Tempering with the Ambrosio Tortelli Approximation of Mumford-Shah Regularizer

CENG 566 Final Project

Aydın Göze Polat

1631092

1. Image Regularization

A common inverse problem of image processing is regularization of images. As any regularization problem, the solution requires one to introduce information that is not necessarily true. For example by assuming that high variation in an image is not natural, denoising and texture removal can be achieved. To do that, one needs to use a formal representation that can incorporate such assumptions. For this purpose, energy functionals are used. High variation and other undesired properties increase the energy functional. Thus, the whole regularization task can be considered as a minimization problem.

Although images can have high variation, this is often undesired since the computational resources to process and classify an image and the amount of information to represent it increase. Therefore, reducing the amount of variation within an image and reducing the edge count in a meaningful way (i.e. acquiring a cartoon image = a simpler and more compact representation of the image) is a prominent interest area for image processing[1].

1.a. Perona Malik Model

The model proposed by Perona Malik incorporates the idea that allowing the image to have variation at critical regions (such as prominent edges) is good and there is no need for penalty. To formalize this idea, they use a diffusion function g, that determines the diffusion speed according to some criteria such as image gradient. This, in effect reduces the penalty of having variations in the regions where graident is high. The energy functional is given below:

$$E_{PM}(u)=\frac{1}{2}\int_{\Omega} g(|\nabla u(x)|^2)\,dx$$

Basic PM equation is then given as:

$$u_t=\operatorname{div}(g(|\nabla u|))\nabla u \qquad \text{in} \qquad \Omega\, x(0,+\infty)$$

$$\frac{\partial u}{\partial n}=0 \qquad \text{in} \qquad \partial\Omega\, x(0,+\infty)$$

$$u(x,0)=u_0(x) \qquad \text{in} \qquad \Omega$$

where denotes picture domain. Equivalently it can also be written in terms of diffusivity function and the derivative of flux, as below:

$$u_t=\Phi'(|\nabla u|)u_{nn}+g(|\nabla u|)u_{ee}$$

where the coordinate n is parallel to the gradient of u and e is perpendicular. Thus, in the coordinate n (gradient direction), there is *forward-backward type diffusion* as illustrated in Figure 1: Right.

1 I selected this topic for two reasons: First, it allowed me to get a nice overview of what we did in the class. Second, I had an idea to improve AT, which I saw as an opportunity of applying what we learned in the class to achieve something new. (Or more correctly, I “hope” it is new, since I did not have enough time to make a good literature survey.)

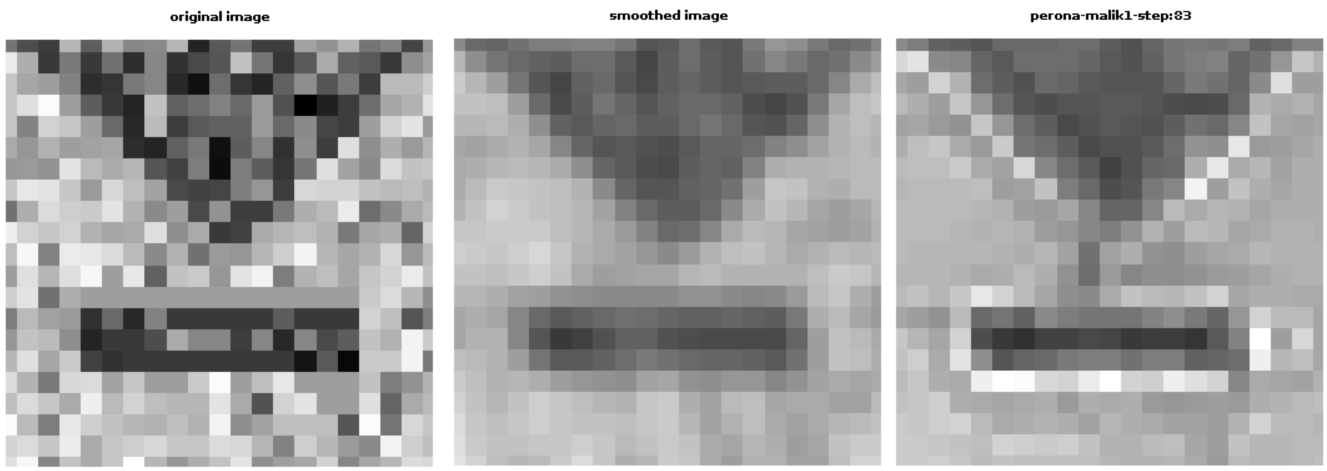

*Figure 1. Left: Original image, Middle: Smoothed image, Right: PM regularized image (*from **upgraded hw3**, parameters used: lambda=0.05, delta t=0.02)*[2]

1.b. Mumford-Shah Model

By minimizing the texture like details in the image, a cartoon version of an image can be acquired (i.e. image regularization). Therefore the cardinality of the edge set in the image contributes to the energy functional. To better punish the "unnecessary" details in an image, Mumford-Shah proposed below formula:

$$E_{MS}(u,\S)=\beta\int_{\Omega}(u-g)^2\,dx+\alpha\int_{\Omega-\S}|\nabla u|^2\,dx+length(\S)$$

The first part of the functional (fidelity component) increases energy if the original image g differs too much from the regularized version of g, namely u. The second part increases as the variation within the image u increases. The third part gives penalty to edges. However it is hard to calculate third part. Therefore, in general an approximation proposed by Ambrosio Tortelli is used for this model.

