EXERCISE 5 - RIGID TRANSFORM BLENDING AND VARIATIONAL METHODS

Akiyo Worou

May 4, 2021

1 PART 1: Understanding and Utilizing Dual Quaternions

1.1 Task 1: Thinking about fundamental properties

Question 1:
Dual quaternions have many useful properties to make fast and easy computation such as constant speed, coordinate invariance, shortest path, multiplication (which can represent 3D translation), easy reversibility. Those properties are useful for blending.

Question 2:
The one fundamental disadvantage of using quaternion is the appearance of artifacts. Indeed, with dual quaternions there is no need to cluster vertices so the artifacts are avoided.

1.2 Task 2: Derivations and deeper understanding

Question 3:
Let's assume that $t > 0$. We know that, $\hat{q}^t = e^{t log(\hat{q})}$ and $log(\hat{q}) = \frac{\hat{s}\hat{\theta}}{2}$.
Hence, $\hat{q}^t = e^{t\frac{\hat{s}\hat{\theta}}{2}}$.

$$
\begin{aligned}
t\frac{\hat{s}\hat{\theta}}{2} &= \frac{t}{2}(s_0 + \varepsilon s_\varepsilon)(\theta_0 + \varepsilon\theta_\varepsilon) \\
&= \frac{t}{2}(s_0\theta_0 + \varepsilon(s_0\theta_\varepsilon + \theta_0 s_\varepsilon) + \varepsilon^2 s_\varepsilon\theta_\varepsilon) \\
&= \frac{t}{2}(s_0\theta_0 + \varepsilon(s_0\theta_\varepsilon + \theta_0 s_\varepsilon)) && (\varepsilon^2 = 0)
\end{aligned}
$$

We can then compute the norm of $t\frac{\hat{s}\hat{\theta}}{2}$:

$$\begin{aligned}
\left\| t\frac{\hat{s}\hat{\theta}}{2} \right\| &= \frac{t}{2}\| s_0\theta_0 + \varepsilon(s_0\theta_\varepsilon + \theta_0 s_\varepsilon)\| \\
&= \frac{t}{2}\left(\|s_0\theta_0\| + \varepsilon\frac{< s_0\theta_0, s_0\theta_\varepsilon + \theta_0 s_\varepsilon >}{\|s_0\theta_0\|} \right) && (\|\hat{q}\| = \|q_0\| + \varepsilon\frac{< q_0, q_\varepsilon >}{\|q_0\|}) \\
&= \frac{t}{2}\left(\|s_0\theta_0\| + \varepsilon\frac{\theta_0\theta_\varepsilon < s_0, s_0 > + \theta_0^2 < s_0, s_\varepsilon >}{\|s_0\theta_0\|} \right) \\
&= \frac{t}{2}\left(\theta_0\|s_0\| + \varepsilon\frac{\theta_\varepsilon}{\|s_0\|} \right) && (< s_0, s_0 >= 1, < s_0, s_\varepsilon = 0, \theta \geq 0) \\
&= \frac{t\hat{\theta}}{2} && (\|s_0\| = \sqrt{< s_0, s_0 >} = 1, \hat{\theta} = \theta_0 + \varepsilon\theta_\varepsilon)
\end{aligned}$$

Finally,

$$e^{t\frac{\hat{s}\hat{\theta}}{2}} = cos\left(\frac{t\hat{\theta}}{2}\right) + \frac{\frac{t\hat{s}\hat{\theta}}{2}}{\frac{t\hat{\theta}}{2}} sin\left(\frac{t\hat{\theta}}{2}\right)$$

The final result is then:

$$\hat{q}^t = cos\left(\frac{t\hat{\theta}}{2}\right) + \hat{s}sin\left(\frac{t\hat{\theta}}{2}\right)$$

Question 4:

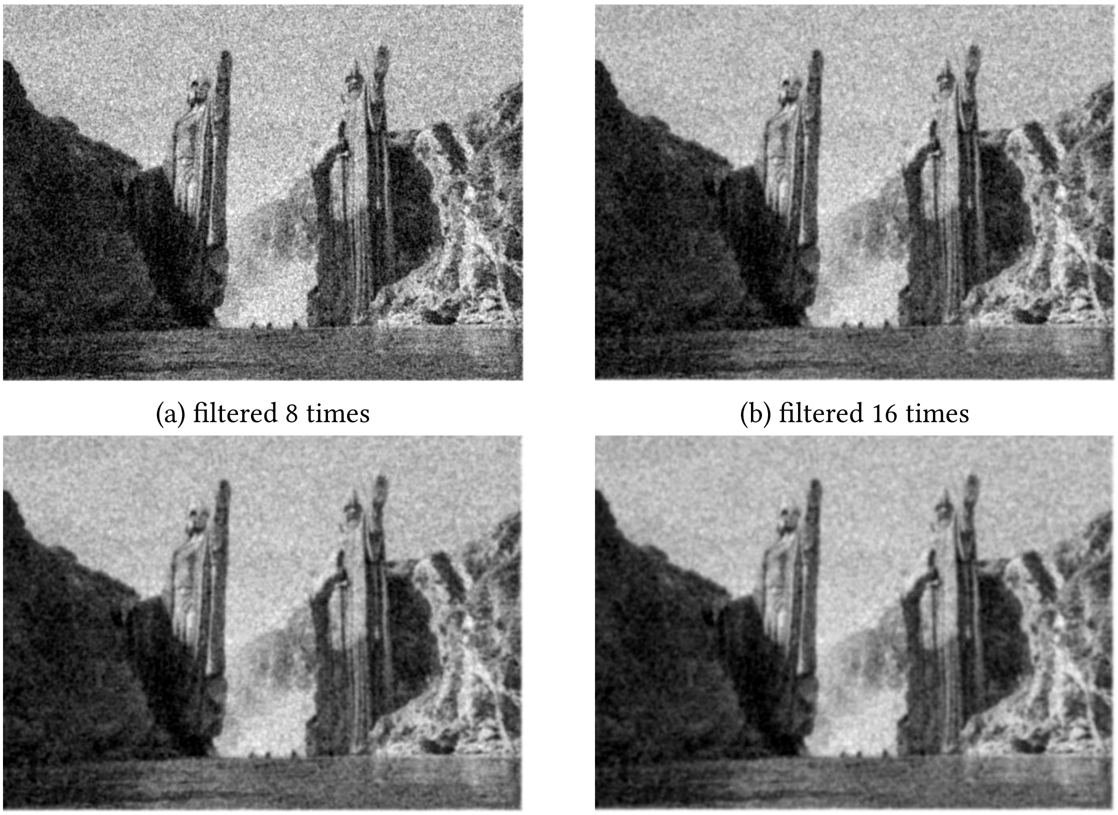
We already know that we can translate in the plan xy. By making the right choice for t we can use $\hat{q}^t$ to make a rotation around z.

2 PART 2: Variational Methods - Denoising problems

There is a small error in the given code for the value of the variance in the noise. The value 0.05 in the code is supposed to be $\sqrt{0.05}$ to get the same image as in the wording of the assignment. The formula used for the error is $error = Normalized(\sum |I_{denoised} - I_{original}|)$

2.1 Task 1: Filtering

The images obtained after the Gaussian filtering are represented below:



(a) filtered 8 times

(b) filtered 16 times

(c) filtered 24 times

(d) filtered 32 times

Figure 1: Denoised image after filtering multiple time

The error curve is represented in the figure below :

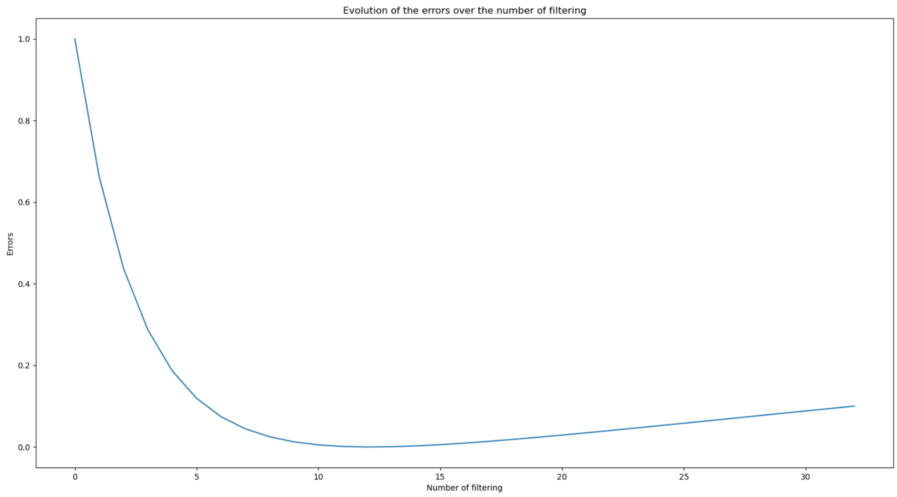
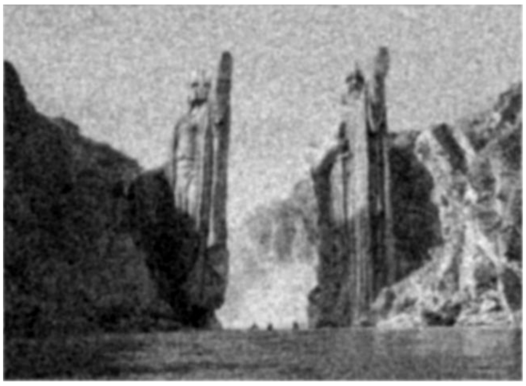


Figure 2: Denoised image after filtering multiple time

2.2 Task 2: Heat diffusion

Similarly as done before the images results are shown in the figure below as well as the error.



