Distributed Anisotropic Diffusion

Release 0.00

Kevin H. Hobbs[1]

December 15, 2010

[1]hobbsk@ohiou.edu
Biological Sciences
Ohio University
Athens Ohio

Abstract

Distributed anisotropic diffusion provides a wrapper program around ITK's anisotropic diffusion filters that allows an input image to be spread across the memory of several computers and the processors of all of the computers to work simultaneously on the output. Distributed anisotropic diffusion allows the Visible Woman Head dataset to be smoothed with 100 iterations of the vector gradient magnitude anisotropic diffusion filter in 47 minutes on an 8 node 64 core cluster versus 53 minutes for just 10 iterations of standard vector gradient magnitude anisotropic diffusion on a single node.

Contents

1 Introduction

Anisotropic diffusion removes unwanted detail or noise from an image while preserving sharp transitions from one intensity or color to another. Because anisotropic diffusion uses an iterative process to move the value of each pixel closer to the values of its neighbors, the input image can not simply be broken into pieces and filtered independently on several computers without including large numbers of padding pixels in each piece. Furthermore, since anisotropic diffusion uses in each iteration the average gradient magnitude to determine which pixels values to smooth, all the pieces would require a preset parameter to replace the average gradient magnitude. Distributed anisotropic diffusion uses the Message Passing Interface (MPI) to handle communication of a single pixel boundary around each piece between MPI processes and to update the average gradient magnitude.

2 Details of Distributed Anisotropic Diffusion

This Section describes the implementation of distributed anisotropic diffusion using ITK and MPI. It lists the information the program expects to be provided on the command line. It briefly describes the image pipeline. It describes how the interfaces between the image pieces are passed between the MPI processes.

2.1 Command Line Arguments

The MPI anisotropic diffusion program takes five parameters from the command line :

- the input file name (should support streamed reading)
- the output file name (must support streamed writing)
- the conductance parameter
- the number of iterations
- the time step

2.2 Image Pipeline

MPI anisotropic diffusion uses a looping pipeline (Figure 1). Each MPI process is responsible for one piece of the total image. Each MPI process reads its piece of the input image along with one extra pixel of padding in every direction. This padded piece is the ”Working Image” in Figure 1. Each MPI process calculates the gradient magnitude of the padded image and sends the sum of all of the pixel values within the region for which it is responsible to the first MPI process. Then the first MPI process returns the average gradient magnitude to every MPI process. The average gradient magnitude is used in a single iteration of anisotropic diffusion of the working image. Each MPI process sends the edges from just within the region for which it is responsible of its diffused image to the MPI processes responsible for the neighboring pieces and receives edges to replace the padding around its piece from them (Figure 2). The updated image replaces the ”Working Image” and the process is repeated. Finally after all iterations are complete, each MPI process sends the region of the working image for which it is responsible to the first MPI process which pastes it into the input image and writes the piece to the output file.

