



Universidade do Minho

Escola de Engenharia

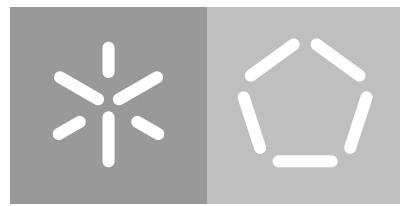
Departamento de Informática

Filipe Costa Oliveira

Im2Model

**Efficient computation to refine atomic models
for TEM image simulation and matching**

October 2018



Universidade do Minho

Escola de Engenharia

Departamento de Informática

Filipe Costa Oliveira

Im2Model

**Efficient computation to refine atomic models
for TEM image simulation and matching**

Master dissertation

Master Degree in Computer Science

Dissertation supervised by

Alberto José Proença

Daniel Grando Stroppa

October 2018

ACKNOWLEDGEMENTS

ABSTRACT

TEM (Transmission Electron Microscopy) is a well-established characterisation tool with wide use on the analysis of nanostructured materials. Even though modern TEM equipment is able to reach atomic resolution regularly, the suitable quantitative interpretation of experimental images is currently limited due to the required extensive and complex analyses.

A self-contained software solution to interpret TEM images of nano-structured materials is not currently available yet: these are often analysed in a qualitative way.

Only few examples of quantitative analysis of TEM images with atomic resolution are currently present in the literature, reflecting the complexity and the time-consuming aspect of the outlined analysis chain.

This dissertation aims to implement a software system, named Im2Model, addressing this gap in nano-materials characterisation. Im2Model will combine transmission electron microscopy, image correlation and matching procedures, enabling the determination of a three-dimensional atomic structure based strictly on a single high-resolution experimental image.

The software architecture approach will contain well-defined boundaries, clear responsibilities, and controlled dependencies. Performance engineering will be present in each step of the traditional software development process, aiming for both execution time-efficiency and resource usage efficiency, of the final software product.

RESUMO

A Microscopia Eletrónica de Transmissão (TEM) é uma ferramenta de caracterização bem estabelecida e com ampla utilização na análise de materiais nanoestruturados. Embora os equipamentos TEM sejam capazes de atingir resolução atómica, a interpretação adequada das imagens experimentais está actualmente limitada devido às extensas e complexas análises manuais necessárias.

Não existe actualmente uma ferramenta de software para interpretar imagens TEM de materiais nanoestruturados.

Existem até ao momento poucos exemplos de análise quantitativa de imagens TEM com resolução atómica, reflectindo a complexidade e o extenso tempo necessário para a sua análise.

Pretendemos implementar uma ferramenta de software, de nome Im2Model, que preencha esta lacuna na caracterização de nanomateriais. O Im2Model combinará a microscopia de transmissão eletrónica e algoritmos de correlação de imagem, permitindo a determinação de uma estrutura atómica tridimensional baseada estritamente em uma única imagem experimental de alta resolução.

Em cada etapa do processo de desenvolvimento de software estará presente uma visão de engenharia paralela e de alta performance, almejando a redução do tempo de execução da aplicação assim como uma utilização eficiente dos recursos de computação.

CONTENTS

| | | |
|--------------|---|-----------|
| 1 | INTRODUCTION | 1 |
| 1.1 | Context | 1 |
| 1.2 | Motivation | 1 |
| 1.3 | Contribution | 2 |
| 1.4 | Dissertation Structure | 3 |
| 2 | ATOMIC MODELS FOR TEM IMAGE SIMULATION AND MATCHING | 5 |
| 2.1 | Related work | 5 |
| 2.1.1 | Nanostructured materials crystallographic indexing from high resolution (S)TEM images | 5 |
| 2.1.2 | TEM image simulation | 8 |
| 2.1.3 | Image correlation | 8 |
| 2.2 | The Im2Model Work-flow | 9 |
| 3 | COMPUTATIONAL EFFICIENCY OF IM2MODEL | 12 |
| 3.1 | Performance: goals and its modelling | 12 |
| 3.2 | Describing Performance | 12 |
| 3.3 | Target Platforms | 13 |
| 3.4 | Target Operating Systems | 14 |
| 3.5 | Transformation for Performance | 14 |
| 3.5.1 | Data Parallelism | 14 |
| 3.5.2 | Data Alignment | 15 |
| 3.5.3 | Memory access and loop transformations | 15 |
| 3.5.4 | Processor Binding | 15 |
| 3.6 | Parallelization approaches | 15 |
| 3.6.1 | Shared Memory Parallelism | 16 |
| 3.6.2 | Distributed Memory Parallelism | 17 |
| 3.6.3 | Use of coprocessors as accelerators | 18 |
| 4 | THE PRODUCT | 19 |
| 4.1 | Use case analysis | 19 |
| 4.1.1 | Image management | 19 |
| 4.1.2 | Image simulation | 20 |
| 4.1.3 | Further simulated image transformations | 21 |
| 4.1.4 | T-focus map | 22 |
| 4.1.5 | Supercell estimation | 25 |
| 4.2 | Describing the problem | 25 |

Contents

| | |
|---|----|
| 4.3 Component architecture | 26 |
| 4.3.1 Dependency analysis | 26 |
| 5 EFFICIENT IMPLEMENTATION OF IM2MODEL WORKFLOW | 27 |
| 5.1 Describing the final goal | 27 |
| 5.1.1 Cross functional requirements | 27 |
| 5.1.2 Modelling the bottlenecks | 27 |
| 5.2 Modelling the solution | 28 |
| 5.2.1 Designing the data model | 28 |
| 5.2.2 Designing the application model | 28 |
| 5.3 The trouble with the modelled solution | 28 |
| 6 THE TEST BOUNDARY | 29 |
| 6.1 Service tests | 29 |
| 6.1.1 Implementing Service tests | 29 |
| 6.2 Performance Tests | 29 |
| 6.2.1 | 30 |
| 6.2.2 | 30 |
| 7 CONCLUSION | 31 |
| 7.1 The future of im2model | 31 |
| 7.1.1 Integrations | 31 |
| 7.1.2 Aiming for observability | 31 |
| 7.1.3 Trusting, but verifying | 31 |
| 7.1.4 Privacy, Tracking, and Compliance | 31 |

LIST OF FIGURES

| | | |
|----------|---|----|
| Figure 1 | Desired sample input and output of Im2Model | 2 |
| Figure 2 | Im2Cr workflow | 7 |
| Figure 3 | High resolution TEM modeling workflow | 10 |
| Figure 4 | Im2Model sample 2D-TEM experimental image input | 19 |
| Figure 5 | Example CeO ₂ nanostructure simulated image | 21 |
| Figure 6 | Border aberrations of HRTEM simulated images | 22 |
| Figure 7 | T-focus map of a CeO ₂ crystal | 23 |
| Figure 8 | 2D planar transformations present on the 2-step image correlation algorithm | 24 |
| Figure 9 | Edge detection of the experimental image ROI algorithm visualization | 25 |

1

INTRODUCTION

1.1 CONTEXT

Nanostructured materials (including nanoparticles) may exhibit new physical properties or new behaviours. Although some of these properties are already known, there may be many more unique physical properties not known to us yet, showing an extremely broad range of potential applications in several areas of research.

Research on new physical properties and applications of nanostructured materials is possible only when nanostructured materials are made available with desired morphology, size, and atomic composition.

Transmission electron microscopy (TEM) has been commonly used in characterisation of nano-structured materials, elucidating the nanometer scale structures. However, it has a serious drawback due to the fact that electron scattering information in a TEM image originates from a three dimensional sample but is projected onto an two-dimensional detector.

The structure information along the electron beam direction is superimposed at the image plane, therefore the information along the thickness direction of the sample is only an accumulated one leading to possibly resulting in tricky interpretations, pose a problem in reconstructing 3D images based on single high-resolution TEM images.

