MATH4230 Project Report

Pan Weiheng*

December 21, 2022

This is a project report for the course MATH4230 Optimization Theory on the paper *An Algorithm for Total Variation Minimization and Applications* by Antonin Chambolle [2].

1 Overview

Image denoising has been a popular topic in image processing for a very long time. Images captured by photodiode sensors are inevitably subject to noise caused by thermal noise in circuits and quantum noise of light. Denoising is hence widely adopted to improve image quality for industrial or aesthetic goals. In 2004, A. Chambolle [2] proposed a novel image denoising method by minimizing the total variation of an image, which is the subject of this report.

The rest of this report is structured as follows:

- Section 2 introduces some preliminary information about this problem. Section 3 contains the formulations of the total variation minimization problem. Section 4 describes the total variation-based image denoising problem. These three sections are better-explained versions of the corresponding sections of the original paper.
- Section 5 lists some experimental results on image denoising using a Python implementation of the algorithms and a comparison of their performances.
- Section 6 is the conclusion of this report.

*Student ID: 1155107657.

2 Notations and Preliminaries

This section aims to clarify the somehow loosely presented notations and preliminaries in the original paper by providing additional supportive details.

The images are $N \times N$ matrices. For simplicity, we define $X = \mathbb{R}^{N \times N}$ and $Y = X \times X$.

Since images are discrete, we need to define some discrete versions of operators.

Definition 1. *The discrete gradient operator $\nabla : X \to Y$ is given by*

$$(\nabla u)_{i,j} = \left((\nabla u)^1_{i,j}, (\nabla u)^2_{i,j}\right)$$

for any image $u \in X$, where

$$(\nabla u)^1_{i,j} = \begin{cases} u_{i+1,j} - u_{i,j} & \text{if } i < N \\ 0 & \text{if } i = N \end{cases}$$

$$(\nabla u)^2_{i,j} = \begin{cases} u_{i,j+1} - u_{i,j} & \text{if } j < N \\ 0 & \text{if } j = N \end{cases}$$

for all $i, j \in \{1, 2, \ldots, N\}$.

Note that this discrete gradient operator is linear and isotropic. The author also claimed that this choice of the discrete gradient operator offered a good compromise between isotropy and stability without offering any justification [2].

Then the total variation of the image $u \in X$ can be defined as the sum of the norms of the components of ∇u.

Definition 2. *The total variation of an image $u \in X$ is defined to be*

$$J(u) = \sum_{1 \le i,j \le N} \|(\nabla u)_{i,j}\|$$

where $\|\cdot\|$ is the standard Euclidean norm in $\mathbb{R}^2$. In this report, we assume that the norm is the standard Eucliean norm in the respective inner product space if not specified.

Chambolle [2] claimed that J is a discretization of the standard total variation for a function $u \in L^1(\Omega)$ defined on an open subset Ω of $\mathbb{R}^n$, which is given by

$$\sup \left\{ \int_\Omega u(x) \operatorname{div} \xi(x) \mathrm{d}x \mid \xi \in C^1_c(\Omega; \mathbb{R}^2),\ |\xi(x)| \le 1\ \forall x \in \Omega \right\}$$

where $C_c^1(\Omega;\mathbb{R}^2)$ denotes the space of all continuously differentiable functions with compact support defined on Ω. Hence the optimal solutions to this optimization problem approximates the optimal solutions to the continuous version of the problem as the pixels become finer and finer.

Next, we need to define a discrete divergence operator. In the continuous setting, the divergence operator is the negative of the "formal" adjoint of the gradient operator, i.e. $\operatorname{div} = -\nabla^*$. By the definition of the adjoint of an operator, we have $\langle -\operatorname{div} p, u\rangle = \langle p, \nabla u\rangle$ for any $p \in Y, u \in X$.

Definition 3. *The discrete divergence* $\operatorname{div} : Y \to X$ *is defined as*

$$(\operatorname{div} p)_{i,j} = \begin{cases} p^1_{i,j} - p^1_{i-1,j} & \text{if } 1 < i < N \\ p^1_{i,j} & \text{if } i = 1 \\ -p^1_{i-1,j} & \text{if } i = N \end{cases} + \begin{cases} p^2_{i,j} - p^2_{i,j-1} & \text{if } 1 < j < N \\ p^2_{i,j} & \text{if } j = 1 \\ -p^2_{i,j-1} & \text{if } j = N \end{cases}$$

for all $i, j \in \{1, 2, \ldots, N\}$ *and* $p = (p^1, p^2) \in Y$.