1.b.1. Ambrosio Tortelli Approximation

Instead of trying to directly incorporate the cardinality of edge set § to the energy formula, Ambrosio and Tortorelli proposed below approximation to the E_{MS}:

$$E_{AT}(u,v)=\int_{\Omega}[\beta(u-g)^2+\alpha(v^2|\nabla u|^2)+\frac{1}{2}(\rho|\nabla v|^2+\frac{(1-v)^2}{\rho})]\,dx \tag{1}$$

The minimization problem can be solved using below coupled PDEs:

$$\frac{\partial u}{\partial t}=\nabla.(v^2\nabla u)-\frac{\beta}{\alpha}(u-g) \tag{2}$$

$$\frac{\partial v}{\partial t}=\nabla^2 v-\frac{v-1}{\rho^2}-2\frac{\alpha}{\rho}|\nabla u|^2 v \tag{3}$$

---

2 *I later updated **myCLMC.m** in **hw3**. After I found a bug and fixed my central difference function in the implementation (namely **myCentralDiff.m** which I used in both PM and AT homeworks), everything worked fine without any other problem. The codes are available in the project folder.*

In Figure 2, the original image and its regularization using AT approximation is illustrated[3].

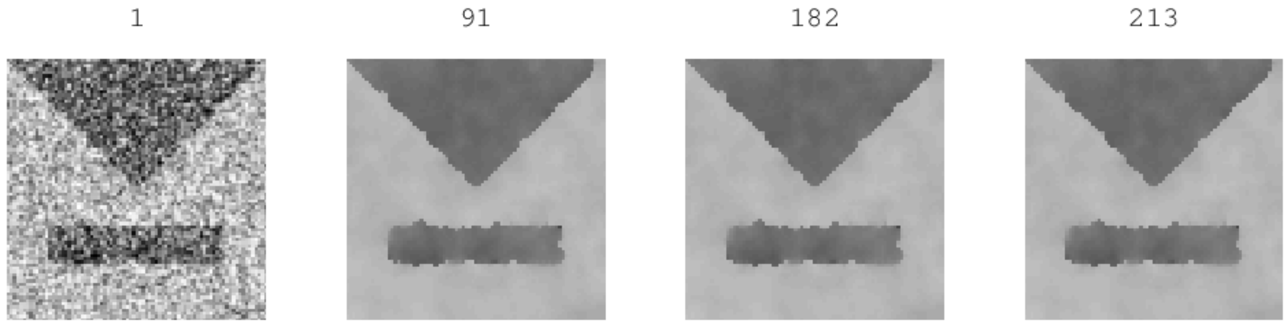

Figure 2. AT for α = 5000, β=200, delta=0.24, ρ=0.05, (regularized image u, converged at step 213, epsilon=0.0005)

Notice that the prominent edges starts to slightly diffuse away, even though jitter like artifacts still exist.

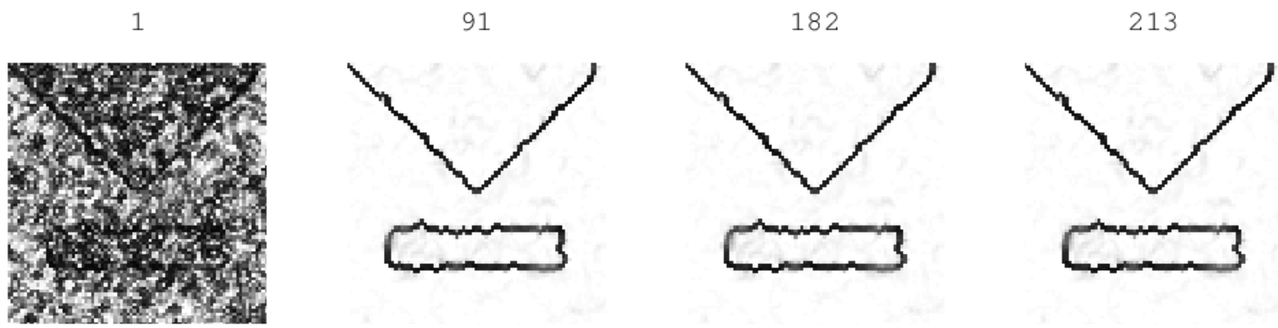

Figure 3. For α = 30000, β=2000, delta=0.24, ρ=0.005, phi=β/α=0.066, $\alpha/\rho=6*10^6$, (edge set)

While AT is generally efficient since it converges in a small number of steps, the regularized image converges to a final state where some unwanted artifacts from noise remain; since the fidelity becomes important, the shapes of edges freezes in their constant form (jitter in the edges due to noise). Another problem is that it can be hard to find the best parameters for various images with varying detail levels. Finding the right scale so that the prominent components of the image does not diffuse away can be time consuming. Moreover, even finding the right scale may not help removing some undesired artifacts. There are several ways to tackle these problems. In general the improvements are made by changing/replacing a term in the AT model. For example Esedoglu and Shen (as cited in Erdem and Tari's work) uses higher order geometric terms for a better approximation of the cardinality of edge set. Erdem and Tari replaces the term for diffusion speed v^2 in Formula 2 with an enhanced measure which, aside from v, also incorporates local information gathered from a larger collection of local neighborhoods. My work focuses on local information that is gathered from one more source that

3 *From upgraded hw5:* ***myAT.m****, there is also the slight modification we did in the class* ***myModifiedAT.m****. I worked on them because this project uses myAT.m (where I personally implemented the original AT approximation) as its starting point and makes improvements on it.*

corresponds to edge diffusivity speed. I will call it as temperature field.