On the scanning mode of transmission electron microscopy (STEM) methods have been developed, like the High-Angle Annular Dark Field (HAADF), that allow the 3D reconstruction of the nano-structured material based on a single image. For the coherent imaging mode (TEM) the satisfactory solution for 3D reconstruction still relies on several high-resolutions images, taken in different orientations, of the same sample.

Figure 1 illustrates the desired sample input and output of application to be developed.

1.2 MOTIVATION

Even though modern TEM equipment is able to reach atomic resolution regularly, the suitable quantitative interpretation of experimental images is currently limited due to the required extensive and complex analyses.

1.3. Contribution

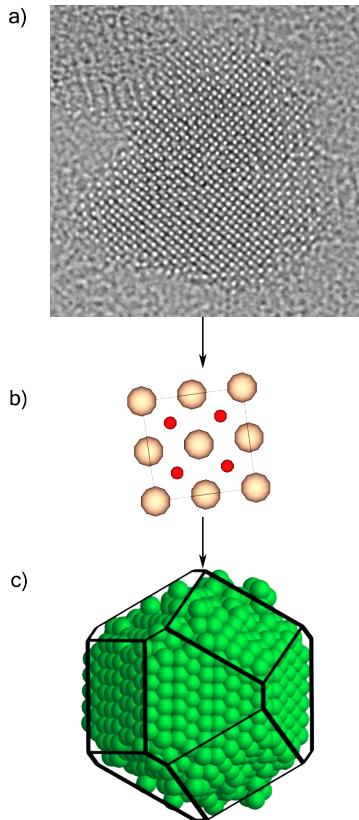


Figure 1: Desired sample input and output of Im2Model.

- a) 2D-TEM image of a CeO₂ (cerium oxide) crystal.
- b) Given crystallographic indexing and orientation information.
- c) Example 3D-TEM reconstruction of a CeO₂ (cerium oxide) crystal.

There is currently no efficient implementation of the preceding method and TEM results are often analysed in a qualitative way. Only few examples of quantitative analysis of TEM images with atomic resolution are currently present in the literature reflecting the complexity and the time-consuming aspect of the outlined analysis chain.

Moreover, more accurate representations and higher resolution models are directly associated with the usage of simulation for the physical processes. HPC is now an integrant component of the nanostructure materials studies, with simulations presenting ever increasing scales.

1.3 CONTRIBUTION

To address the presented problem, the introduction of an automated software tool capable of automating the structural information acquisition from coherent transmission electron microscopy imaging mode (TEM) of nanostructured materials, producing a three dimen-

1.4. Dissertation Structure

sional atomic model of the observed sample, will greatly reduce the time needed for the analysis of experimental images.

The possibility of creating an atomic model from a single high-resolution TEM image would increase the information presented to electron microscopy users and would support the automation of tasks and information retrieval(currently retrieved manually).

The software tool to be developed, named Im2Model, is therefore presented as a bridge between data visualisation of nano-structured materials crystallographic information and quantitative interpretation in a real-time basis of experimental attained high-resolution TEM images.

1.4 DISSERTATION STRUCTURE

This dissertation has 7 chapters, each with their summary presented bellow. The first 3 chapters go through the fundamental ideas that apply to both scientific software and high performance software, whether running on a single machine or distributed across a cluster of machines.

- **Introduction:** Presenting the motivation for this dissertation, and illustrating the desired sample input and output of Im2Model.
- **Atomic Models for TEM Image Simulation and Matching:** Presenting the state-of-the-art regarding the crystallographic indexing of nano-structured materials, methodologies for TEM image simulation, and image correlation algorithms, presenting us to the internals of scientific simulation.
- **Computational Efficiency:** Presenting the state-of-the-art art in terms of hardware and software performance metrics and modelling.

It examines what we actually mean by words like efficiency and scalability, and how we can try to achieve these goals.

Later, the following 4 chapters will turn to the particular issues of the simulation of the described physical process in distributed systems.

- **The Product:** Describing in detail the current solution legacy systems dependencies, and dividing different models and their components by their specific Ubiquitous Language and forming multiple Bounded Contexts.
- **Efficient implementations of Im2model workflow:**

1.4. Dissertation Structure

The previously described architecture, had functionality built on the same machine. These functional groups are tightly coupled, symbiotic systems. They depend on local, shared resources like disk, network interface, or inter-process communications.

This chapter focus on modelling and implementing the system to achieve the level of reliability, agility, and scale expected, built out of many different components running in different processes and communicating over distinct means spread across multiple machines, with elastic on-demand resource allocation. We will also be focusing attention on the resources that are exchanged between Bounded Contexts.

Considering we are modelling software to runs on several machines, connected by several distinct networks, faults and failures will be also taken in account in the design and implementation stages. We are no longer operating in an ideal system model, but in an non-deterministic one with the possibility of partial failures. Therefore, we will discuss a wide range of problems that can occur in the modelled distributed system.

- **The test boundary:**

This chapter dives into the different types of tests we have run to effectively and efficiently test the system functionalities when they spanned into a distributed system, while also breaking down the problems associated with testing finer-grained systems. It presents an overview of the results obtained by the preliminary experimental work developed.

- **Conclusion:** Presenting an overview of Im2Model major challenges until the current developing stage and future steps.

2

ATOMIC MODELS FOR TEM IMAGE SIMULATION AND MATCHING

2.1 RELATED WORK

The conventional electron tomography method to determine the three-dimensional structure of the nano-structured material relies on a number of images taken along different viewing angles (?). The acquisition of several high resolution images may add inaccuracy to the consistent reconstruction of the model, since the physical properties of the nano-structured material may be modified due to the high-energy electron beam required by TEM tomography. These small modifications on the acquired images may lead to different nanostructure reconstructions.

The 3D shape of several nanostructured materials has been reconstructed with a significant reduction of the required slice images, by using a discrete tomography technique in STEM (?).

The scanning process pixel by pixel in STEM generates high-resolution images. The captured intensity from each pixel in a High-Angle Annular Dark-Field imaging technique (HAADF) proves ideal for atomic reconstruction(?). It generates strong contrast that has a non-linear relation to the thickness of the observed atomic structure.

For TEM images this process can not be used, due to wave interferences registered on the captured image. To overcome this problem the current applied method uses an atomic structure model and a simulation of optical parameters, varied stepwise in such a way that the image calculated on this basis provides a best fit to the experimental image. With the comparison of the simulated images with the experimental image is determined if the model has a good approximation or not from the sample used. This method can be applied for both instrumental modes of TEM, being the most reliable process for atomic structure reconstruction.

2.1.1 *Nanostructured materials crystallographic indexing from high resolution (S)TEM images*

A precedent step of high resolution TEM image simulation is the characterization of nanocrystalline materials at higher resolutions as obtainable in a transmission electron-microscope

2.1. Related work

(TEM). Nanostructured materials crystallographic indexing refers to labeling the diffraction rings and spot with appropriate Miller indices (hkl).

Crystallographic indexing of using high resolution (S)TEM images can be carried out manually by the users by application of the reciprocal lattice and kinematical theory of electron diffraction, resulting in a reduced TEM characterization yield and significant user bias.

There is an alternative (S)TEM technique that works on the basis of a new software tool to aid the crystallographic indexing of nanostructured materials using high resolution (S)TEM images. Im2Cr (?) implementation aims for a minimal user interaction, supporting the detection of zone-axis oriented particles, and including an efficient peak detection process applied to the images Fourier Transform (FT). With basis on the FT peaks distances and relative angles, crystallographic indexation is carried out autonomously via comparison with a list of candidate structures named by the user, and a ranking of the best matching combinations of crystallographic structures and viewing zone axes are generated. This technique provides efficient and reliable data analysis reducing the complexity and time needed to produce a crystallographic analysis. Figure 2 illustrates the Im2Cr workflow.