It can be verified [2] that the discrete divergence div defined above satisfies this relationship with respect to the discrete gradient operator.

Note that $J(u)$ may also be written as

$$J(u) = \sup_{p \in Q} \langle p, \nabla u\rangle$$

where

$$Q = \{p \in Y \mid \|p_{i,j}\| \leq 1 \ \forall i, j \in \{1, 2, \ldots, N\}\}$$

This is a consequence of the Cauchy-Schwarz inequality:

$$\begin{aligned} \langle p, \nabla u\rangle &= \sum_{1 \leq i,j \leq N} \langle p_{i,j}, (\nabla u)_{i,j}\rangle \\ &\leq \sum_{1 \leq i,j \leq N} |\langle p_{i,j}, (\nabla u)_{i,j}\rangle| \\ &\leq \sum_{1 \leq i,j \leq N} \|p_{i,j}\| \cdot \|(\nabla u)_{i,j}\| \\ &\leq \sum_{1 \leq i,j \leq N} \|(\nabla u)_{i,j}\| \\ &= J(u) \end{aligned}$$

where equality holds if and only if $p_{i,j}$ is a unit vector collinear with $(\nabla u)_{i,j}$ for all $i, j \in \{1, 2, \ldots, N\}$.

Let

$$K = \{\operatorname{div} p \mid p \in Q\} = \{\operatorname{div} p \mid p \in Y,\ \|p_{i,j}\| \leq 1\ \forall i, j \in \{1, 2, \ldots, N\}\}$$

Since $Q = -Q$, we have

$$\begin{aligned} J(u) &= \sup_{p \in Q} \langle p, \nabla u \rangle \\ &= \sup_{p \in Q} \langle -\operatorname{div} p, u \rangle \\ &= \sup_{p \in Q} \langle \operatorname{div} p, u \rangle \\ &= \sup_{v \in K} \langle v, u \rangle \\ &= \sup_{v \in K} \langle u, v \rangle \end{aligned}$$

Note that $J(u)$ is exactly the support function of K. By Example 7.1.2 of [1], the convex conjugate $J^*(v)$ of $J(u)$ is just the indicator function of K, i.e.,

$$J^*(v) = \sup_{u \in X} \left(\langle u, v \rangle - J(u)\right) = \chi_K(v) = \begin{cases} 0 & \text{if } v \in K \\ +\infty & \text{if } v \notin K \end{cases}$$

3 Total Variation Minimization

The primal problem is

Problem 1 (Total Variation Minimization)

$$\min_{u \in X} \quad J(u) + \frac{\|u - g\|^2}{2\lambda}$$

where $g \in X, \lambda > 0$ are given.

So here we are minimizing the total variation of the image plus an L^2 regularization term, which ensures the solution would not deviate from the original image too much.

Denote the primal optimal solution by u^*.

Differentiating the objective function with respect to u,

$$0 \in \frac{u - g}{\lambda} + \partial J(u) \iff \frac{g - u}{\lambda} \in \partial J(u) \iff u \in \partial J^* \left(\frac{g - u}{\lambda} \right)$$

Which is then equivalent to

$$0 \in \frac{g - u}{\lambda} - \frac{g}{\lambda} + \frac{1}{\lambda} \partial J^* \left(\frac{g - u}{\lambda} \right) \iff 0 \in w - \frac{g}{\lambda} + \frac{1}{\lambda} \partial J^*(w) \iff 0 \in \partial h(w)$$

where $w = \frac{g-u}{\lambda}$ and

$$h(w) = \frac{\left\| w - \frac{g}{\lambda} \right\|^2}{2} + \frac{1}{\lambda} J^*(w)$$

So w is the minimizer of $h(w)$.

By the definition of J^* and the projection theorem, we know the minimizer of $h(w)$ is just the point in K that is the "closest" to $\frac{g}{\lambda}$, i.e. the orthogonal projection of $\frac{g}{\lambda}$ to K. Hence

$$\frac{g - u}{\lambda} = w = \pi_K \left(\frac{g}{\lambda} \right)$$

Thus

$$u = g - \lambda w = g - \lambda \pi_K \left(\frac{g}{\lambda} \right) = g - \pi_{\lambda K}(g)$$

So finally we have

$$u^* = g - \pi_{\lambda K}(g) \tag{1}$$

Since g is known, all we need to do to solve for u^* is just to compute the nonlinear projection $\pi_{\lambda K}(g)$.