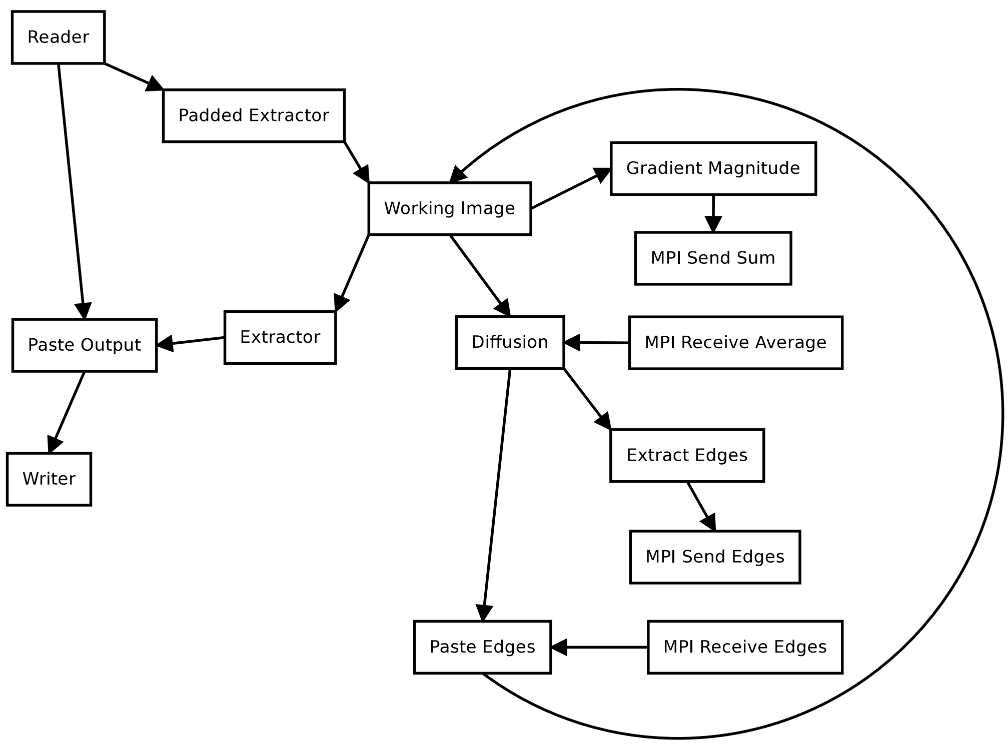

Figure 1: The looping pipeline of distributed anisotropic diffusion.

2.3 Image Overlap

Each pixel in the image piece is influenced by the pixels surrounding it during each iteration of anisotropic diffusion. To process the pixels on the edges of the piece correctly one extra pixel is added to both directions in every dimension. After each iteration of anisotropic diffusion the values of these pixels must be replaced with values from the neighboring pieces. Each MPI process sends the values on the edges of its unpadded image piece to replace the padding pixel values around the neighboring image piece. Each MPI process also receives the pixel values from the edge of the neighboring unpadded image piece to replace the padding pixel values around its image piece (Figure 2).

3 Output

Figure 3 shows one slice of an input image which was processed using the ITK vector gradient magnitude anisotropic diffusion filter inside of the distributed anisotropic diffusion program. The program took 47 minutes to process the image. The computer was an eight node cluster. Each node had two Intel Xeon X5550 2.67GHz processors with four cores each and 12 GB of RAM. The nodes were connected with gigabit Ethernet. One MPI process was started on each node with ITK's multi-threading using all processor cores. The parameters were conductance parameter = 0.2, iterations = 100, and time step = 0.015. Figure 4 shows a slice of the output image.