2.1. Related work

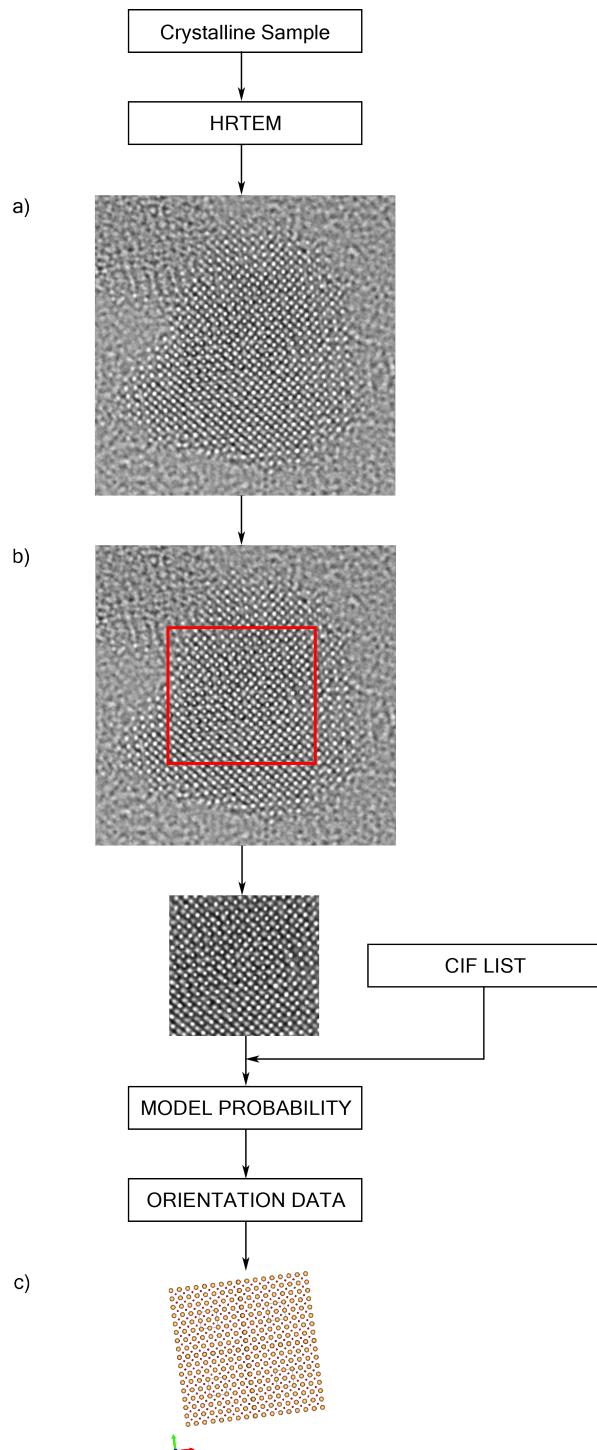


Figure 2: Im2Cr workflow.

- 2D-TEM image of a CeO₂ (cerium oxide) crystal.
- Selected ROI from the 2D-TEM image.
- Generated crystallographic structures and viewing zone axes.

2.1. Related work

2.1.2 TEM image simulation

Transmission electron microscopy image simulation is essential in TEM characterization of nanostructured materials. Here, the nanostructure is sectioned into thin layers along the direction of the electron beam. Each layer is prepared to represent a narrow height interval of the nanostructure, which allows the weak phase object approximation for the calculations at each layer. The multislice algorithm becomes exact only in the limit of very thin layers.

The scattered wave is then propagated in vacuum to the next layer and the process is repeated until the desired sample thickness is reached.

It is fundamental for the simulation software to employ a multislice simulation method capable of including the effects of geometrical aberrations (image properties), and wavefront aberrations (transfer function properties).

Due to the crystalline materials long range order property, super cells are utilised to simplify the TEM multislice simulation input with respect to the atoms information.

There exist a number of free as well as commercially available software based on the multislice algorithm for conventional TEM image simulations:

1. Dr. Probe (?)
2. QSTEM (?)
3. JEMS (?)
4. MacTempas (?)
5. STEM_CELL (?)

The multislice simulation will be carried out using Dr. Probe (?) command-line software tools, composed of three tools, each related to one of the three Dr. Probe simulation steps. This separation gives birth to future work-flow optimisations when doing parameter variations, as not all steps need to be repeated depending on the level where the parameter variation takes place.

2.1.3 Image correlation

Algorithms for aligning images are among the oldest and most widely used in computer vision. One of the most commonly used algorithm is to shift or warp the images relative to each other and to look at how much the pixels agree, directly minimizing pixel-to-pixel dissimilarities, normally referred as the patch-based translational alignment (?). A variety of such parametric motion models are possible, from simple 2D transforms, to most complex 3D perspective transformations.

2.2. The Im2Model Work-flow

To use the precedent method, a suitable error metric must first be chosen to compare the images, which normally is achieved recurring to one of the following image similarity metrics(?):

- **Squared difference** – Match the squared difference of the patch and the input image. Perfect match will be 0, and bad matches will be large.
- **Cross Correlation** – Multiplicative match the patch against the input image. Perfect match will be large, and bad matches will be small or 0.
- **Correlation Coefficient** – Match the patch relative to its mean, against the input image relative to its mean. Perfect match will be 1, and a perfect mismatch will be -1. 0 represents random alignments.

We should obtain more accurate matches, at the cost of more computation effort, as we move from simple measurement methods like the squared difference, to more sophisticated ones like correlation coefficient.

The normalized methods of the precedent error metrics are extremely useful since they help reducing the effects of lightning differences between the images being correlated.

Computer vision operations will be carried out by recurring to the free Open Source Computer Vision Library (OpenCV) that optionally can be highly optimized by loading the commercial Intel Integrated Performance Primitives (IPP)(?), or developing GPU-accelerated image processing that will be integrated with native OpenCV C++11 methods.

2.2 THE IM2MODEL WORK-FLOW

To achieve the Im2Model goal, a preliminary crystallographic analysis of TEM images will be carried using the Im2Cr software, recently developed at UMinho (?). Im2Cr provides the most likely crystalline structure and its orientation from a single region of interest in an experimental image.

A validation step will be then carried out through an image simulation of a small segment and its comparison with the experimental image. This procedure is often referred as thickness-defocus maps in TEM simulation. This step will be carried out using Dr. Probe (?) software, and (i) will confirm the crystallographic indexing by Im2Cr (?), (ii) validate the experimental setup parameters informed by the user, and (iii) estimate the local sample thickness.

A super cell model representing the whole structure will be built based on the crystallographic structure retrieved by Im2Cr (?) and the estimated dimensions of a certain region of interest. Iterative steps of image simulation, images comparison and model refinement will be carried out until a convergence threshold.

2.2. The Im2Model Work-flow

During these steps, the atomic model and the TEM experimental parameters will be refined, leading to a complete description of the sample structure.

Figure 3 illustrates the high-resolution TEM modelling workflow, including the preliminary phases required as Im2Model input parameters, provided by Im2Cr (?) and by the experimental results.