According to the definition of K, calculating $\pi_{\lambda K}(g)$ can be done by solving the following optimization problem:

Problem 2 (Nonlinear Projection)

$$\begin{aligned} \min_{p \in Y} \quad & \|\lambda \operatorname{div} p - g\|^2 \\ \text{subject to} \quad & \|p_{i,j}\|^2 - 1 \leq 0 \quad \forall i, j \in \{1, 2, \ldots, N\} \end{aligned}$$

Let p^* be the primal optimal value of this optimization problem. Then $\pi_{\lambda K}(g) = \lambda \operatorname{div} p^*$. So the solution to Problem 1 in terms of the solution to Problem 2 is

$$u^* = g - \lambda \operatorname{div} p^*$$

by equation 1.

Now we try to solve Problem 2. We have

$$\nabla \left(\|\lambda \operatorname{div} p - g\|^2 \right) = 2\nabla(\lambda \operatorname{div} p - g)$$

The Karush-Kuhn-Tucker (KKT) conditions[1] for this problem are:

$$\begin{aligned} \|p_{i,j}\|^2 - 1 \leq 0 \quad & \forall i, j \in \{1, 2, \ldots, N\} \\ \alpha_{i,j} \geq 0 \quad & \forall i, j \in \{1, 2, \ldots, N\} \\ \alpha_{i,j}(\|p_{i,j}\|^2 - 1) = 0 \quad & \forall i, j \in \{1, 2, \ldots, N\} \\ (\nabla(\lambda \operatorname{div} p - g))_{i,j} + \alpha_{i,j} p_{i,j} = 0 \quad & \forall i, j \in \{1, 2, \ldots, N\} \end{aligned}$$

where $\alpha \in \mathbb{R}^{N \times N}$ is the Lagrange multiplier.

Fix some $i, j \in \{1, 2, \ldots, N\}$.

Case 1. If $\alpha_{i,j} > 0$, then $\|p_{i,j}\| = 1$ by the third condition. Substituting into the fourth condition and taking norms on both sides gives $\alpha_{i,j} = \|(\nabla(\lambda \operatorname{div} p - g))_{i,j}\|$.

Case 2. If $\alpha_{i,j} = 0$, then $(\nabla(\lambda \operatorname{div} p - g))_{i,j} = 0$ by the fourth condition. Hence we also have $\alpha_{i,j} = \|(\nabla(\lambda \operatorname{div} p - g))_{i,j}\| = 0$.

In both cases, we have

[1] In the original paper, the first term in the fourth condition has a minus sign in front of it. This could be a mistake, since the objective function is being minimized instead of being maximized.

$$\alpha_{i,j} = \|(\nabla(\lambda \operatorname{div} p - g))_{i,j}\| = \lambda \left\| \left(\nabla \left(\operatorname{div} p - \frac{g}{\lambda} \right) \right)_{i,j} \right\|$$

This unique property of the Lagrange multiplier α has inspired Chambolle [2] to invent the following semi-implicit gradient descent algorithm:

Algorithm 1

Parameter: step size parameter $\tau > 0$.

Initialization: $p^0 = 0$.

Iteration:

$$p_{i,j}^{n+1} = p_{i,j}^n + \tau \left(\left(\nabla \left(\operatorname{div} p^n - \frac{g}{\lambda} \right) \right)_{i,j} - \left\| \left(\nabla \left(\operatorname{div} p^n - \frac{g}{\lambda} \right) \right)_{i,j} \right\| p_{i,j}^{n+1} \right)$$

Or explicitly,

$$p_{i,j}^{n+1} = \frac{p_{i,j}^n + \tau \left(\nabla \left(\operatorname{div} p^n - \frac{g}{\lambda} \right) \right)_{i,j}}{1 + \tau \left\| \left(\nabla \left(\operatorname{div} p^n - \frac{g}{\lambda} \right) \right)_{i,j} \right\|}$$

In Theorem 3.1 of the original paper [2], Chambolle proved the convergence of this algorithm for small τ. He showed that $\lambda \operatorname{div} p \to \pi_{\lambda K}(g)$ as $n \to \infty$ for $\tau \leq \frac{1}{8}$. The detailed proof is not included here for brevity.