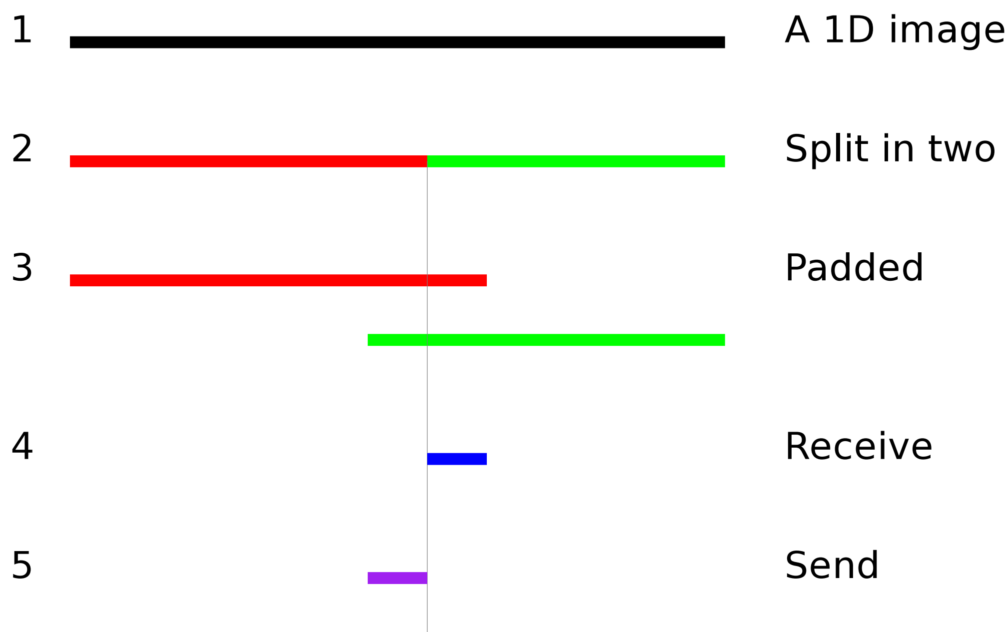

Figure 2: (1) A one dimensional image is (2) split into two regions, one red and one green. (3) The red and green regions are padded by one pixel. (4) The red region receives the blue pixel from the green region. (5) The red region sends the purple pixel to the green region.

4 Scaling

When run on 50, 100, and 200 slice slabs of the Visible Woman Head data set with conductance parameter = 0.2, iterations = 10, and time step = 0.015 standard vector gradient magnitude anisotropic diffusion always took more time than the distributed version (Figure 5). Adding more than 4 nodes did not greatly decrease processing time until the 200 slice data set.

5 Software Used

This program was built using :

- Insight Toolkit (GIT)
- CMake (GIT)
- MPI (openmpi-1.4.3)

6 Acknowledgements

This work was done in Scott L. Hooper's lab at Ohio University hooper@ohio.edu.

The Visible Woman Head data set used in Figs. 3, 4, and the piece used in the tests that accompany this paper was downloaded from `http://public.kitware.com/pub/itk/Data/VisibleWomanHead/RawRGB/`

This work was funded by NSF Award # IOS-0958926 and NIH Award # ARRA-RC1DA028494.