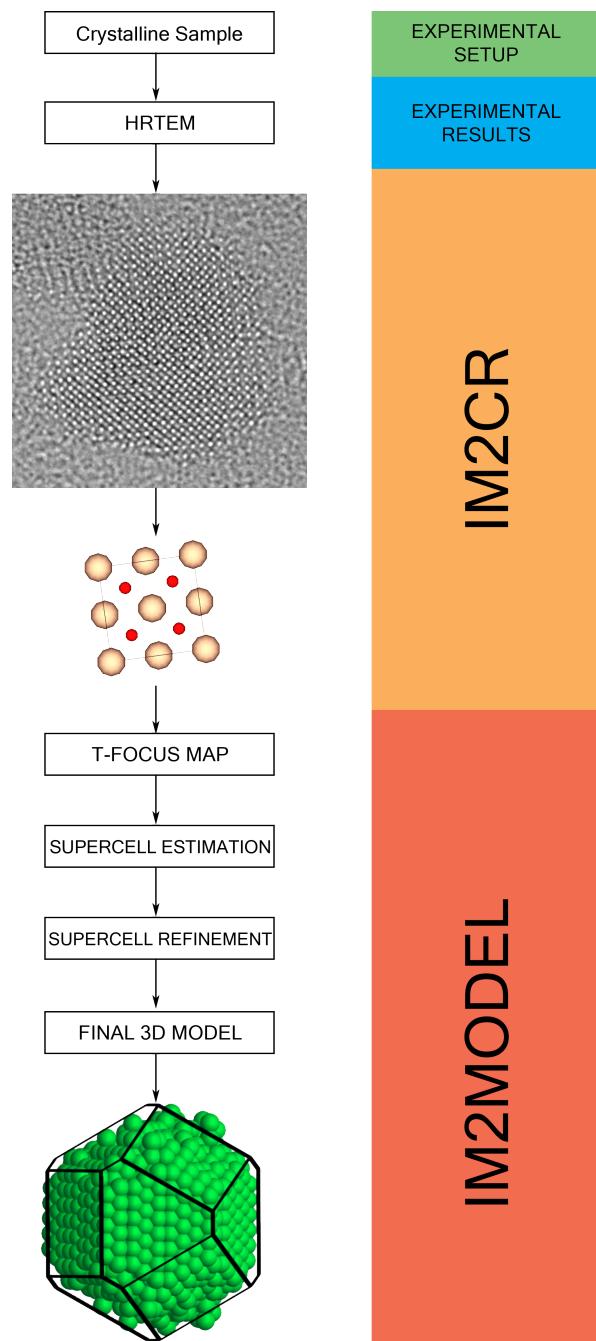


Figure 3: High resolution TEM modeling workflow

2.2. The Im2Model Work-flow

The integration and refinement of the resulting application code and associated data structures will follow efficiency methods to lead to faster execution times and optimum resource usage, taking advantage of code vectorization, data locality (both NUMA and cache levels), code parallelism (both thread-level and process-level) and computing paradigms (using computing accelerators, such as Intel MIC devices and NVidia GPUs). If such levels of efficiency are reached, it could open new horizons regarding the use of TEM technology.

3

COMPUTATIONAL EFFICIENCY OF IM₂MODEL

3.1 PERFORMANCE: GOALS AND ITS MODELLING

A performance goal provides direction for our performance analysis work and will help the optimization pipeline. Our main objective is to provide a time-efficient and unbiased work-flow for the quantitative interpretation of high-resolution TEM images, by the use of efficient computation methods, enhancing the productivity on materials characterization at the atomic scale. Considering that the work-flow presents such a demanding and wide-ranging requirements, a single approach can no longer meet all of its computing, data processing and storage needs. Instead, the work is broken down into smaller tasks that can be performed efficiently, and those different tools are stitched together using an distributed system design approach.

The composite distributed system must provide certain guarantees:

1. Reliability
2. Scalability
3. Maitainability

The resource utilization efficiency for a given application workload can be considered as the secondary objective.

3.2 DESCRIBING PERFORMANCE

This chapter will refer to some commonly used techniques by which application performance can be improved: selecting an I/O size, caching, buffering, concurrency and parallelism, non-blocking I/O, and processor binding. Some of them are strictly related to the target platforms and to the target operating systems.

Performance issues can arise from anywhere, including software, hardware, and any component along the data path. Its is important to include when available and applicable,

3.3. Target Platforms

network, operating system, file system, memory, and CPU performance analysis which can also identify application level issues.

3.3 TARGET PLATFORMS

Concerning the computational efficiency evaluation of Im2Model the target platform should extend the hardware performance limits to the maximum. Therefore clustering environments are ideal for high performance computing development.

The SeARCH6 cluster at the University of Minho (Services and Advanced Research Computing with HTC/HPC clusters) is a computing infrastructure containing 800 cores distributed over 54 nodes with dual Intel Xeon processor. Each node contains two multi-core CPU devices, either Ivy Bridge processors (E5-2650v2, E5-2670v2, E5-2695v29) or Nehalem processors (E5520, X5650, E5649), and eventually one or more computing accelerator, for example, a GPU (Graphics Processing Unit) or a many-core CPU device, such as the Intel Many Integrated Cores (MIC) device.

Regarding the coprocessors and computing accelerators distribution SeARCH6 has 21 accelerators distributed over 12 nodes: 9 Intel Xeon Phi (87120 15110 and) and 12 NVIDIA Tesla (5x K20m Kepler, Fermi M2090 1x, 2x and 4x Fermi M2070 Fermi C2050).

The access time for main memory is non-uniform (NUMA) – therefore varies based on its location relative to the CPU memory architecture for each two-processor computing node. The main memory amount ranges from 8 to 64GB, with 28 nodes possessing the largest configuration. The speed of the memory bus, for any architecture, is often dictated by the memory interface standard supported by the processor and system board. Therefore it should be evaluated based on the CPU units description.

Disk I/O can cause significant application latency and is therefore an important target of systems performance analysis. Flash-memory-based solid-state disks with 240GB will be the main storage system for each computing node. A distributed Network-Attached storage system of 98TB is integrate in the cluster environment, consisting of Dot Hill SAN with 40TB of raw capacity, connected by 16Gbit/s Fibre Channel links to 4 dual-Xeon nodes, each with 12TB of raw capacity. These nodes also facilitate access to the entire storage space via NFS and GlusterFS.

Two network interfaces are available across the cluster, one with low latency 10 Gbit/s Mirynet, and another 1 Gbit/s Ethernet.

In order to obtain the best performance possible the described clustering platform will be used in the evaluation of performance in the development and testing stages. In the production stage the application is to be run on desktops or laptops, normally present in TEM laboratories.

3.4. Target Operating Systems

Despite the described clustering environment possessing features above the currently used laptops, the evolution of the computer systems will soon match or overcome these characteristics.

3.4 TARGET OPERATING SYSTEMS

An understanding of the operating system and its kernel is essential for systems performance analysis. The SeARCH6 cluster is based on the Rocks 6.1 cluster management distribution, a Linux distribution intended for high-performance computing environments.

Other Unix based distributions will also be validated for later development stages, as well as a Microsoft Windows version.

3.5 TRANSFORMATION FOR PERFORMANCE

There are a number of possible user-level optimizations that have been found effective for ultimate performance. It is unlikely that peak performance will be achieved without considering some of these optimizations:

3.5.1 *Data Parallelism*

Vectorization

Vector instructions are a special set of instructions based on the Single Instruction Multiple Data (SIMD) model, where a single instruction is simultaneously applied on multiple data.

In previous generations of processors, vectorization of operations was of lesser importance, mainly because of the shorter vector units. With the arise of 256-bit and 512-bit long vectors utilization is increasingly necessary to achieve high performance.

This has drawn the auto-vectorization capabilities of modern compilers into focus, however such compilers at best could only vectorize a few classes of applications with regular memory access and computation patterns, such as structured grids or multimedia.

Gaining good performance from vector units is not only important for CPUs but also for emerging co-processors. Very explicit programming techniques specific to the hardware are required to gain maximum performance.

3.6. Parallelization approaches

3.5.2 Data Alignment

Vectorization works best on unit-stride vectors, the data being consumed is contiguous in memory. Data structure transformations can increase the amount of data accessed with unit-strides and should therefore be considered.

3.5.3 Memory access and loop transformations

Cache Optimisation

The most effective use of caches comes by paying attention to maximising the locality of references, blocking to fit in L2 cache dimensions present on the SeARCH6 processors, and ensuring that prefetching is utilised (by hardware, by compiler, by library or by explicit program controls).