However, in the remark of the theorem, he stated that the supremum of τ can be improved to $\frac{1}{4}$ in practice without any proofs. In a more recent paper by Zhu et al. [5], they used $\tau = 0.248$ for Chambolle's algorithm while comparing the performances of different total variation minimization algorithms. Unfortunately, they admitted that they did not have a proof either.

4 Image Denoising

In the paper, a noisy image g is modelled as an a priori piecewise smooth image u plus a random Gaussian noise with variance σ^2 [2]. To recover the uncorrupted image u, one can solve the optimization problem:

Problem 3 (Image Denoising)

$$\begin{aligned} \min_{u \in X} \quad & J(u) \\ \text{subject to} \quad & \|u - g\|^2 - N^2\sigma^2 = 0 \end{aligned}$$

where $\sigma > 0$ is given.

The intuition here is that the addition of noise will introduce a large amount of total variation to the original image due to the granular feature of the noise. In other words, the random fluctuations of the noise will contribute a lot towards the total variation of the image. Since we assume u to be piecewise smooth, it should own a very low total variation.

This idea is first formulated in the paper [4] by Rudin et al., in which they proposed a gradient-projection method and solved nonlinear partial differential equations to tackle this problem.

Note that Problem 3 has a similar form compared to Problem 1. The difference between Problem 1 and Problem 3 is that the energy (L_2 norm squared) constraint is represented by a penalty term in Problem 1, while it is presented directly by an equality constraint in Problem 3. In fact, in another paper [3] by Chambolle and Lions, it is proved that assuming $\|g - \overline{g}\| \geq N\sigma$, Problem 3 is equivalent to Problem 1 and has a unique solution given by the solution of Problem 1, where $\overline{g}$ is the average pixel value of g. So we may indeed use Algorithm 1 to solve it.

However, Problem 1 depends on λ greatly. The larger the λ, the stronger the denoising. Choosing a good λ value blindly is inevitably difficult. On the other hand, the standard deviation σ of the Gaussian noise can be estimated with other algorithms. This suggests we may use a given σ to solve the problem.

Recall that by equation 1 we have

$$\|\pi_{\lambda K}(g)\|^2 = \|g - u^*\|^2 = \|u^* - g\|^2 = N^2\sigma^2$$

So the problem is to find some $\lambda > 0$ such that $\|\pi_{\lambda K}(g)\| = N\sigma$.

Now fix $g \in X$. Let $f : (0, +\infty) \to [0, +\infty)$ be a function defined by $f(s) = \|\pi_{sK}(g)\|$ for any $s > 0$. Note that f can be computed by Algorithm 1.

Algorithm 2

Initialization: Choose some $\lambda_0 > 0$. Let $v_0 = \pi_{\lambda_0 K}(g)$. Let $f_0 = f(\lambda_0) = \|v_0\|$.

Iteration: $\lambda_{n+1} = \frac{N\sigma}{f_n}\lambda_n$, $v_{n+1} = \pi_{\lambda_{n+1}K}(g)$, $f_{n+1} = f(\lambda_{n+1}) = \|v_{n+1}\|$.

The projections $\pi_{\cdot K}(g)$ are calculated with Algorithm 1.

In Lemma 4.1 of [2], it is proved that f satisfies the following properties:

- $f([0, +\infty)) = [0, \|g - \overline{g}\|]$
- f is non-decreasing
- $s \mapsto \frac{f(s)}{s}$ is non-increasing
- $f \in W^{1,\infty}([0, +\infty])$
- $0 \leq f(s) \leq \frac{f(s)}{s} \leq 2\sqrt{2}N$ for $s \geq 0$ almost everywhere

where $W^{1,\infty}([0, +\infty])$ is the notation for a Sobolev space. So $f \in L^\infty([0, +\infty))$ with f and its weak derivative of order 1 have a finite L^∞ norm.

Using these properties, Theorem 4.2 of [2] assures that f_n will converge to $N\sigma$ and $g - v_n$ will converge to u^*.

Below Theorem 4.2, Chambolle also mentioned that one can replace λ with[2] $\frac{N\sigma}{\|\operatorname{div} p^*\|}$ after each iteration of Algorithm 1 to get a faster convergence rate, where p^* is the final value of a variable p in Algorithm 1. He did not provide a proof, though.