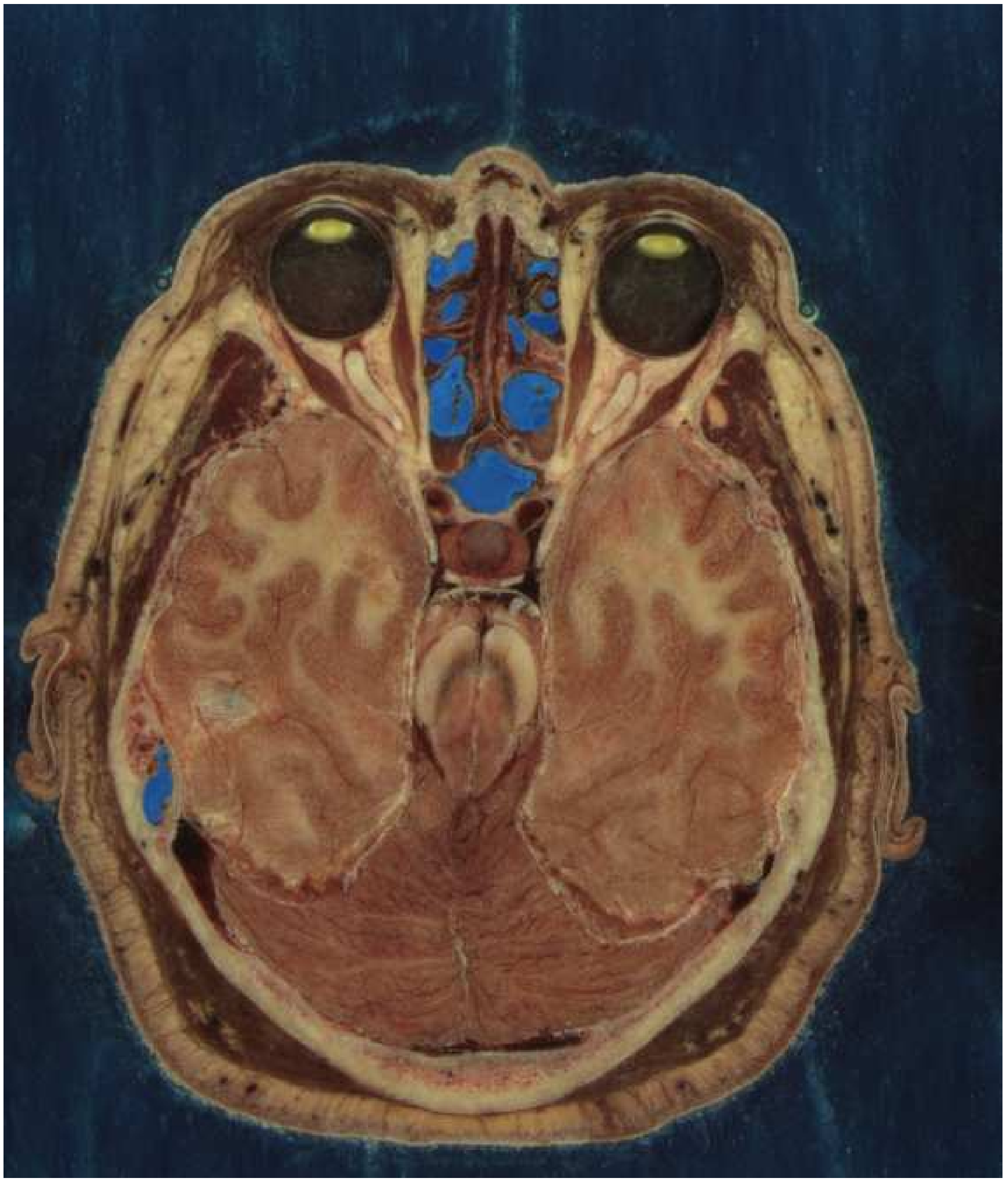

Figure 3: A slice of the input image.