2. Introducing Another Diffusivity to the AT Approximation

I propose to add diffusion speed to v. The diffusion speed can be considered as a temperature term. My assumption is that if an edge is too isolated, than it probably is not very important. Thus it should diffuse away faster. It can be imagined as a water droplet that will diffuse much faster than the main body if the temperature is high. In a way, heat dependent diffusion is expected to reduce the "surface" area of the edges. Therefore, if the edges have initially low temperature, and the low intensity parts of the image have low temperature, the edges that are more isolated than others would diffuse more quickly, since they would have a lower "volume/surface" ratio. Since in general, I initialized the temperature field from the original image, the temperature field (or diffusivity metric for v) h can also be considered as some transformation of the image.

2.1. Modification 1: Temperature as the *Diffusivity of v*

In this section, I change the Formula 1 as below:

$$E_{AT}(u,v)=\int_{\Omega}[\beta|\nabla h|^2(u-g)^2+\alpha(v^2|\nabla u|^2)+\frac{1}{2}(\rho(h^2|\nabla v|^2)+\frac{(1-v)^2}{\rho})]dx \tag{4}$$

where fidelity depends on temperature (h) gradient. If the heat flow is high, fidelity is important, else the the equation will allow a more cartoon like u. Another way to look at this is the speed of heat diffusion depends on the error between u and g. If they are similar, then temperature gradient can become large, else temperature gradient should become smaller to minimize energy. The whole system with the addition of a third PDE:

$$\frac{\partial u}{\partial t}=\nabla.(v^2\nabla u)-\frac{\beta}{\alpha}|\nabla h|^2(u-g) \tag{5}$$

$$\frac{\partial v}{\partial t}=\nabla.(h^2\nabla v)-\frac{v-1}{\rho^2}-\frac{2\alpha v|\nabla u|^2}{\rho} \tag{6}$$

$$\frac{\partial h}{\partial t}=\nabla.(\beta(u-g)^2\nabla h)-\frac{h\rho|\nabla v|^2}{2} \tag{7}$$

2.1.1 Discretization of Modification 1

Using semi implicit discretization:

Discretization for Formula 5:

$$u^{k+1}-u^k=\frac{\Delta t}{2}[v^{2.}(Lu^k)-u^k.(Lv^2)+L(v^2.u^k)]-\Delta t\frac{\beta}{\alpha}|\nabla h|^2(u^{k+1}-g) \tag{8}$$

$$u^{k+1}(1+\Delta t\frac{\beta}{\alpha}|\nabla h|^2)=u^k+\Delta t\frac{\beta}{\alpha}|\nabla h|^2 g+\frac{\Delta t}{2}[v^{2.}(Lu^k)-u^k.(Lv^2)+L(v^2.u^k)] \tag{9}$$

$$u^{k+1}=(1+\Delta t\frac{\beta}{\alpha}|\nabla h|^2)^{-1}[u^k+\Delta t\frac{\beta}{\alpha}|\nabla h|^2 g+\frac{\Delta t}{2}(v^{2.}(Lu^k)-u^k.(Lv^2)+L(v^2.u^k))] \tag{10}$$

Discretization for Formula 6:

$$v^{k+1}-v^k=\frac{\Delta t}{2}[h^{2.}(Lv^k)-v^k.(Lh^2)+L(h^2.v^k)]-\frac{\Delta t(v^{k+1}-1)}{\rho^2}-\frac{2\alpha\Delta t U_s v^{k+1}}{\rho} \tag{11}$$

$$v^{k+1}+\frac{\Delta t\, v^{k+1}}{\rho^2}+\frac{2\,\alpha\,\Delta t\, U_s\, v^{k+1}}{\rho}=v^k+\frac{\Delta t}{2}[h^{2.}(Lv^k)-v^k.(Lh^2)+L(h^2.v^k)]+\frac{\Delta t}{\rho^2} \tag{12}$$

$$v^{k+1}=\left(1+\frac{\Delta t}{\rho^2}+\frac{2\,\alpha\,\Delta t\, U_s}{\rho}\right)^{-1}\left[v^k+\frac{\Delta t}{2}[h^{2.}(Lv^k)-v^k.(Lh^2)+L(h^2.v^k)]+\frac{\Delta t}{\rho^2}\right] \tag{13}$$

where $U_s=|\nabla u|^2$ and L is the square matrix (mnxmn) form of the Laplace operator (*step size is one*).

Discretization for Formula 7:

$$h^{k+1}-h^k=\Delta t\,\nabla.(\beta(u-g)^2\nabla h^k)-\frac{h^{k+1}\rho\,\Delta t|\nabla v|^2}{2} \tag{14}$$

$$h^{k+1}=\left(1+\frac{\rho\Delta t|\nabla v|^2}{2}\right)^{-1}[h^k+\Delta t\,\nabla.(\beta(u-g)^2\nabla h^k)] \tag{15}$$

2.1.2. Experiments for Modification 1

2.1.2.1. *Edge strengthening*

Increasing α dramatically, makes the edges in h become increasingly darker and thicker, giving an increasingly powerful edge strengthening effect, as long as the edge exists in the edge set, if the edge disappears, the edge also starts to diffuse as illustrated in Figures 4 and 5.

