ℓ_1 Image Inpainting

The ℓ_1-regularized image reconstruction is an alternative method to solve the image inpainting problem:

$$\min_{x\in\mathbb{R}^{mn}} ||\Psi x||_1 \qquad \text{s.t.} \qquad ||Ax-b||_\infty \le \delta, \tag{1}$$

where $\Psi_{mn\times mn}$ transfers the image x to the frequency domain via Discrete Cosine Transformation; $A_{s\times mn}$ and $b\in\mathbb{R}^s$ are as in Total Variation Minimization, and $\delta > 0$ is the error threshold for reconstruction Ax at undamaged pixels b.

We derived the linear programming formulation of (3) as well as its dual problem. We implemented and tested the the program on sample grey-scale images. We examined the reconstruction quality and speed of the model under different sets of parameters.

LP formulation

We first reformulate (3) as a linear program by noting that

$$\begin{aligned} &\min_{x\in\mathbb{R}^{mn}} ||\Psi x||_1 && \text{s.t.} && ||Ax-b||_\infty \le \delta \\ =&\min_{x\in\mathbb{R}^{mn}} \mathbf{1}^\top|\Psi x| && \text{s.t.} && |Ax-b| \le \delta\mathbf{1} \\ =&\min_{x,t\in\mathbb{R}^{mn}} \mathbf{1}^\top t && \text{s.t.} && t\ge \pm\Psi x, \quad \pm(Ax-b)\le\delta\mathbf{1}. \end{aligned} \tag{2}$$

Here, $\mathbf{1}$ denotes the all-one vectors of appropriate sizes and $|v|$ is interpreted element-wise on vector v.

Dual problem

Next, we derive the associated dual of (2). Rewrite (2)

$$\begin{aligned} &\min_{x,t\in\mathbb{R}^{mn}} \mathbf{1}^\top t && \text{s.t.} && t\ge\pm\Psi x, \quad \pm(Ax-b)\le\delta\mathbf{1} \\ =&\min_{x,t\in\mathbb{R}^{mn}} \mathbf{1}^\top t && \text{s.t.} && \pm\Psi x + t\ge 0, \quad \pm Ax \ge -\delta\mathbf{1}\pm b \\ =&\min_{x,t\in\mathbb{R}^{mn}} \left[0^\top \,|\, \mathbf{1}^\top\right]\begin{bmatrix} x\\ t\end{bmatrix} && \text{s.t.} && \begin{bmatrix} \Psi & I_{mn} \\ -\Psi & I_{mn} \\ A & 0_{s\times mn} \\ -A & 0_{s\times mn}\end{bmatrix}\begin{bmatrix} x\\ t\end{bmatrix} \ge \begin{bmatrix} 0\\ 0\\ -\delta\mathbf{1}+b \\ -\delta\mathbf{1}-b\end{bmatrix}. \end{aligned}$$

where I_r denotes the $r\times r$ identity matrix and $0_{p\times q}$ is the $p\times q$ all-zero matrix.

The dual is then given by

$$\begin{aligned} &\max_{\substack{u,v\in\mathbb{R}^{mn}\\ y,z\in\mathbb{R}^s}} \left[0^\top \,|\, 0^\top \,|\, -\delta\mathbf{1}^\top + b^\top \,|\, -\delta\mathbf{1}^\top - b^\top\right]\begin{bmatrix} u & v & y & z\end{bmatrix}^\top \\ &\quad\text{s.t.} \quad \begin{bmatrix} \Psi^\top & -\Psi^\top & A^\top & -A^\top \\ I_{mn} & I_{mn} & 0_{mn\times s} & 0_{mn\times s}\end{bmatrix}\begin{bmatrix} u & v & y & z\end{bmatrix}^\top = \begin{bmatrix} 0\\ \mathbf{1}\end{bmatrix}, \quad u,v,y,z\ge 0 \\ =&\max_{\substack{u,v\in\mathbb{R}^{mn}\\ y,z\in\mathbb{R}^s}} b^\top(y-z) - \delta\mathbf{1}^\top(y+z) \\ &\quad\text{s.t.} \quad \Psi^\top(u-v)+A^\top(y-z)=0, \quad u+v=\mathbf{1}, \quad u,v,y,z\ge 0 \\ =&\max_{\substack{t\in\mathbb{R}^{mn}\\ r,w\in\mathbb{R}^s}} b^\top w - \delta\mathbf{1}^\top(w+r) \\ &\quad\text{s.t.} \quad \Psi^\top t + A^\top w = 0, \quad r\ge 0, \quad -1\le t\le 1 \\ =&\max_{\substack{t\in\mathbb{R}^{mn}\\ w\in\mathbb{R}^s}} \left(b^\top - \delta\mathbf{1}^\top\right) w \\ &\quad\text{s.t.} \quad \Psi^\top t + A^\top w = 0, \quad -1\le t\le 1. \end{aligned}$$