3.5.4 Processor Binding

For NUMA environments like SeARCH6 compute nodes, it can be advantageous for a process or thread to remain running on a single CPU and to run on the same CPU as it did previously after performing I/O. This can improve the memory locality of the application, reducing the cycles for memory I/O and improving overall application performance.

3.6 PARALLELIZATION APPROACHES

Time-sharing systems provide the ability to load and begin executing multiple runnable programs, vulgarly described as program concurrency. While their runtimes may overlap, they do not necessarily execute on-CPU at the same instant. Each of these programs may be an application process.

Apart from executing different applications concurrently, different functions or code sections within an application can also be made concurrent. This can be achieved using multiple processes (multiprocess) or multiple threads (multithreaded), each performing its own task.

Frameworks for parallelising both on shared and distributed memory paradigms are presented next.

3.6. Parallelization approaches

3.6.1 Shared Memory Parallelism

Posix Threads (PThreads)

Pthreads programming model is part of the C programming language and composed of programming types and procedure calls, that get called and included by a pthread.h header thread library.

To fully exploit the computational potential presented by Pthreads, a program has to be structured into distinct autonomous processes which then get implemented and run in parallel.

Posix Threads are composed of sub-routines, mainly classified as thread management, mutexes, condition variable, and synchronization sub-routines.

OpenMP 4.5

OpenMP is a portable, scalable model that gives parallel programmers a simple and flexible interface for developing portable parallel applications. OpenMP is primarily designed for shared memory multiprocessors, taking a directive-based approach for supporting parallelism, allowing the same code base to be used for single-processor and multiprocessor computing platforms, and promoting an incremental approach to parallelism.

The latest versions of OpenMP standards have introduced a number of new directives designed to support heterogeneous computational offloading in order to target many-core devices, while also enabling multiple levels of parallelism. The developer can now describe a group of thread teams, and then distribute the iterations of a loop to the master threads in each team. Depending on the target, this could place threads on the cores of a CPU, or block them onto the streaming multiprocessors of a GPU or coprocessor.

Intel Thread Building Blocks (TBB)

Intel Threading Building Blocks (TBB) is a C++ template library allowing the programmer to decompose computation into tasks that can be created dynamically at run time.

The programmer does not need to be concerned with dividing the computation evenly among a fixed number of threads. Instead, TBB promotes the practice of breaking down the computation into tasks of appropriate granularity.

The tasks are submitted to a task pool and the Task Scheduler dispatches the tasks to the hardware units in an efficient way. Moreover, TBB implements a task stealing mechanism that takes into account task data locality and system's workload, making it suitable for applications having tasks with unpredictable execution time and thus very difficult to balance manually. Both functional and data parallelism are supported by TBB.

3.6. Parallelization approaches

Intel Cilk Plus

Intel Cilk Plus extends C and C++ to enable writing composable deterministic parallel software that can exploit both the thread and vector parallelism commonly available in modern hardware. Cilk Plus breaks the algorithm into independent computations (tasks) and handle allocation of tasks to threads during run-time to ensure efficient execution. The explicit vector language makes it easier to exploit SIMD instructions efficiently.

It maintains a stack with the remaining workflow, employing a work stealing heuristic similar to the prior described Intel TBB.

3.6.2 Distributed Memory Parallelism

Compute Unified Device Architecture (CUDA)

Based on the CUDA architecture, NVIDIA GPUs with the new Tesla unified graphics and computing architecture run CUDA C/C++ programs and are widely available in laptops, PCs, workstations, and servers.

CUDA provides three key abstractions – a hierarchy of thread groups, shared memories, and barrier synchronization, that provide a clear parallel structure to conventional C code for one thread of the hierarchy.

Multiple levels of threads, memory, and synchronization provide data parallelism nested with thread and task parallelism.

Message Passing Interface (MPI)

The Message Passing Interface (MPI) provides a simple API for parallel programming in distributed memory environments, with the main goals being high performance, scalability, and portability. Both point-to-point and collective communication are supported.

Most MPI implementations consist of a specific set of routines directly callable from C, C++, Fortran. The data must be explicitly split and passed among the processes by the programmer.

Intel developed an MPI Library for the Intel MIC Architecture, adding the possibility for the node to be either a host processor or coprocessor.

A combination of MPI and OpenMP is regarded as a suitable programming model for work sharing between computing nodes – commonly referred as hybrid parallel programming. Developers can employ MPI to communicate between nodes and OpenMP for parallelization within the computing node.

3.6. Parallelization approaches

3.6.3 Use of coprocessors as accelerators

Aside from vectorization, being limited by memory bandwidth on processors can indicate an opportunity to improve performance with a coprocessor. For this to be most efficient, an application needs to exhibit good locality of reference and utilise caches well in its core computations.

Two additional considerations arise from the coprocessors being physically separated from the processors and on a PCIe bus. The first being the need to fit problems or subproblems into the more limited memory on the coprocessor card, and the other is the overhead of data transfers that favour minimization of communication and synchronization aiming to persist as much data as possible on the target device and only synchronise data when strictly necessary.

Techniques to load balance work across all the cores available should be considered.

4

THE PRODUCT

4.1 USE CASE ANALYSIS

4.1.1 *Image management*

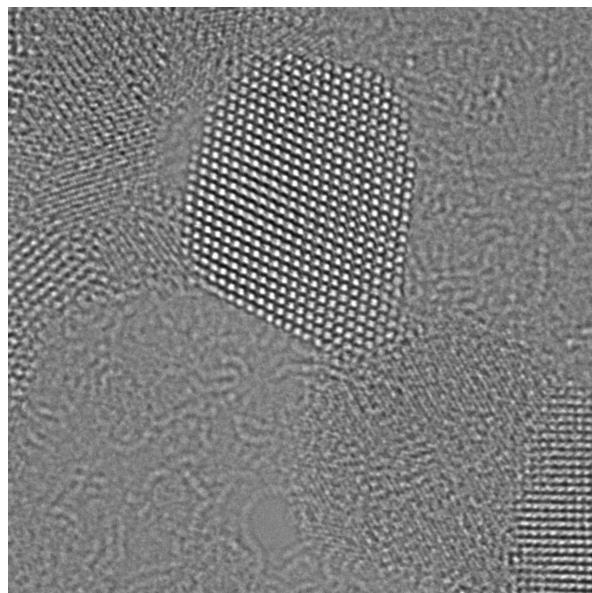


Figure 4: Im2Model sample 2D-TEM experimental image input (image of a CeO₂ crystal).

Im2Model heavily relies on image processing algorithms, which tend to be computationally heavy. Therefore, data access and data replication directly influence the algorithms efficiency.

Image management will be carried out by recurring to the OpenCV class **Mat**. Mat has two data parts: (i) the matrix header, containing control information such as the size of the matrix, the method used for storing, the memory address in which the matrix is stored, etc, and a (ii) array containing the pixel values. The matrix header size is constant. The array containing the pixel values varies in size from image to image.

4.1. Use case analysis

The reference counting system of OpenCV, proves to be ideal for our specific image data transformations. For example, to create a region of interest (ROI) in an experimental image we just create a new header with the new boundaries, referring to only a subsection of the full experimental data, reducing the number of unnecessary copies of potentially large images.

Each Mat object has its own header, however the image data may be shared between two or more instances by having their matrix pointers point to the same memory address.

Figure 4 illustrates an example Im2Model 2D-TEM experimental image input.

4.1.2 *Image simulation*

In order to assess the workflow listed in the chapter 2.2, a prototype implementation of the TEM image simulation recurring to Dr. Probe command-line software tools (?) was developed. A wrapper class was produced for each of Dr. Probe command-line tools:

- Task 1 – **celslc** – responsible for the slices generation based on atomic structure data that can be supplied either in form of a CEL file or a CIF file. The orientation of the atomic structure, namely the zone axis and upward vector orientations and supercell size are also an input parameter.