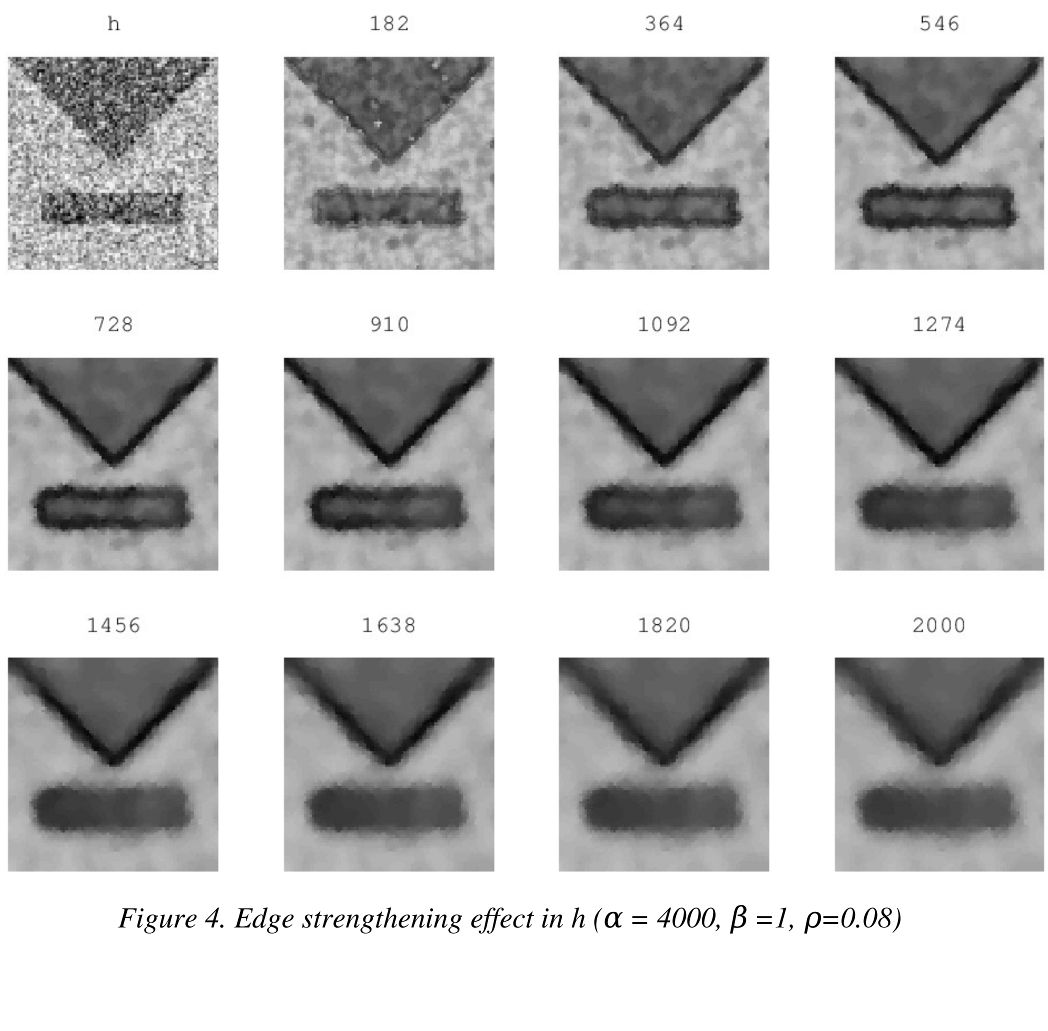

Figure 4. Edge strengthening effect in h (α = 4000, β =1, ρ=0.08)