Implementation

We implemented the ℓ_1 inpainting algorithm with `linprog` function in MATLAB R2020a. The `interior-point` method was chosen for speed concerns.

```
...

% linprog
% min(c'x) s.t. Mx <= d
del     = delta*ones(s, 1);
I       = speye(m*n);
ze      = sparse(s, m*n);
c       = [zeros(m*n, 1); ones(m*n, 1)];
M       = [-Psi -I; Psi -I; -A ze; A ze];
d       = [zeros(2*m*n, 1); del-b; del+b];
options = optimoptions('linprog', 'Algorithm', 'interior-point', ...
                       'ConstraintTolerance', 1e-3, 'Display', 'iter');
x       = linprog(c, M, d, [], [], [], [], options);

...
```

(For complete source code, please refer to the attached `.m` files.)

Results

We tested the model on sample grey-scale images of 256×256 and 512×512 pixels. The results were obtained with a 4.0 GHz Quad-Core Intel Core i7-4790K processor and 32 GB 2133 MHz memory. The quality of the reconstructed images was assessed via the PSNR value:

$$\text{PSNR} := 10\cdot\log_{10}\frac{mn}{||x-u^*||^2},$$

where x is the reconstructed image and $u^* = \text{vec}(U^*)$ is the ground truth.

Overall performance

The ℓ_1 block model reconstructs 256×256 images contaminated by 50% random noise with $\text{PSNR}\approx 20$. The average runtime is around 40 sec with a maximum of 20 iterations. We notice that the algorithm takes significantly longer at each iteration, and more iterations to converge, if the block size bsz is set at 16 (the value recommended by the lecturer for low-res images). For instance, image $(\mathbf{a})$ takes nearly 8 min to reconstruct at $\text{bsz}=16$, along with a 2.3 increase in PSNR, and images $(\mathbf{b})$ and $(\mathbf{c})$ fail to converge within a 200-iteration limit. After examining matrix Ψ, we find that it is because the number of non-zero elements Ψ increase proportionally to bsz^2, resulting in significantly more computations.

	Ground Truth	Ground Truth + 50% Noise	Reconstructed Images	PSNR	Runtime (Iterations)
$(\mathbf{a})$	c	2 (2)	buildings_bsz_8_21.6	21.6	38.4 sec (17)
$(\text{bsz}=16)$			256_house+rand_50_del0.06_474s	23.9	7.9 min
$(\mathbf{b})$	B	2	256_hand+random_50_del0.06_48s	22.3	44.2 sec (20)
$(\mathbf{c})$	A	casino	256_casino+rand_50_del0.06_49s	20.4	44.5 sec (19)

Table 1. ℓ_1 inpainting results of 256×256 images contaminated by 50% random noise, $\delta = 6\times 10^{-2}$, $\text{bsz}=8$, constraint tolerance $=10^{-3}$, optimality tolerance $=10^{-6}$

bsz

Figure 1. Sparsity patterns of Ψ with $\text{bsz}=32$ (Left) and $\text{bsz}=64$ (Right), $m=n=256$

In the task of reconstructing 512×512 image polluted by 50% random noise, the algorithm achieves roughly the same PSNR compared to low-res images within similar number of iterations, while the runtime becomes roughly four times as long. We also observed that as the noise intensity increased, the image quality deteriorates drastically although the algorithm seems to converge faster.

Noise Percentage	Ground Truth	30%	50%	70%
Ground Truth + Noise	512_512_lena	lena_rand30	lena_rand50	lena_
Reconstructed Images	-	512_lena+rand_30_del0.06_265s_25.5	512_lena+rand_50_del0.06_186s_22.6	512_l 17
PSNR	-	25.5	22.6	17.0
Runtime (Iterations)	-	4.4 min (26)	3.1 min (21)	2.8 mi

Table 2. ℓ_1 inpainting results of 512×512 images contaminated by $30\%, 50\%$, and 70% random noises, $\delta = 6\times 10^{-2}$, $\text{bsz}=8$, constraint tolerance $=10^{-3}$, optimality tolerance $=10^{-6}$

In reconstructing images with non-random damages, the algorithm performs roughly the same as in the 50% random noise case with mesh, handwriting, and mild scratches. However, when inpainting the hard-scratched image, the algorithm performs poorly with $\text{PSNR}=13.9$, a score even lower than with the 70% random noise, which covers more area of the image than the scratches. From this, we conclude that the algorithm is sensitive to the distribution of the damage. In particular, the algorithm performs better if the damage distributes more evenly, i.e., there is no large "chunks" of damage area on the image (as in hard scratches).