A preliminary crystallographic analysis by Im2Cr (?) of the experimental TEM image is necessary. In order to automate the slice generation process an configuration file manager was produced for the given tool.

- Task 2 – **msa** – responsible for performing the multislice algorithm for a chosen input wave function and a given set of object transmission functions. The tool is controlled via a parameter list provided in a text file. In order to automate the electron wavefunctions calculation process an configuration file manager was produced for the given tool.
- Task 3 – **wavimg** – responsible for calculating image intensity distributions from electron wave functions for a given set of TEM imaging parameter. The tool is controlled via a parameter list provided in a text file, and requires a complex valued wave function as input data, produced by the prior multislice algorithm. In order to automate the image intensity distributions calculation process an configuration file manager was produced for the given tool.

All the above command-line tools were pre-compiled for 64-bit Unix/OSX/Microsoft Windows operating systems for single-thread calculations, and therefore are not accountable for direct application performance improvements. The described command-line tools source code is currently inaccessible.

4.1. Use case analysis

Given the described task chain for HRTEM image simulation, an example CeO₂ (cerium oxide) nanostructure TEM simulated image was produced and it is presented on figure 5.

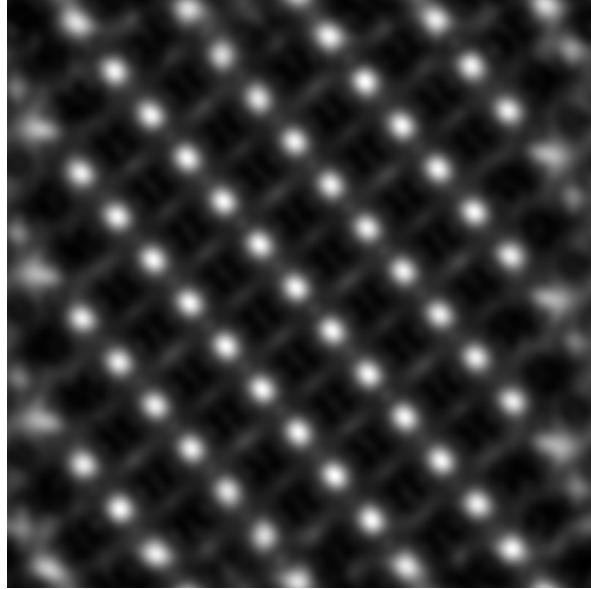


Figure 5: Example CeO₂ (cerium oxide) nanostructure simulated image.

Miller Indices (hkl): [-2 -1 -1].

Coordinates uvw: [-1 2.461462 -0.461462].

Supercell size abc: 2 2 16.

Ht: 200 kV.

C_s: -17000.

C_d: -8.5.

Thickness: 6.18 nm.

Defocus: -10 nm.

4.1.3 Further simulated image transformations

The objective apertures, the spatial coherence of the beam and its wavevelenght have effect on border aberrations. Those aberrations may appear particularly remarked in the edges of the unit-cell, therefore compromising the achievable best correlation between the 2D-TEM experimental image and the simulated images.

A supercell model representing the hole nanostructure should only posses border aberrations on the outer shape and not on every unit cell. Those aberrations should therefore be eliminated by ignoring sufficient outer pixels from the simulated TEM images.

In order to proceed to the comparison with the experimental image, both experimental and simulated image must possess the same sampling rate. Each pixel must represent the same physical dimension expressed in nanometres per pixel (nm/px).

4.1. Use case analysis

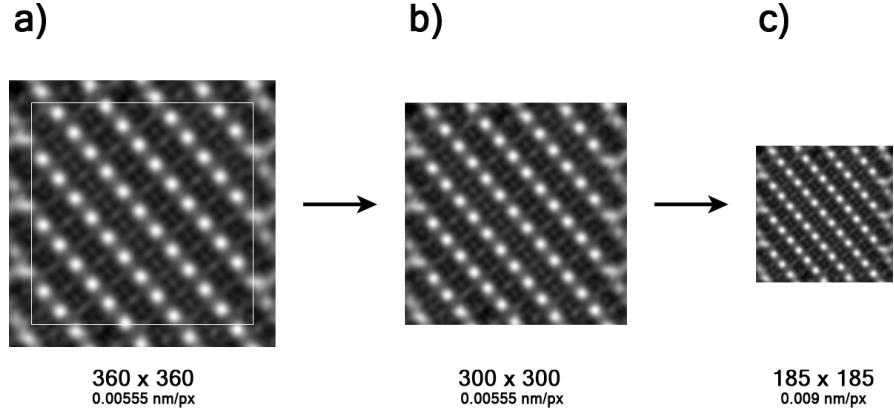


Figure 6: Border aberrations of HRTEM simulated images.

- a) Simulated image of an CeO₂ (cerium oxide) crystal.
- b) Cropped simulated image of an CeO₂ (cerium oxide) crystal, given an crop size of 30 pixels.
- c) Final simulated image of an CeO₂ (cerium oxide) crystal, following the crop and reshape operations.

Figure 6 exemplifies the previously enumerated obstacles when there is a need to correlate simulated and experimental images. The simulated image possesses border aberrations, and both simulated and experimental images have different sampling rates (0.009 nm/px vs 0.0055 nm/px).

The initial simulated image, visible in figure 6 a), will be cropped by 30 pixels in every border directions. Given the sampling rate mismatch the already cropped simulated image, visible in figure 6 b), will be reshaped by a factor of 0.617284. The final simulated image, visible in figure 6 c), presents a size of [185,185] pixels.

4.1.4 T-focus map

Both defocus and sample thickness have a significant effect on the image intensity distribution. In order to estimate the best fit to the experimental image of both parameters, several step-wise variations should be computed. Therefore, given an initial estimate of both the defocus and thickness variation ranges, and the desired sample number in order to discretize each continuous range interval, a 2D variation map is created. An example T-focus map of a CeO₂ crystal is presented in figure 7 b).

Visual comparison between the give experimental image ROI and each of the simulated images present on the T-focus map can now be executed. To evaluate that comparison between the images we produced a 2-step correlation algorithm, with the first being a patch-based translation alignment recurring to a correlation coefficient, and the second step