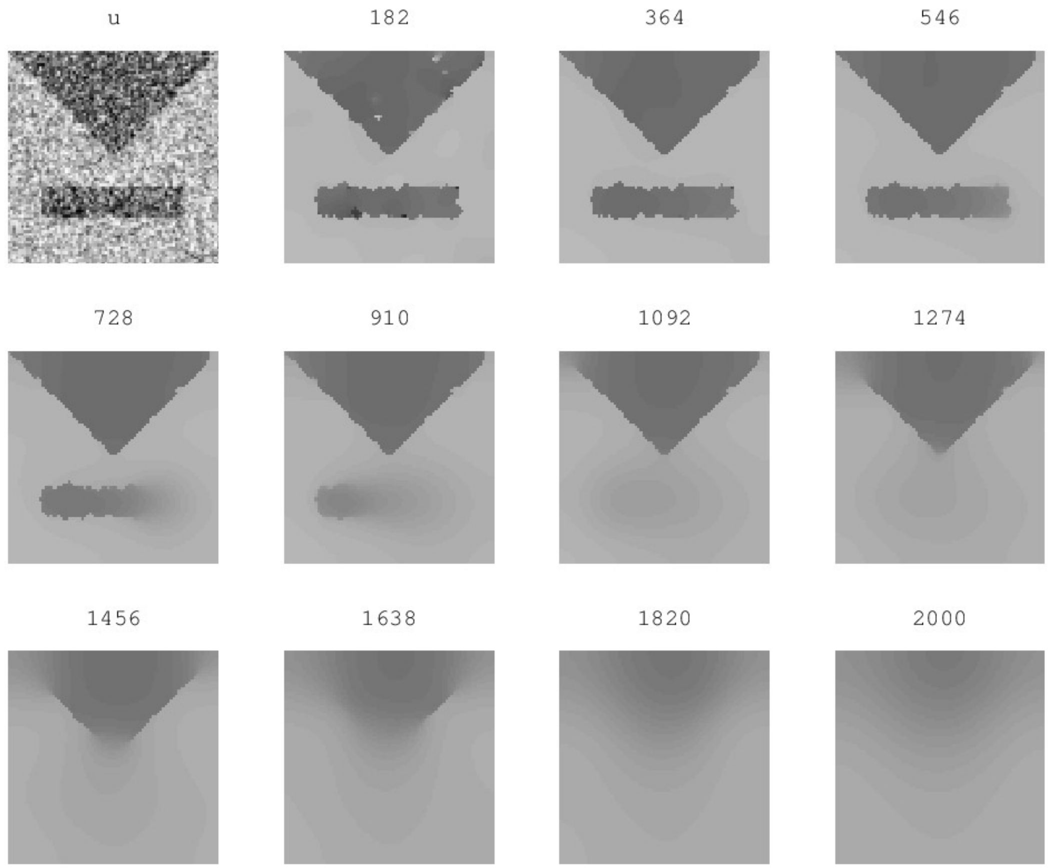

Figure 5. Corresponding u image (α = 4000, β =1, ρ=0.08)

Notice that after an edge disappears, the strengthened edges only gradually disappears.

Why the edges of the temperature field get stronger when α is increased?

This is due to the diffusion or variation component (or edge-gradient consistency penalty). Since the main problem becomes decreasing any kind of variation in the system (unless there is an edge), the solution can be either allowing the system to quickly dissolve into uniform state (by removing all edges) or settling into a local minimum where the effects of interactions are minimized along the edges (to prevent inconsistencies between gradient and edges). Since the punishment of increasing variation is quite high and black regions minimize the diffusion rate, edges of h become darker and darker (or “colder”). If the image has too many edges, the temperature field would become the darkest. Also observe that even from beyond a protective layer of “cold” regions, the system eventually gets out from the local minimum to the global one (uniform state), since the edges continue to diffuse.

2.1.2.2. *Hierarchical Texture Removal*

The modification 1 was especially made for the purpose of selectively removing “isolated” parts from the image. When α is extremely small (*contrast of v is reduced and the intensity becomes close to one everywhere*), the temperature field h can be effectively used as a regularizer for seemingly scale independent texture removal, as illustrated in Figure 6 and 7.