(a) Diffusion at time 25

(b) Diffusion at time 50

(c) Diffusion at time 75

(d) Diffusion at time 100

Figure 3: Denoised image after diffusion with $\tau = 0$

The error curve is represented in the figure below :

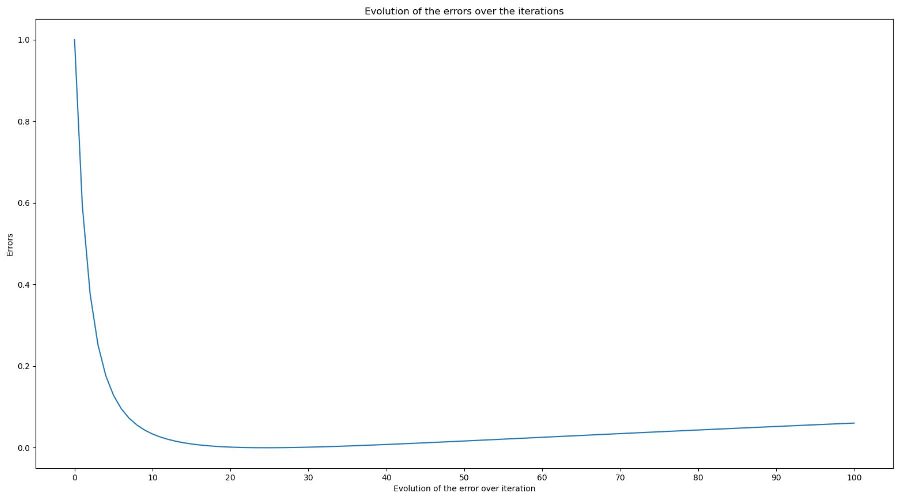


Figure 4: Denoised image after filtering multiple time

2.3 Task 3: Variational approach

Energy derivation

Let h an image such as $h \longrightarrow 0$

$$\begin{aligned} E(I+h) &= \int_\Omega \left[(I+h-I_0)^2 + \lambda\|\nabla(I+h)\|^2\right] dx \\ &= E(I) + \int_\Omega \left[(I-I_0)h + \lambda(\|\nabla I\|^2 + \|\nabla h\|^2 + 2\lambda\nabla h \cdot \nabla I)\right] dx \\ &= E(I) + 2\int_\Omega [(I-I_0-\lambda div(\nabla I))h]dx + 2\int_\Omega \lambda\nabla(h\nabla I)dx \\ because \quad & \nabla h \cdot \nabla I = \nabla \cdot (h\nabla I) - div(\nabla I)h \end{aligned}$$

$$E(I+h) - E(I) = 2\int_\Omega [(I-I_0-\lambda div(\nabla I))h]dx \qquad \text{h} \approx 0$$

$$E(I+h) - E(I) = \int_\Omega \frac{\delta E}{\delta I} h dx$$

Finally,

$$\frac{dE}{dI} = \int_\Omega (I - I_0 - \lambda div(\nabla I))dx$$

We then easily get, the final equation by writing $\frac{dE}{dI} = 0$. The equation is :

