User Transparent Parallel Image Processing
Seinstra, F.J.

Citation for published version (APA):

General rights
It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations
If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: http://uba.uva.nl/en/contact, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.
Contents

1 Introduction ............................... 1
   1.1 The Need for Speed in Image Processing ................. 2
   1.2 The Gap Between Computing and Imaging ................. 3
   1.3 Thesis Outline .......................... 4

2 A Sequential Programming Model for Efficient Parallel Image Processing 5
   2.1 High Performance Computing Architectures .............. 7
   2.2 Software Development Tools .......................... 9
      2.2.1 General Purpose Parallelization Tools .............. 10
      2.2.2 Tools for Parallel Image Processing ............... 13
      2.2.3 Discussion .................................. 16
   2.3 A Sustainable Software Architecture for User Transparent Parallel Image Processing 18
      2.3.1 Architecture Requirements ......................... 18
      2.3.2 Architecture Overview ......................... 19
   2.4 Conclusions .................................. 23

3 Parallelizable Patterns in Low Level Image Processing Algorithms 25
   3.1 Algorithmic Patterns: The Horus Approach .............. 27
   3.2 Integration of Parallelism in Horus .................... 29
   3.3 Data Parallel Image Processing ........................ 31
      3.3.1 Data Parallelism versus Task Parallelism ........... 32
      3.3.2 Representation of Digital Images ................. 33
   3.4 Parallelizable Patterns ............................ 34
      3.4.1 Generic Description ............................ 35
      3.4.2 Default Parallelization Strategy .................. 37
      3.4.3 Example 1: Parallel Reduction .................... 37
      3.4.4 Example 2: Parallel Generalized Convolution ...... 39
      3.4.5 Discussion ................................. 40
   3.5 Conclusions and Future Work ......................... 41
4 Semi-Empirical Modeling of Parallel Low Level Image Processing Operations 43
   4.1 Computer System Performance Estimation 45
      4.1.1 Estimation Technique: Requirements 45
      4.1.2 Estimation Techniques in the Literature 46
   4.2 Semi-Empirical Modeling 49
   4.3 Abstract Parallel Image Processing Machine 49
      4.3.1 APIPM Components 50
      4.3.2 APIPM Instruction Set 51
      4.3.3 Discussion 52
      4.3.4 Related Work 53
   4.4 APIPM-Based Performance Models 54
      4.4.1 Extended Model for Point-to-Point Communication 56
      4.4.2 Discussion 56
   4.5 Measurements and Validation 57
      4.5.1 Detection of Curvilinear Structures 57
      4.5.2 Parallel Execution 59
      4.5.3 Performance Evaluation 60
   4.6 Conclusions 63
   4.A APIPM Instruction Set Definition 64
   4.B APIPM Model Parameterization 69

5 A Communication Model for Automatic Decomposition of Regular Domain Problems 73
   5.1 Modeling of Message Passing Programs 75
      5.1.1 Model Requirements 75
      5.1.2 Relevant Models in the Literature 77
   5.2 The P-3PC Model 79
      5.2.1 Part I: 3PC 79
      5.2.2 3PC versus LogGP 81
      5.2.3 Part II: P-3PC 81
   5.3 Application of the P-3PC Model 82
   5.4 Measurements and Validation 85
      5.4.1 Distributed ASCI Supercomputer (DAS) 86
      5.4.2 Beowulf at SARA 90
   5.5 Conclusions 90

6 A Finite State Machine for Global Optimization of Application Performance 93
   6.1 The Performance Optimization Problem 94
      6.1.1 Abstract Function Specifications 95
      6.1.2 Default Algorithm Expansion 96
      6.1.3 Inefficiencies from Default Algorithm Expansion 97
   6.2 Finite State Machine Definition 98
      6.2.1 States and Lifespan of (Distributed) Data Structures 99