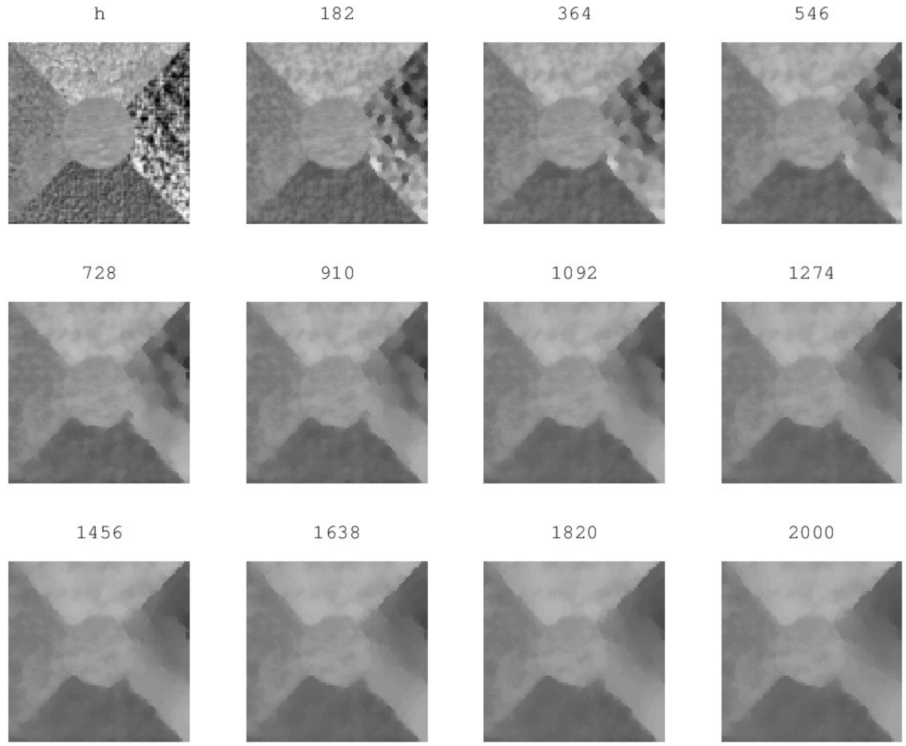

Figure 6. Change of h in time with α *= 0.005,* β*=1.0* ρ*=0.5*

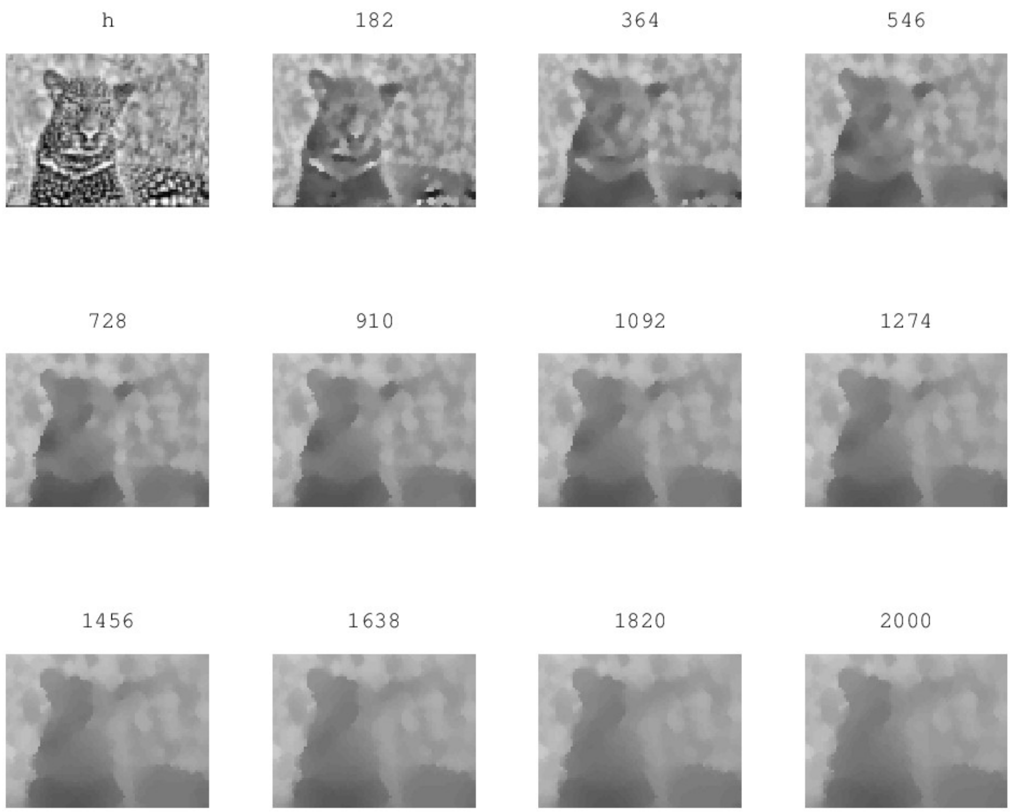

Figure 7. Stripping the texture away in a seemingly bottom up hierarchical manner (same parameters)

2.1.2.3. *Noise Removal*

Because h indirectly helps reducing the total variation, it can be used for roughly reducing noise (with some serious side effects).