4.1. Use case analysis

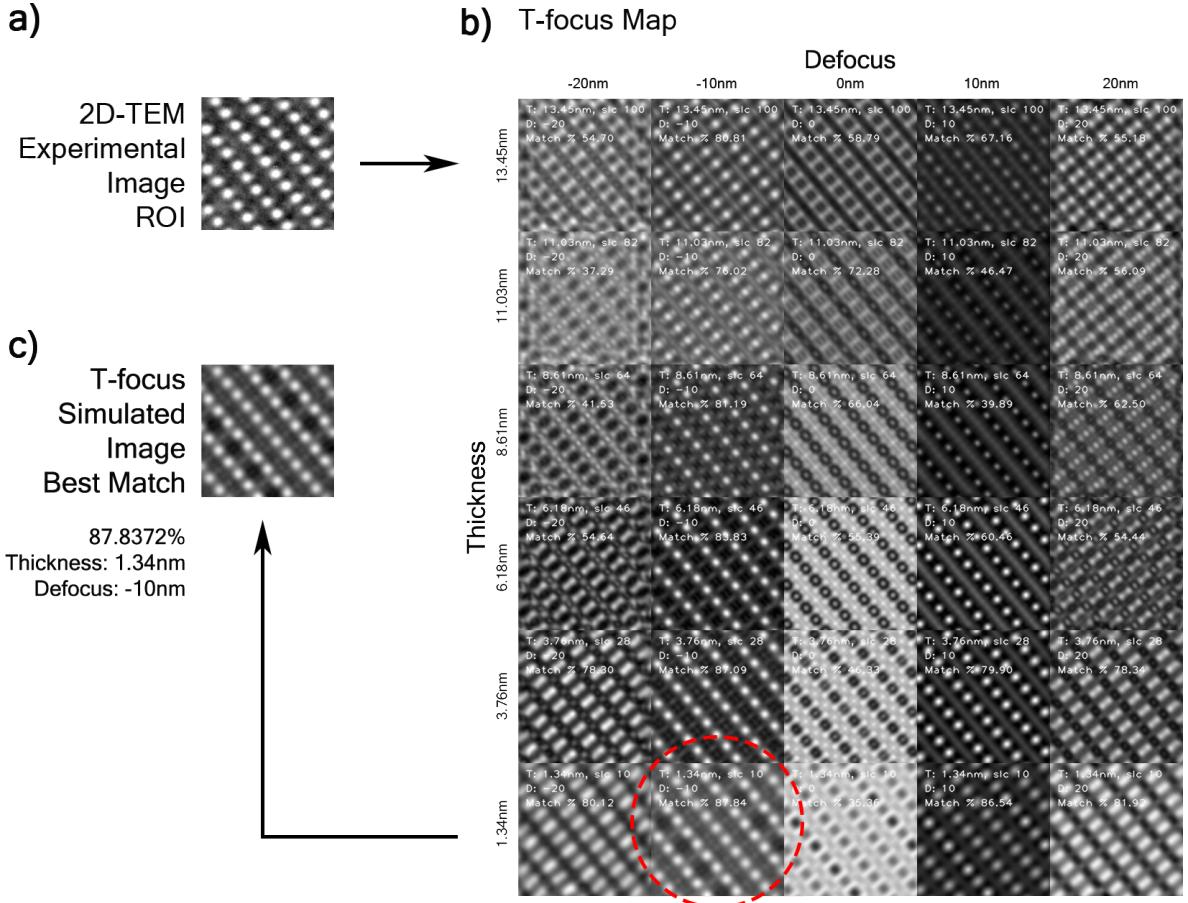


Figure 7: T-focus map of a CeO₂ crystal.

- a) Selected ROI from the 2D-TEM image.
- b) T-focus Map of a CeO₂ (cerium oxide) crystal.

Thickness variation interval (nm): [1.34,13.45], distinct thickness samples samples: 6.

Defocus variation interval (nm) [-20,20], distinct thickness samples samples: 5.

Other experimental setup parameters: Miller Indices (hkl): [-2 -1 -1], Coordinates uvw: [-1 2.461462 -0.461462], Supercell size abc 2 2 16, Ht 200 kV, C_s: -17000, C_d: -8.5.

- c) Thickness and Defocus best fit to the experimental image.

being an incremental refinement for the best given match in step 1, by enabling an euclidean translation transformation on that image.

Since Im2Cr (?) crystallographic information may contain an certain amount of error regarding the angle differences between experimental and simulated images, the seconds step enables refined transformations in scale, angle and position. We can therefore confirm both Im2Cr (?) inputs and user inputs regarding the sampling rate of the experimental image.

Figure 8 demonstrates the 2D planar transformations associated to the 2-step correlation algorithm. In the red dashed rectangle we can observe the search domain of the To patch

4.1. Use case analysis

image (simulated image) in I₁ (experimental image). Step 1 is performer for all simulated images calculate in T-focus Map.

After the patch-based translation performed in step 1, we can now visualise the corresponding best match in the Experimental Image ROI (dashed yellow rectangle). Since minor errors may occur, due to either user input or Im2Cr (?) input, the euclidean translation transformation enables further refinements in the correlation process.

Figure 8 b') and c') are the corresponding experimental and simulated images in which we should base the next steps of Im2Model.

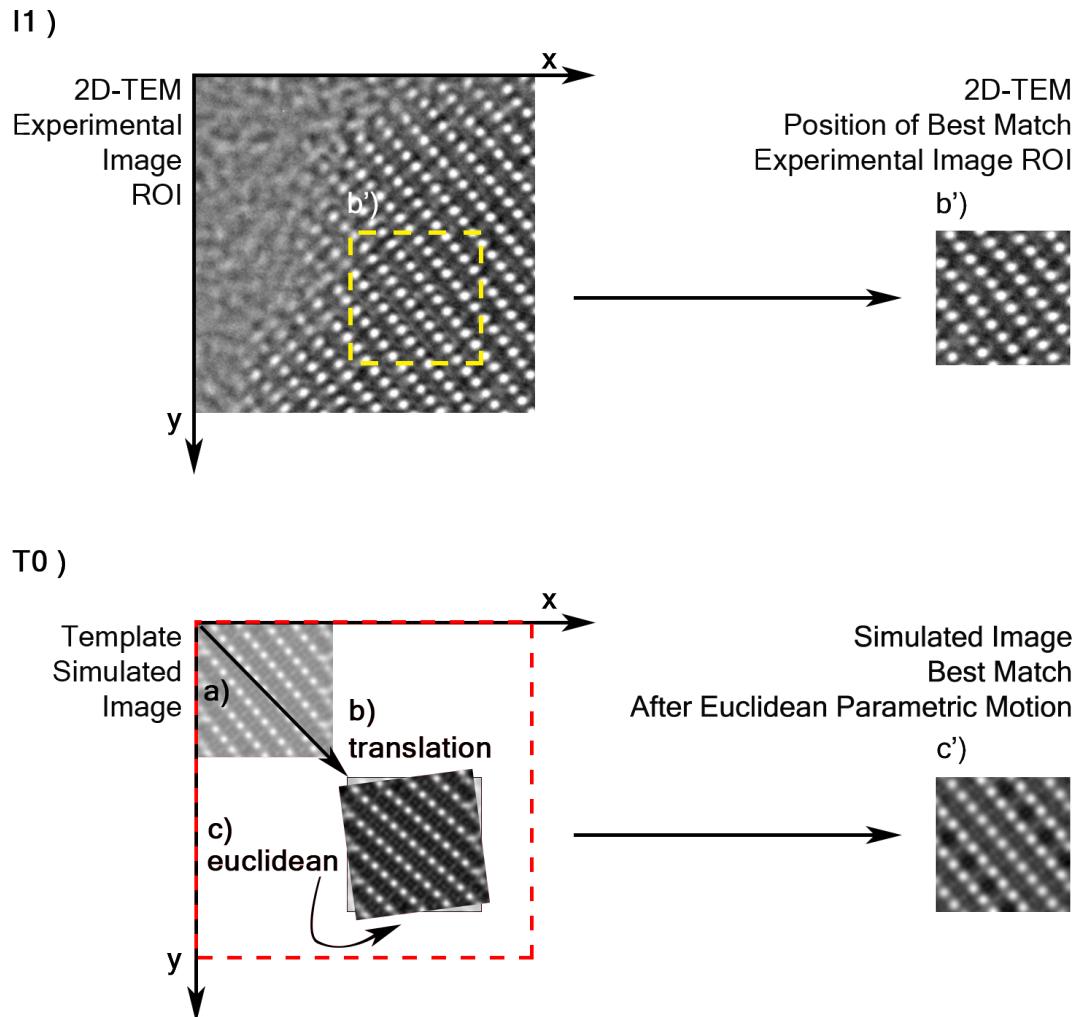


Figure 8: 2D planar transformations present on the 2-step image correlation algorithm.

- a) Template Simulated Image initial position.
- b) Template Simulated Image after patch-based translation alignment.
- b') Position of Best Match in the Experimental Image ROI.
- c) Template Simulated Image after euclidean translation transformation.
- c') Simulated Image Best Match after euclidean parametric motion.

4.2. Describing the problem

Given the best match calculated by the T-focus map, we can now have an estimation of the local thickness and defocus values for the given experimental ROI.

4.1.5 Supercell estimation

Outline of the X-Y crystal shape

A typical way to compute edges in an image is to find local variations in intensity levels. Recurring to the Canny edge detector algorithm (?), a well known technique to find edges, we are capable of computing these derivatives and return an image with the approximated boundaries.

