IMAGE DEBLURRING - COMPUTATION OF CONFIDENCE INTERVALS

VICTORIA TAROUDAKI
Prof. Dianne P. O’Leary

UNIVERSITY OF MARYLAND

MAY 3, 2011
Outline

1. Introduction
2. Project Background
3. Approach
4. Implementation
5. Database
6. Validation
7. Testing
8. Schedule-Milestones
9. Deliverables
10. Summary
11. Bibliography
Blurred Images are everywhere

(a) Actual scene  (b) Blurred Image  (c) Ideal recovered image
The image

- An image is an array of pixels
- For a grayscale image, these values are in the interval \([0,255]\)
  
  \[0 = \text{black} \]
  
  \[255 = \text{white} \]
The problem

- **Notation:**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Size</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K$</td>
<td>$m \times n$</td>
<td>Matrix defined through the Point Spread function (PSF) in the case of a linear problem</td>
</tr>
<tr>
<td>$X$</td>
<td></td>
<td>Original Clear Image</td>
</tr>
<tr>
<td>$x$</td>
<td>$n \times 1$</td>
<td>Vector containing the values corresponding to the pixels of the image $X$</td>
</tr>
<tr>
<td>$B$</td>
<td></td>
<td>The blurred image we measure</td>
</tr>
<tr>
<td>$b$</td>
<td>$m \times 1$</td>
<td>Vector which contains the values of the pixels of the blurred image $B$</td>
</tr>
<tr>
<td>$e$</td>
<td>$m \times 1$</td>
<td>Noise Vector</td>
</tr>
</tbody>
</table>

- $b = Kx + e$
PSF-Blurring matrix

- The Point Spread Function (PSF) used is the Gaussian.
- The blurring matrix is block diagonal with so many diagonal blocks as the number of columns of the PSF matrix.
- Example for a $3 \times 3$ PSF with a $5 \times 5$ image:

(d) Gaussian PSF of size $3 \times 3$  
(e) Blurring matrix of an image $5 \times 5$ using a PSF $3 \times 3$
Goal

Computation of Confidence Intervals for Images blurred by

\[ b = Kx + e \]
Confidence Intervals

**Definition**

One-at-a-time confidence intervals bound each $\varphi_k$ individually with probability $\alpha$ ($\alpha\%$ confidence).

$$Pr\{l_k \leq \varphi_k \leq u_k\} = \alpha, \ k = 1, 2, \ldots, p$$

**Definition**

Simultaneous confidence intervals which bound all the $\varphi_k$ simultaneously with a probability greater than or equal to $\alpha$.

$$Pr\{l_k \leq \varphi_k \leq u_k, \ k = 1, 2, \ldots, p\} \geq \alpha$$
Assumptions- Notation

Assumptions

- Noise ($e$) normally distributed with mean zero and standard deviation $S$. ($e \sim \mathcal{N}(0, S^2)$)
- $S$ nonsingular and symmetric

Notation

- $lb_i$: The lower bound of the confidence interval corresponding to the pixel $i$
- $ub_i$: The upper bound of the confidence interval corresponding to the pixel $i$
Serial Algorithm - The image as a Whole

Algorithm

for each pixel $i$ of the image

solve $\min \{ w_k^T x : \|Kx - b\|_S \leq \mu, 0 \leq x \leq 255 \}$ for the $lb_i$

solve $\max \{ w_k^T x : \|Kx - b\|_S \leq \mu, 0 \leq x \leq 255 \}$ for the $ub_i$

end
Parallel Algorithm - The image as a Whole

Algorithm

\texttt{parfor} each pixel $i$ of the image

\begin{align*}
\text{solve } & \min \{ w_k^T x : \| Kx - b \|_S \leq \mu, 0 \leq x \leq 255 \} \text { for the } lb_i \\
\text{solve } & \max \{ w_k^T x : \| Kx - b \|_S \leq \mu, 0 \leq x \leq 255 \} \text { for the } ub_i
\end{align*}

\texttt{end}
Main problem of the full image approach

- For each one of the pixels in the image we use iterative methods for the minimization (or maximization) of the norms.
- These methods use the blurring matrix $K$ which for an $n \times n$ image is of size $n^2 \times n^2$ (e.g. for a $256 \times 256$ image, $K$ is of size $65536 \times 65536$).

To deal with this problem, we use another approach, that of the sub-images and the sub-matrices.
Construction of Sub-Images

\[ x_s \] the sub-image of desirable size
\[ x_b \] the boundary of appropriate size
Notation for the sub-problems

- **Notation:**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Size</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K_s$</td>
<td>$rc \times rc$</td>
<td>Part of the blurring matrix $K$ that corresponds to the subimage</td>
</tr>
<tr>
<td>$X_s$</td>
<td>$r \times c$</td>
<td>Original Clear Sub-Image</td>
</tr>
<tr>
<td>$x_s$</td>
<td>$rc \times 1$</td>
<td>Vector containing the values corresponding to the pixels of the sub-image $X_s$</td>
</tr>
<tr>
<td>$x_t$</td>
<td></td>
<td>The vector of values of the rest of the pixels in the image</td>
</tr>
<tr>
<td>$K_t$</td>
<td></td>
<td>Part of the blurring matrix $K$ that corresponds to $x_t$</td>
</tr>
<tr>
<td>$b_s$</td>
<td>$rc \times 1$</td>
<td>Vector which contains the values of the pixels of the blurred image $B_s$</td>
</tr>
</tbody>
</table>