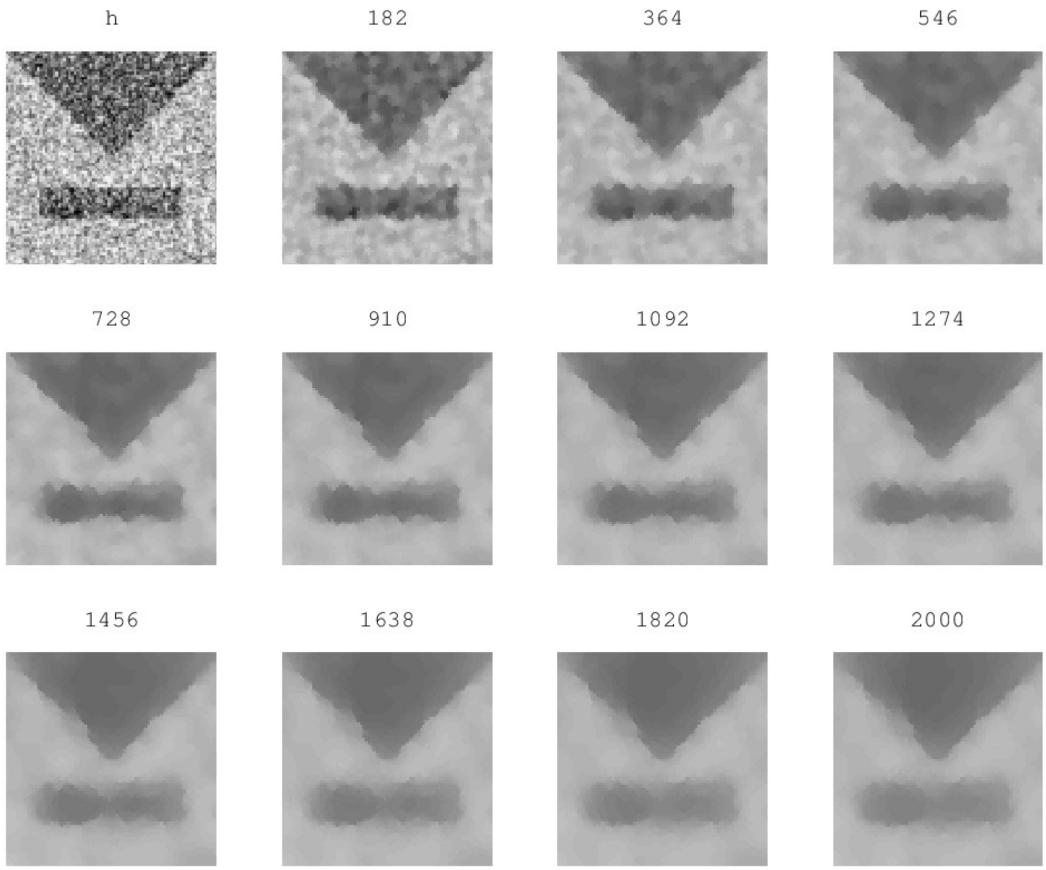

Figure 8. Change of h in time with $\alpha = 0.005$*,* $\beta=1.0$ $\rho=0.5$

Overall, it can be observed that h can be used for hierarchically removing textures with increasing scale. (The effect of stripping away textures is quite visible in Figure 6 and 7.) In Figure 8, although h has a fair shape preserving capability, *intensity in all regions gets closer and closer to the average intensity*, making the final image look paler than the original image. The edge strengthening effect and its slower diffusion rate can be exploited.

2.2. Modification 2: Cartoon Images are *Curvy*

I modified Formula 4 as below to gain further control on the temperature field and achieve a better regularization effect:

$$E_{AT}(u,v)=\int_{\Omega}\left[\frac{\beta}{\alpha\omega}|\nabla h|^2(u-g)^2+\left(\frac{(1-\omega)}{\omega}v^2|\nabla u|^2+h^2|\nabla u|^2\right)+\frac{1}{2\alpha\omega}\left(\rho\left(h^2|\nabla v|^2\right)+\frac{(1-v)^2}{\rho}\right)\right]dx \qquad (16)$$

The modification was made to take advantage of the fact (illustrated in Figure 4) that α and β can be adjusted so that the temperature field will tend to create relatively “colder” regions along the edges. By adding another component where high temperature along the gradient of u introduces high energy and low temperature along the edges is encouraged, aside from v, another way to control u becomes possible.

In the minimization process, temperature field will contribute to shape of u and important edges will still be preserved for a longer period of time compared to the more isolated or spiky edges (e.g. jitters, corners and salt&pepper noise) where they are more likely to have higher temperature (due to temperature diffusion). This also relaxes the edge-gradient consistency penalty, which would prevent jitters from disappearing.

Also note that, by adding this term, the formula achieves a better expressive power thanks to ω which allows one to adjust how much the gradient of u will depend on v and how much on h. Since changing gradient of u has a direct effect on v, changing ω also changes how much v depends on h. The whole system is described in terms of below PDEs:

$$\frac{\partial u}{\partial t}=\nabla.([(1-\omega)v^2+\omega h^2]\nabla u)-\frac{\beta}{\alpha}|\nabla h|^2(u-g) \tag{17}$$

$$\frac{\partial v}{\partial t}=\nabla.(h^2\nabla v)-\frac{(v-1)}{\rho^2}-\frac{2\alpha v(1-\omega)|\nabla u|^2}{\rho} \tag{18}$$

$$\frac{\partial h}{\partial t}=\nabla.((u-g)^2\nabla h)-\frac{h(\rho|\nabla v|^2+2\alpha\omega|\nabla u|^2)}{2\beta} \tag{19}$$

where ω is simply a small weighting factor between 0 and 1 to control the effect of the new modification.

2.2.1 Discretization of Modification 2

Semi-implicit discretization for Formula 17 is given by:

$$u^{k+1}-u^k=\frac{\Delta t}{2}[b^2.(Lu^k)-u^k.(Lb^2)+L(b^2.u^k)]-\frac{\Delta t\beta|\nabla h|^2(u^{k+1}-g)}{\alpha} \tag{20}$$

$$u^{k+1}=\left(1+\frac{\Delta t\beta|\nabla h|^2}{\alpha}\right)^{-1}\left[u^k+\frac{\Delta t}{2}[b^2.(Lu^k)-u^k.(Lb^2)+L(b^2.u^k)]+\frac{\Delta t\beta|\nabla h|^2 g}{\alpha}\right] \tag{21}$$

where $b=(1-\omega)v^2+\omega h^2$.

Semi-implicit discretization for Formula 18 is given by:

$$v^{k+1}-v^k=\Delta t\nabla.(h^2\nabla v^k)-\frac{\Delta t(v^{k+1}-1)}{\rho^2}-\frac{2\Delta t\alpha v^{k+1}(1-\omega)|\nabla u|^2}{\rho} \tag{22}$$

$$v^{k+1}=\left(1+\frac{\Delta t}{\rho^2}+\frac{2\Delta t\alpha(1-\omega)|\nabla u|^2}{\rho}\right)^{-1}\left[v^k+\Delta t\nabla.(h^2\nabla v^k)+\frac{\Delta t}{\rho^2}\right] \tag{23}$$

Semi-implicit discretization for Formula 19 is given by:

$$h^{k+1}-h^k=\Delta t\nabla.((u-g)^2\nabla h^k)-\frac{\Delta t h^{k+1}(\rho|\nabla v|^2+2\alpha\omega|\nabla u|^2)}{2\beta} \tag{24}$$

$$h^{k+1}=\left(1+\frac{\Delta t(\rho|\nabla v|^2+2\alpha\omega|\nabla u|^2)}{2\beta}\right)^{-1}\left[h^k+\Delta t\nabla.((u-g)^2\nabla h^k)\right] \tag{25}$$

2.2.2. Experiment for Modification 2

Aside from texture removal and edge enhancement skills, of which this model is still capable [4], new model seems to be fairly good at noise removal. Thanks to the parameter ω, a good trade-off between a temperature field that that can quickly get rid of undesired noise and jitters, and a u that preserves the features of the original image can be found. That is, an important issue with the original AT approximation was that, when one makes fidelity more important by adjusting α and β, the shapes of edges freezes in their constant form (jitter on the edges due to noise). Trying to get rid of the jitters however, results in important edges to quickly fade away. With the addition of ω, my approach overcomes this limitation as illustrated in Fig 9-11.