$$I_0 = I_u - \lambda \cdot div(\nabla I_u)$$

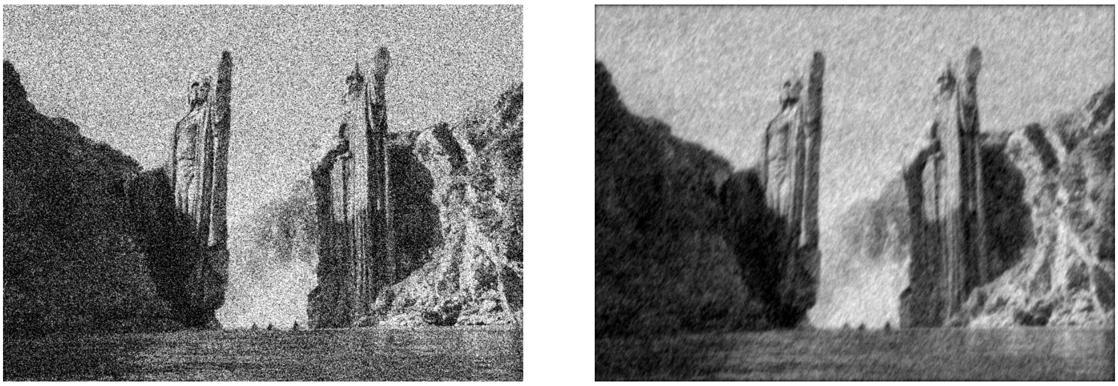
Solving the variational problem

The matrix A is of size $(H \times W) \times (H \times W)$ and can be rewritten by using other matrix A_λ of size $H \times H$, B_λ of size $H \times H$ and C_λ of size $H \times H$. Here are the expressions of those matrix:

$$A_\lambda = \begin{bmatrix} a & b & 0 & \dots & 0 \\ b & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ & \ddots & \ddots & \ddots & b \\ 0 & \dots & 0 & b & a \end{bmatrix} B_\lambda = \begin{bmatrix} 0 & b & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ & & \ddots & \ddots & 0 \\ & & & \ddots & b \\ 0 & \dots & & \dots & 0 \end{bmatrix} C_\lambda = \begin{bmatrix} 0 & \dots & & \dots & 0 \\ b & \ddots & & & \vdots \\ 0 & \ddots & \ddots & & \\ & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & b & 0 \end{bmatrix} A = \begin{bmatrix} A_\lambda & B_\lambda & 0 & \dots & 0 \\ C_\lambda & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ & \ddots & \ddots & \ddots & B_\lambda \\ 0 & \dots & 0 & C_\lambda & A_\lambda \end{bmatrix}$$

We just need the pseudo inverse of A (A is singular) and use the vectorized expression of I and I_u. In python we can directly solve the equation by using spsolve. For $\lambda = 2$ we have $a = 9$ and $b = -2$.

The results are in the figure below:



(a) Noisy image (b) Variational denoising

Figure 5: Denoised image with variational method

2.4 Task 4: Comparison / Questions

- The Gaussian convolution error is null after filtering 12 times. Going further will make it converges to a steady state. The state clearly depends on the boundary conditions. For Neumann the state will be an image with only white pixels and for Dirichlet it will be an image with only black pixels. It is the optimization of a convex function,the algorithm have to stop after reaching the minimum.
- Applying infinitely heat diffusion implies the convergence of the error to a steady state. The state clearly depends on the boundary conditions. For Neumann the state will be an image with only white pixels and for Dirichlet it will be an image with only black pixels.
- Let consider a functional J defined by $J(x) = \int_{t_0}^{t_1} L(t, x(t), \dot{x}(t))\,dt$ on G and for every function x ∈ G, stationary we have:

$$\frac{L}{x} - \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{x}}\right) = 0$$

. The equation is used to find a local minimum of an Energy.

- The solution is not globally optimal and to change the smoothing we can change the value of the parameter λ and also I_0 which represents the starting point. An idea could be to choose λ not too large or too small(for speed), to reach a minimum. But this minimum is not necessarily global. We need to choose different I_0 (starting point) to ensure that we reach the global minimum.

- The results show that we can get a less noisy image but the parameters play an important role in the effectiveness of each method. They all seem quite effective even if the variational method does not perform as well as the first two. Even the first two methods are similar the Gaussian filter reach the optimal solution faster than the heat diffusion.

  The first two methods reach the global minimum (We can decide to stop when it is reached) but the last one does not reach the global minimum. The variational method seems to be faster than the two first methods, and the heat diffusion is faster than the Gaussian filtering. According to efficiency I think the order is reversed. For all the methods we need to set the right parameters to get a good result.

- The idea behind the first two methods are similar: a certain combination of the noises in a region is smaller than the initial noise at the pixel position considered. For instance for the heat diffusion the idea is to get the new noise smaller $noise_{new} = (n_1+n_2+n_3+n_4)-4\cdot n_0 < n_0$ with n_1, n_2, n_3, n_4 the noises of the neighbors and n_0 the noise at the current pixel. We can easily understand that the new noise is smaller than it's previous value if the noises have the same nature or distribution. The idea is the same for Gaussian filtering but just with more neighbors taken into consideration.

  The idea behind the variational method is to minimize the energy, but more precisely minimize the entropy (reach zero). The noises represent a disorder in the image and the variational method is used to make things back in the order. That's how I understand the variational method.