Final Assignment
Mastermath Course Parallel Algorithms 2024

Diffusion filtering

In order to reduce the noise in an image one can use a technique called diffusion filtering. This technique makes the image smoother by considering it as the solution of a diffusion-reaction equation.

The diffusion-reaction equation is given by:

$$\Delta\phi + \lambda(\phi_0 - \phi) = 0 \quad \in \Omega. \tag{1}$$

Here ϕ represents the filtered image, ϕ_0 the original image, and Δ the Laplace operator. The Laplace operator diffuses the pixel values, and thereby makes the images smoother. This is analogous to diffusion of heat. The so-called fidelity parameter $\lambda \geq 0$ determines how much confidence we have in the original image. For the domain Ω we can take the square $(x, y) \in (0, n_x) \times (0, n_y)$, in which n_x and n_y are the number of pixels in x and y direction.

On the boundary Γ we take the homogeneous Neumann condition

$$\frac{\partial\phi}{\partial\mathbf{n}} = 0 \text{ on } \Gamma, \tag{2}$$

which imposes that there is no diffusion (”heat transfer”) through the boundary. In this equation $\mathbf{n}$ is the outward normal vector.

To discretise the system we define a grid with the nodes in the centers of the pixels. The discretisation of the Laplace operator is done using equation 4.59 in the book. The Neumann boundary condition is a reflection condion (or symmetry condition). In order to implement this we extend the grid with one row/column of gridpoints on every side of the domain and we impose that the pixel values outside the domain are equal to the values in the neighbouring pixel inside the domain. The resulting linear system is given by

$$(-A + \lambda I)\phi = \lambda\phi_0$$

in which A is the Lapacian matrix for the Neumann problem), I the identity operator, ϕ the vector with pixel values of the filtered image and ϕ_0 the vector with pixel values of the original image.

We solve this system with the Conjugate Gradient method. The CG algorithm is given for example as Algorithm 4.9 in the book.

Sequential program

The program `phantom` implements the denoising algorithm described in the previous section. It consists of the following steps:

- The Shepp-Logan phantom image is generated to test the denoising algorithm. The Shepp-Logan phantom is a classical image for testing medical applications. It (vaguely) resembles a brain. You can choose the size of the image with the commandline option `-npx`. For example, `./phantom -npx 100`, which will result in a Shepp-Logan image of 100×100 pixels. Default: npx= 512 pixels.

- Noise is added to the image with standard deviation σ. For example `./phantom -sigma 0.5` generates a noisy image with noise with a standard deviation $\sigma = 0.5$. Default $\sigma = 0.$: no noise is added.

- The noisy image is filtered using the diffusion based denoising method outlined above. The fidelity parameter λ can be specified using the option `-lambda`. Default $\lambda = 0$: no noise filtering. The standard deviation of the noise in the filtered image is given as output.

- Three image are written to file in gif-format: phantom.gif, noisy.gif, and filtered.gif.

Example: `./phantom -npx 512 -sigma 0.5 -lambda 1.`

Assignment

The assignment is to parallelise the denoising part of the program.

1. Download the Fortran code, and compile it. Run the program for different choices of the parameters σ and λ to get a feeling of their effect. Keep the image size fixed to the default value. Select two values for σ and determine corresponding values for λ that significantly reduce the noise in these images. Use these two combinations of σ and λ in further experiments.

2. The next step is to distribute the image over the processors. Generate the noise-free image on the main processor. Define a block-wise domain decomposition, with d_x blocks in x-direction and d_y blocks in y direction. Assign a domain to each processor and send the corresponding part of the image to this processor. Every processor should add noise to its part of the image. At the end of the program, all the processors have to send their part of the (noisy and filtered) image to the main processor. This processor should assemble all these parts of the image to one image and write this to file in gif format. Add this to the code. Do not apply filtering yet, test if it works correctly for the noisy image.

3. Given the block-wise domain decomposition, parallelise the CG algorithm, in particular the matrix-vector multiplication and inner products.

4. Determine the BSP cost function for the filtering part (so only the CG algorithm). Measure the BSP parameters.

5. Perform experiments for a range of values for the number of processors, image sizes, and subdomain sizes. Always motivate your experiment: what is the question you want to answer with the experiment? Investigate one question per experiment. Relate your results to the BPS-cost model.

6. The generation of the noiseless image and writing the images to file are sequential parts of the program. Analyse the program using Amdahl's law and Gustafson's law to predict the attainable speed-up of the complete program. Compare these model predictions with measurements.

Optional questions

7. Analyse your algorithm using the roofline model.

8. Use diamond-shaped subdomains.

Assessment

The assignment counts towards the final grade in three different ways:
Presentation (counts 10%), on 11 December

- On location during regular class hours.
- Stick to your time limit (10 min per presentation, including 2 min time for questions/discussion).
- If you made the assignment with a colleague you have to give a joint presentation.

Report (counts 30%), due December 20, 2024, 17.00 hour

- Length between 15-20 pages including figures, tables, and references. The upper limit is a strict 20 pages. No tricks with small fonts or small figures are allowed.
- Hand-in electronically as an assignment in the ELO system
- Be concise. Avoid verbosity.
- Submit the code with the report.
- You may work in pairs. Hand in one report per pair.

The oral exam (counts in total 60 %)

- Part of the questions during the oral exam may be related your project. The oral exam is individual.