An area estimation of the X-Y crystal shape is also computed. Figure 9 illustrates the Canny edge detector algorithm of an CeO₂ (cerium oxide) crystal.

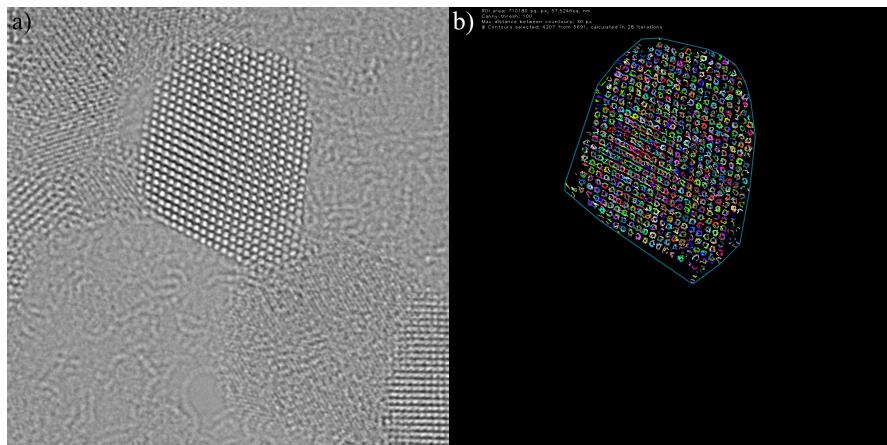


Figure 9: Edge detection of the experimental image ROI algorithm visualization.

4.2 DESCRIBING THE PROBLEM

Following the state of the art presented in chapter 2.2, this dissertation aims to apply the concepts presented and proven in the previous sections regarding the construction of atomic structures based on high resolution TEM images interpretation, and simulation of optical parameters correlating with the experimentally acquired image.

The Im2Model application should be structure trough the following steps:

- **T-focus Map** – Validation of the experimental setup parameters informed by the user and the preliminary crystallographic analysis by Im2Cr (?) of TEM images through image simulations of a small segment and its comparison with the experimental image, with the optical parameters varied stepwise in such a way that the image with

4.3. Component architecture

the best fit to the experimental image will estimate the local sample thickness and defocus.

This procedure is often referred as thickness-defocus maps in TEM simulation, and will be carried out using Dr. Probe (?) command-line software tools, composed of three tools, each related to one of the three Dr. Probe simulation steps. Wrappers for the target operating systems will be created.

The best fit to the experimental image will be carried out by recurring to the free Open Source Computer Vision Library (OpenCV), with two step image correlation algorithms and their respective suitable error metric, the first being a patch-based translation alignment with the similarity metric being the normalized correlation coefficient, and the second being an more sophisticated euclidean parametric motion with the same similarity metric as the first correlation step.

- **Supercell estimation** – A super cell model representing the whole structure will be built based on the crystallographic structure retrieved by Im2Cr (?) and the estimated dimensions of a certain region of interest. At a first step an outline of the X-Y crystal shape will be produced based on the ROI. A rough estimation of the 3D nanostructure will be achieved after this workflow step.
- **Supercell refinement** – Iterative steps of image simulation, images comparison and model refinement will be carried out until a convergence threshold. Input parameters will also be refined and validated.

During these steps, the atomic model and the TEM experimental parameters will be refined, leading to a complete description of the sample structure. We should have an atom by atom description of the nanostructure.

4.3 COMPONENT ARCHITECTURE

4.3.1 Dependency analysis

5

EFFICIENT IMPLEMENTATION OF IM₂MODEL WORKFLOW

5.1 DESCRIBING THE FINAL GOAL

Im₂Model main objective is to provide a time-efficient and unbiased work-flow for the quantitative interpretation of high-resolution TEM images, by the use of efficient computation methods, enhancing the productivity on materials characterisation at the atomic scale. The resource utilisation efficiency for a given application workload can be considered as the secondary objective.

In order assess Im₂Model efficiency on the prior metris a theoretical performance model should be developed. Based on it we could impose theoretical performance roofs, explain experimental results and provide direction for the next software improvement step.

5.1.1 *Cross functional requirements*

Response time / latency Availability Durability of data

5.1.2 *Modelling the bottlenecks*

Track inbound response time at a bare minimum.

Once you've done that, follow with error rates and then start working on application-level metrics.

Track the health of all downstream responses, at a bare minimum including the response time of downstream calls, and at best tracking error rates.

Standardize on how and where metrics are collected.

Monitor the underlying operating system so you can track down rogue processes and do capacity planning.

5.2. Modelling the solution

5.2 MODELLING THE SOLUTION

5.2.1 Designing the data model

Storage and retrieval

Distributed data

5.2.2 Designing the application model

Designing for testability

5.3 THE TROUBLE WITH THE MODELLED SOLUTION

Unreliability

Designing for resiliency and self-healing

By rebuilding an application as a network of microservices, each single point of failure is naturally easier to make resilient against machine or network failure by adding redundancy. Each service can have multiple instances running, and calls to them can be distributed between running versions. A key motivation for switching to microservices in the first place is to provide such runtime resilience.

Distributed data

6

THE TEST BOUNDARY

6.1 SERVICE TESTS

Service tests are designed to bypass the user interface and test services directly. In the previous architecture, we might just be testing a collection of classes that provide a service to the UI. For a system comprising a number of services, a service test would test an individual services capabilities.

The reason we want to test a single service by itself is to improve the isolation of the test to make finding and fixing problems faster, while also identifying the first bottlenecks of each service. To achieve this isolation, we need to stub out all external collaborators so only the service itself is in scope, as FigureXXXXXXX shows.

6.1.1 *Implementing Service tests*

Our service tests want to test a slice of functionality across the whole service, but to isolate ourselves from other services we need to find some way to stub out any downstream services.

When using a mock, I actually go further and make sure the call was made. If the expected call is not made, the test fails. Implementing this approach requires more smarts in the fake collaborators that we create, and if overused can cause tests to become brittle. As noted, however, a stub doesn't care if it is called 0, 1, or many times. Sometimes, though, mocks can be very useful to ensure that the expected side effects happen. For example, I might want to check that when I create a customer, a new points balance is set up for that customer. The balance between stubbing and mocking calls is a delicate one, and is just as

6.2 PERFORMANCE TESTS

When decomposing systems into smaller bounded contests, running on different components in different processes and spread across multiple machines we increase the number

6.2. Performance Tests

of calls that will be made across network boundaries. Where previously an operation might have involved one database call, it may now involve three or four calls across network boundaries to other services, with a matching number of database calls. All of this can decrease the speed at which our systems operate.

Tracking down sources of latency is especially important. When you have a call chain of multiple synchronous calls, if any part of the chain starts acting slowly, everything is affected, potentially leading to a significant impact. This makes having some way to performance test your applications even more important than it might be with a more monolithic system.

To generate worthwhile results, we've run extensive scenarios with gradually increasing numbers of simulated payloads. This allows to see how latency of calls varies with increasing load.

6.2.1

6.2.2

7

CONCLUSION

Im2Model has for main objective to provide a time-efficient and unbiased workflow for the quantitative interpretation of HRTEM images. The major challenge regarding this pre-dissertation work frame was perceiving the essential nanomaterials theory in order to fully understand the challenges and efforts towards the main goal completion. This process was longstanding and it was only managed to do it with help of both advisors.

Concerning the performance modelling, several potential exploratory paths can be taken, leaving the essential performance-related decisions to post Im2Model software requirements completion.

The Im2Model workflow has been fully characterised, and it is now time to complete the remaining steps and to "think towards efficiency and parallel".

7.1 THE FUTURE OF IM2MODEL

7.1.1 *Integrations*

7.1.2 *Aiming for observability*

7.1.3 *Trusting, but verifying*

7.1.4 *Privacy, Tracking, and Compliance*