Damage Types	Ground Truth	Mesh	Handwriting	Mild Scratches
Damaged Images	512_512_lena	mesh	handwritting	scratch
Reconstructed Images	-	512_lena+mesh_del0.06_253s_21.6	512_lena+writing_50_del0.06_253s_21.5	scratch FIX
PSNR	-	21.6	21.5	25.3
Runtime (Iterations)	-	4.2 min (26)	4.2 min (21)	4.4 min $(2$

Table 3. ℓ_1 inpainting results of 512×512 images with non-random damage, $\delta = 6\times 10^{-2}$, $\text{bsz}=8$, constraint tolerance $=10^{-3}$, optimality tolerance $=10^{-6}$

Effects of adjusting termination tolerances

We investigate the effects of adjusting optimality tolerance on the reconstruction quality by inpainting the 50% random noise contaminated image. We observe almost no changes either in image quality or runtime, when optimality tolerance ranges from 10^{-7} to 10^{-5}. Curiously, when optimality tolerance is larger than 10^{-4}, the algorithm gets significantly slower and fails to converge within 200 iterations.

Optimality Tolerance	Ground Truth	$10^{-3,-4}$	10^{-5}	10^{-6}	10^{-7}
Reconstructed Images	512_512_lena	-	optol_e-5	512_lena+rand_50_del0.06_186s_22.6	optol_e-7
PSNR	-	-	22.6	22.6	22.6
Runtime (Iterations)	-	(>200)	4.4 min (29)	3.1 min (21)	3.5 min (23)

Table 4. Effects of adjusted optimality tolerance on ℓ_1 inpainting results of 512×512 images contaminated by 50% random noise, $\delta = 6\times 10^{-2}$, $\text{bsz}=8$, constraint tolerance $=10^{-3}$

Our results also suggest the constraint tolerance has no significant impacts on reconstruction quality or speed. However when constraint tolerance is below 10^{-6}, the algorithm is significantly slower and struggles to converge within the 200-iteration limit.

Constraint Tolerance	Ground Truth	10^{-3}	10^{-4}	10^{-5}	$10^{-6,-7}$
Reconstructed Images	512_512_lena	512_lena+rand_50_del0.06_186s_22.6	contol_e-4	contol_e-5	-
PSNR	-	22.6	22.6	22.6	-
Runtime (Iterations)	-	3.1 min (21)	3.5 min (22)	3.4 min (22)	(>200)

Table 5. Effects of adjusted constraint tolerance on ℓ_1 inpainting results of 512×512 images contaminated by 50% random noise, $\delta = 6\times 10^{-2}$, $\text{bsz}=8$, optimality tolerance $=10^{-6}$

Effects of adjusting δ and comparison to the TV model

We investigate the effects of altering the error threshold δ on the reconstruction quality by inpainting the 50% random noise contaminated image. We observe that the image quality slightly decreases as the threshold grows from 0.006 to 0.06, but is still within the acceptable range. Nevertheless, when δ reaches 0.3, the reconstructed image becomes visibly darker and coarser, and at $\delta=0.6$, the image is almost unidentifiable.

Compared with the Total Variation model in part 1 of the project, the ℓ_1 block model is significantly inferior in both reconstruction quality and speed. The excessive runtime of the ℓ_1 model is partly due to a denser coefficient matrix compared with TV model. In this example, the density of the coefficient matrix in the ℓ_1 model is 8.33×10^{-5} while in the TV model, the number is 3.82×10^{-6}, almost two magnitudes lower.

δ	Ground Truth	TV (Interior Point)	0.006	0.06
Reconstructed Images	512_512_lena	lena-random50-interior-point-34.5214-19.0156	512_lena+rand_50_del0.006_276s_26.3	512_lena+rand_50_del0.06_18 2.6
PSNR	-	34.5	26.3	22.6
Runtime (Iterations)	-	11.0 sec (11)	4.6 min (30)	3.1 min (21)

Table 6. Effects of adjusted δ on ℓ_1 inpainting results of 512×512 images contaminated by 50% random noise, $\text{bsz}=8$, constraint tolerance $=10^{-3}$, optimality tolerance $=10^{-6}$, TV model result included for comparison

Denoising

We finally test the ℓ_1 block model under the denoising setting: $A=I_{mn}, b=u+\sigma\text{randn}(mn,1)$ where A,b,u are as in the TV minimization model, and randn adds scaled Gaussian white noise to the stacked image u. The following results are obtained with parameters set at $\sigma=0.1, \delta=0.9\sigma, \text{bsz}=8$.

Ground Truth	GT + Gaussian Noise	Denoised Image
512_512_lena	noised	denoised
PSNR	5.7	22.8

Table 7. ℓ_1 denoising result of 512×512 images contaminated by Gaussian white noise, $\sigma=0.1, \delta=0.9\sigma, \text{bsz}=8$.

Conclusion