So we have the following accelerated version of Algorithm 2:

Algorithm 2 (Accelerated)

Initialization: Choose some $\lambda_0 > 0$. Let $v_0 = \pi_{\lambda_0 K}(g)$. Let $f_0 = f(\lambda_0) = \|v_0\|$.

Iteration: $\lambda_{n+1} = \frac{N\sigma}{f_n}\lambda_n$, $v_{n+1} = \pi_{\lambda_{n+1}K}(g)$, $\lambda_{n+1} = \frac{N\sigma}{\|\operatorname{div} p^*\|}$, $f_{n+1} = f(\lambda_{n+1}) = \|v_{n+1}\|$.

The projections $\pi_{\cdot K}(g)$ are calculated with Algorithm 1 and p^* is the final value of p in Algorithm 1.

In [2], Chambolle mentioned that he set the stopping criterion to be that the maximum variation between $p_{i,j}^n$ and $p_{i,j}^{n+1}$ is less than 0.01. However, this is not applicable to Algorithm 2. Instead, I decided to stop both algorithms when $\|u_{n+1} - u_n\| \leq \varepsilon N$, where u is the denoised image and $\varepsilon > 0$ is a threshold parameter.

In the next section, I will present some results of denoising images with the two algorithms as well as compare their performances.

---

[2] I did not use the original notation $\operatorname{div} p^n$ here because it is misleading. p^n here should be the final value of p, but n is also the iteration index in Algorithm 2.

5 Results

I wrote a direct implementation of both Algorithm 1 and the accelerated version of Algorithm 2 in Python, available here on GitHub. It also includes a piece of example code, so you may play with it easily.

Due to typesetting issues, the tables in this section are located at the end of this report. You may click the tables' numberings to quickly navigate to them.

5.1 Effect of Adjusting Lambda in Algorithm 1

First, we inspect the effect of using different λ values in Algorithm 1.

A 600×600 test image in grayscale is used. The threshold ε is 10^{-4}. The maximum number of iterations is set to 50. u_n is the noisy image generated by adding Gaussian noise of standard deviation $\sigma = 30$ to u. u_d is the denoised image. u is the original ground-truth image. Δu is the difference image, defined as $\Delta u = u_n - u_d$.

From Table 1 we can see that the strength of the denoising increases as λ increases. This is expected, since a large λ corresponds to a small L_2 norm regularization term in Problem 1, so there is less constraint to the minimization of the total variation of the image.

5.2 Effect of Adjusting Sigma in Algorithm 2

Next, we compare the denoising strength of using different σ value in Algorithm 2. In reality, the estimation of σ is error prone, so it is important to investigate this. The setting here is exactly the same as the last experiment in the previous subsection, so the noise has $\sigma = 30$.

As we can see from Table 2, the denoising is insufficient when $\sigma < 30$, adequate when $\sigma = 30$, and excessive when $\sigma > 30$. This is very evident and well anticipated. We conclude that it is important to estimate σ accurately for Algorithm 2 to function correctly.

5.3 Comparison

Finally, we compare the performances of the two algorithms in terms of the number of iterations needed for convergence.

Two test images are used. The λ parameter of Algorithm 1 is manually adjusted to a suitable value by trial-and-error. I denotes the number of iterations needed for convergence, which is a single integer if the image is grayscale, or a 3-tuple for the three channels if the image is a RGB image. All parameters are set to be the same for the two algorithms.

[3]The σ here is both the standard deviation of the noise and the input parameter to Algorithm 2.

As Table 3 and Table 4 showed, it is evident that Algorithm 2 takes less iterations to converge while achieving almost the same denoising performance as Algorithm 1. Hence Algorithm 2 is indeed more performant than Algorithm 1.

6 Conclusion

To conclude, this report has showed the basic derivations of the problems in the original paper, introduced the algorithms, and showed some extra results using the algorithms. Due to length constraint, I could not include the details on the proofs of convergence of the algorithms.

I personally find the paper quite difficult to read. Many important details are skipped and the logic flow is a bit unclear, especially at the connections of different parts. As a consequence, I have spent a large amount of time trying to understand it, although the problems and algorithms themselves are in fact not very difficult.

