X													
DD1							X	X	X										
DD??			X	X						X									
DD??																			
DD??																			
IM??											X	X		X	X	X			X
IM??												X	X			X	X	X	
IM??														X					X
IM??													X					X	
LC??				X															
LC??								X											
LC??									X										
LC??											X								
LC??												X							
LC??															X				

Table 7: Traceability Matrix Showing the Connections Between Assumptions and Other Items

8 Development Plan

[This section is optional. It is used to explain the plan for developing the software. In particular, this section gives a list of the order in which the requirements will be implemented. In the context of a course this is where you can indicate which requirements will be implemented as part of the course, and which will be "faked" as future work. This section can be organized as a prioritized list of requirements, or it could should the requirements that will be implemented for "phase 1", "phase 2", etc. —TPLT]

9 Values of Auxiliary Constants

[Show the values of the symbolic parameters introduced in the report. —TPLT]

[The definition of the requirements will likely call for SYMBOLIC_CONSTANTS. Their values are defined in this section for easy maintenance. —TPLT]

[The value of FRACTION, for the Maintainability NFR would be given here. —TPLT]

References

Daniel M. Hoffman and Paul A. Strooper. *Software Design, Automated Testing, and Maintenance: A Practical Approach.* International Thomson Computer Press, New York, NY, USA, 1995. URL http://citeseer.ist.psu.edu/428727.html.

Eric Lifshin, Yudhishthir P. Kandel, and Richard L. Moore. Improving Scanning Electron Microscope Resolution for Near Planar Samples Through the Use of Image Restoration. *Microscopy and Microanalysis*, 20(1):78–89, February 2014. ISSN 1431-9276, 1435-8115. doi: 10.1017/S1431927613013688. URL https://www.cambridge.org/core/journals/microscopy-and-microanalysis/article/improving-scanning-electron-microscope-resolution-for-near-planar-samples-through-the-use-of-image-restoration/A8715B3B599E6178E189B1887803FC79#. Publisher: Cambridge University Press.

W. Spencer Smith and Lei Lai. A new requirements template for scientific computing. In J. Ralyté, P. Àgerfalk, and N. Kraiem, editors, *Proceedings of the First International Workshop on Situational Requirements Engineering Processes – Methods, Techniques and Tools to Support Situation-Specific Requirements Engineering Processes, SREP'05*, pages 107–121, Paris, France, 2005. In conjunction with 13th IEEE International Requirements Engineering Conference.

W. Spencer Smith, Lei Lai, and Ridha Khedri. Requirements analysis for engineering computation: A systematic approach for improving software reliability. *Reliable Computing, Special Issue on Reliable Engineering Computation*, 13(1):83–107, February 2007.