- $b_s - K_t x_t = K_s x_s \leftrightarrow b_{st} = K_s x_s$
Serial Algorithm - Use of Sub-Images

Algorithm

for each sub-image $s$

    for each pixel $i$ of the sub-image

        solve $\min \{ w_k^T x_s : \| K_s x_s - b_{st} \|_S \leq \mu, 0 \leq x_s \leq 255 \}$ for the $lb_i$

        solve $\max \{ w_k^T x_s : \| K_s x_s - b_{st} \|_S \leq \mu, 0 \leq x_s \leq 255 \}$ for the $ub_i$

    end

end
Algorithm

for each sub-image $s$

\begin{align*}
\text{parfor each pixel } i \text{ of the sub-image} \\
\text{solve } & \min \{ w_k^T x_s : \|K_s x_s - b_{st}\|_S \leq \mu, 0 \leq x_s \leq 255 \} \text{ for the } lb_i \\
\text{solve } & \max \{ w_k^T x_s : \|K_s x_s - b_{st}\|_S \leq \mu, 0 \leq x_s \leq 255 \} \text{ for the } ub_i
\end{align*}

end

end
Database

Gray-scale images of various sizes.

Examples of test images:
Example of sub-images

(f) 4 $16 \times 16$ sub-images

(g) 8 $16 \times 16$ sub-images
Validation

For $\alpha\%$ confidence interval with $(0 < \alpha < 100)$ the validation is done by multiple runs of the code using the same data and values of parameters.
"Eiffel Tower 32", subimages $4 \times 4$
“Eiffel Tower 32”, subimages $8 \times 8$
Boundary

- Formula to compute the number of pixels in the boundary ($BP$) of a subimage given its size ($n \times n$) and the size of the PSF ($p \times p$) is the following

$$BP = 4 \left( n + \frac{p - 1}{2} \right) p - 1 = 2(p - 1) \left( n + \frac{p - 1}{2} \right)$$

- Table with the number of the pixels on the boundary for several subimage and PSF sizes.

<table>
<thead>
<tr>
<th>Subimage Size</th>
<th>PSF size</th>
<th>3</th>
<th>5</th>
<th>7</th>
<th>9</th>
<th>11</th>
<th>13</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>4×4</td>
<td>20</td>
<td>48</td>
<td>84</td>
<td>128</td>
<td>180</td>
<td>240</td>
</tr>
<tr>
<td>64</td>
<td>8×8</td>
<td>36</td>
<td>80</td>
<td>132</td>
<td>192</td>
<td>260</td>
<td>336</td>
</tr>
<tr>
<td>256</td>
<td>16×16</td>
<td>68</td>
<td>144</td>
<td>228</td>
<td>320</td>
<td>420</td>
<td>528</td>
</tr>
<tr>
<td>1024</td>
<td>32×32</td>
<td>132</td>
<td>272</td>
<td>420</td>
<td>576</td>
<td>740</td>
<td>912</td>
</tr>
<tr>
<td>4096</td>
<td>64×64</td>
<td>260</td>
<td>528</td>
<td>804</td>
<td>1088</td>
<td>1380</td>
<td>1680</td>
</tr>
</tbody>
</table>
(h) Histogram with subimages $4 \times 4$  (i) Histogram with subimages $8 \times 8$

**Figure:** Validation Histograms.
“stripes 8” image, fp
"Capitol 32" image, 4s
"Capitol 64" image, 4p
"La Rochelle 64" image, 4s and 8p
Size of image vs running time
Size of image vs running time \((8 \times 8)\)
Size of image vs running time (4 × 4)
Conclusions

- The parallel codes work better than the corresponding serial ones.

- The sub-images code is quicker than the full image except for the case when the codes deal with the image on the same way, i.e., when the sub-image is of the same size of the original image.
Size and number of sub-images vs running time

[Graphs showing running time vs subimage size and number of subimages for serial and parallel subimages]
Conclusions

- When we double the size of the sub-images, the number of them is divided by 4.

- With increasing size of the sub-images, the running time increases.

- Size of sub-images more dominant than their number.
Schedule-Milestones

Phase I: Serial Code for the computation of the confidence intervals

Finished

Phase II: Parallel Code for computing the confidence intervals using sub-images and sub-matrices

Finished
Deliverables

- Code
  - Full-image serial code
  - Full-image parallel code
  - Sub-images serial code
  - Sub-images parallel code

- Database images

- Validation module

- Report
Summary

- Code that computes the confidence intervals of a whole image and of sub-images
- Parallelization of the codes to improve speed
- Experiment with different types and sizes of images, different sizes of PSF functions
- Comparison of the results with respect to time

Results showed that the parallelized method of computing the confidence intervals using sub-images works the best.
Tony F. Chan and Jianhong (Jackie) Shen, "Image Processing and Analysis", SIAM, Philadelphia, 2005

Martin Hanke, James Nagy and Robert Plemmons, "Preconditioned Iterative Regularization For Ill-Posed Problems", IMA Preprint Series n 1024, 1992


Bibliography


Thank you