Due to time limitations, I have only focused on the first four sections of the paper. The omitted Section 5 was about using total variation minimization to solve another image processing problem, image zooming. However, I seriously doubt the applicability of it. In the problem formulation, a full-resolution image is reduced to 1/4 of its original size by first dividing it into 2×2 pixel grids, and then merge the upper left pixels of the grids. Optically, each 2×2 grid is reduced to a pixel whose value is the average of the four pixels instead. Consequently, the algorithm in Section 5 is not applicable to this new model, since this is a deconvolution problem and the resulting image is no longer a subset of the original image now. As for Section 6, I can roughly understand the idea of mean curvature motion of a curve, but I do not have enough background knowledge in advanced real analysis and differential geometry to fully understand it.

To sum up, I have learned a new class of optimization problem, a novel denoising method, how to use Python to write algorithms and process images, and how to use LaTeXto write a long, structured report with images and tables.

References

[1] Dimitri P. Bertsekas, Angelia Nedić, and Asuman E. Ozdaglar, *Convex analysis and optimization*, Athena Scientific Optimization and Computation Series, Athena Scientific. OCLC: 248882658.

[2] Antonin Chambolle, *An Algorithm for Total Variation Minimization and Applications* **20** (January 2004), no. 1/2, 89–97.

[3] Antonin Chambolle and Pierre-Louis Lions, *Image recovery via total variation minimization and related problems* **76** (April 1, 1997), no. 2, 167–188.

[4] Leonid I. Rudin, Stanley Osher, and Emad Fatemi, *Nonlinear total variation based noise removal algorithms* **60** (November 1992), no. 1-4, 259–268.

[5] Mingqiang Zhu, Stephen J. Wright, and Tony F. Chan, *Duality-based algorithms for total-variation-regularized image restoration* **47** (November 2010), no. 3, 377–400.

Table 1: Effect of Adjusting Lambda in Algorithm 1

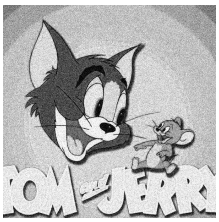
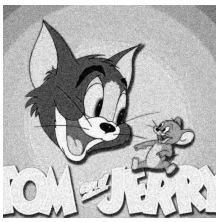
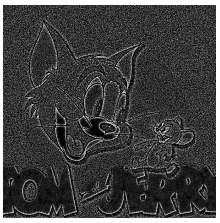
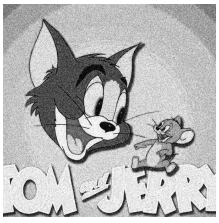
λ	u_n	u_d	u	Δu
5 000				
10 000				
20 000				
40 000				

Table 2: Effect of Adjusting Sigma in Algorithm 2

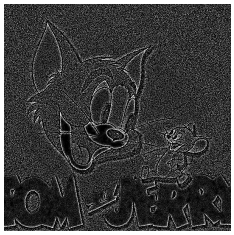
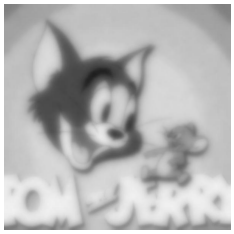
σ	u_n	u_d	u	Δu
15				
30				
45				
60				

Table 3: Image Denoising Performance of Algorithm 1

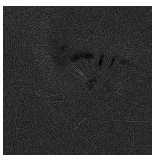
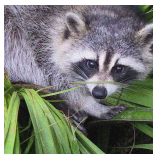
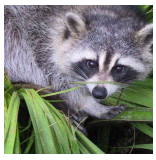
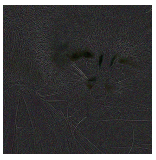
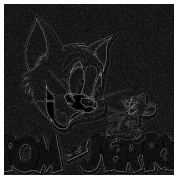
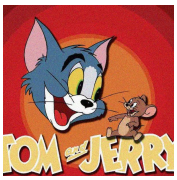
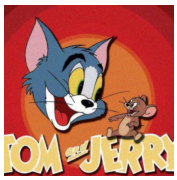
N	σ	λ	ε	I	u_n	u_d	u	Δu
600	15	10 000	10^{-4}	18				
600	15	10 000	10^{-4}	(18,18,18)				
768	30	10 000	10^{-4}	11				
768	30	10 000	10^{-4}	(11,11,11)				

Table 4: Image Denoising Performance of Algorithm 2

N	σ^3	ε	I	u_n	u_d	u	Δu
600	15	10^{-4}	4				
600	15	10^{-4}	(4,4,4)				
768	30	10^{-4}	9				
768	30	10^{-4}	(9,9,9)