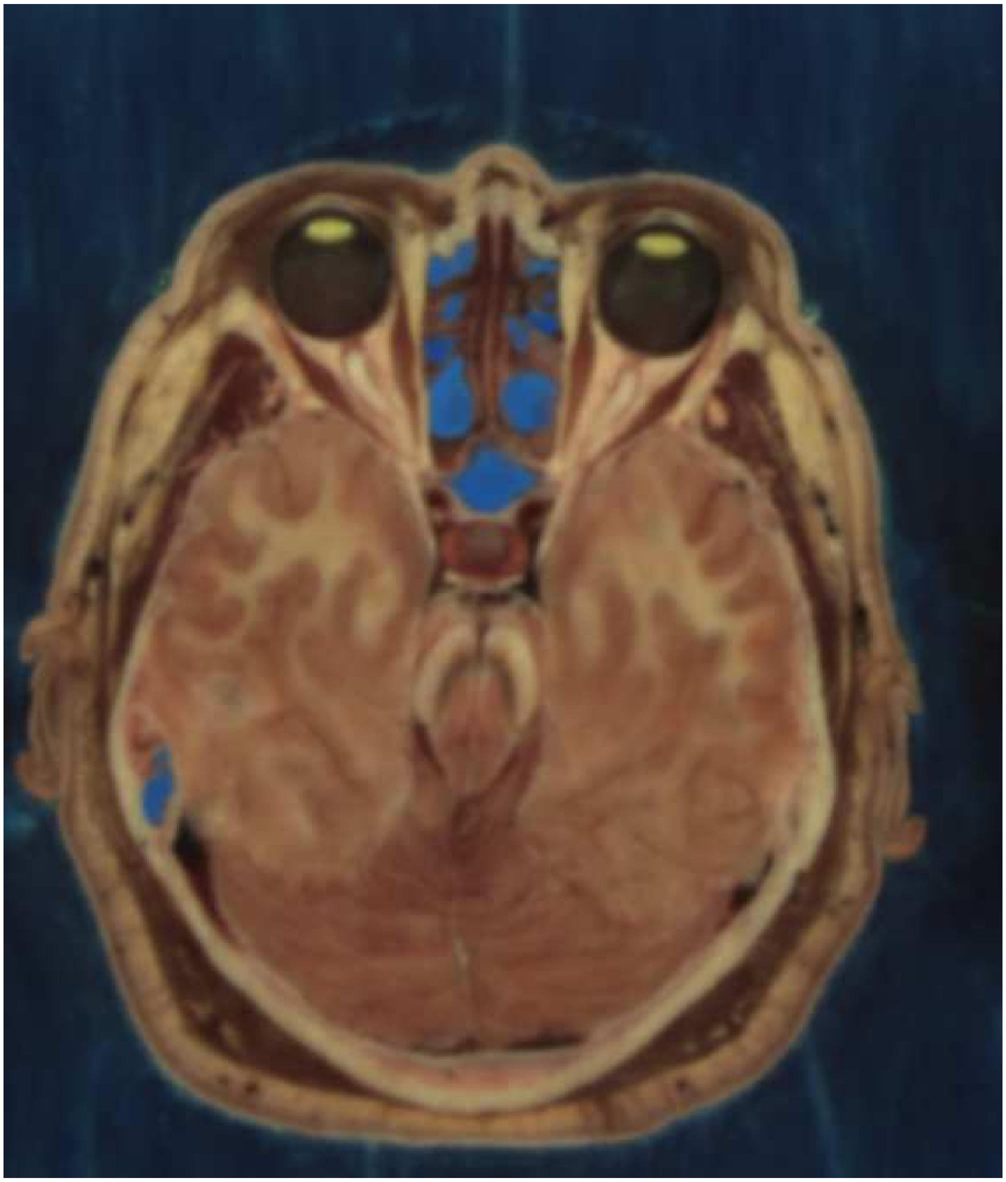

Figure 4: A slice of the output image.

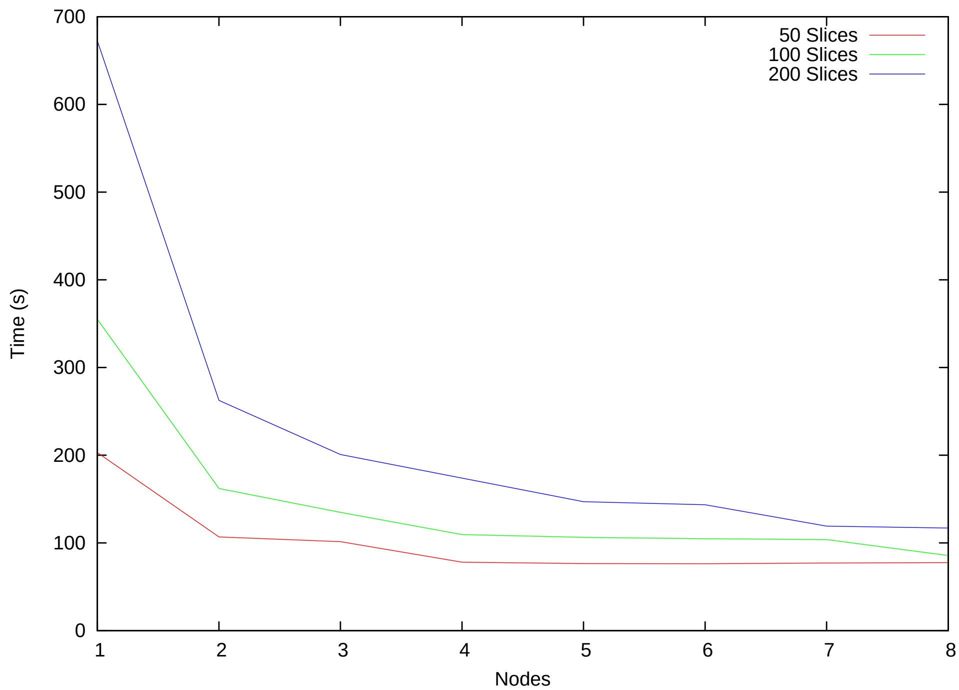

Figure 5: Processing times with three different data sizes for 10 iterations of standard (1 node) and distributed (≥ 2 nodes) vector gradient magnitude anisotropic diffusion