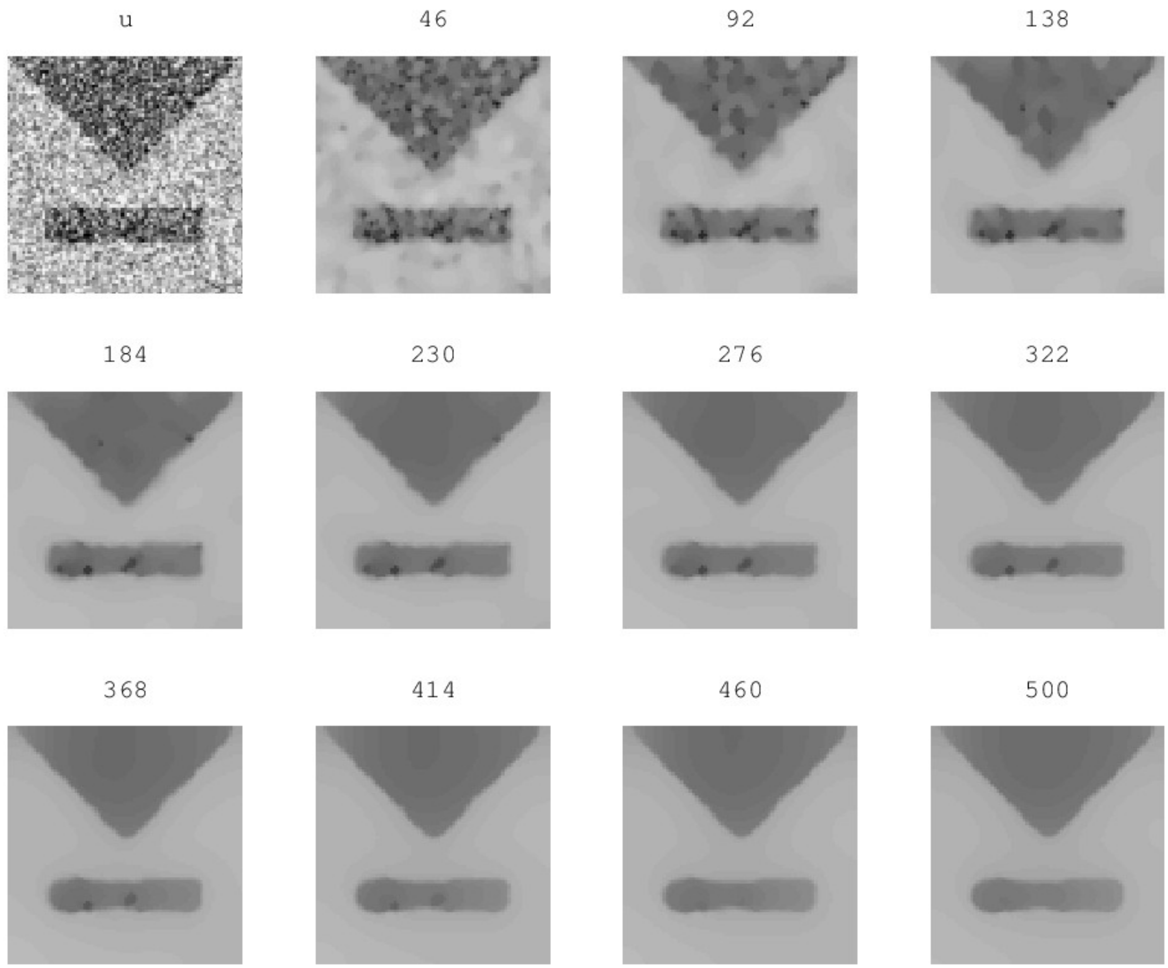

Figure 9. Change of u, using the modification 2, with α=10000, β=5000, rho=0.2, ω=0.2

Notice that, as a side effect, corners are lost (and the intensity of the image is ever slightly affected from the temperature field). However, when compared to the original AT (e.g. observe the jitters in Fig 2), the figure illustrates that, with the modification 2, a fair improvement to the original AT result can be achieved, since the desired effect of removing jitters was successful. In this example h was initialized as h=g. However, there may be a better way to initialize it. Yet, initializing h=1-g gives slightly worse results: the bar below the triangle gets paler and visibly more dilated (Fig 12-14). Also in the implementation, in each iteration, each update equation of the three PDEs were applied only once. A more efficient update scheme may be possible.

Overall, the idea of incorporating assumptions in the form of new fields may deserve a more rigorous investigation. After a literature survey, this idea will be further investigated in a research paper.

4 The proposed model in modification 2 is still capable of everything the model in modification 1 can do, since when ω is extremely small (and α is increased accordingly), the two models become equivalent.

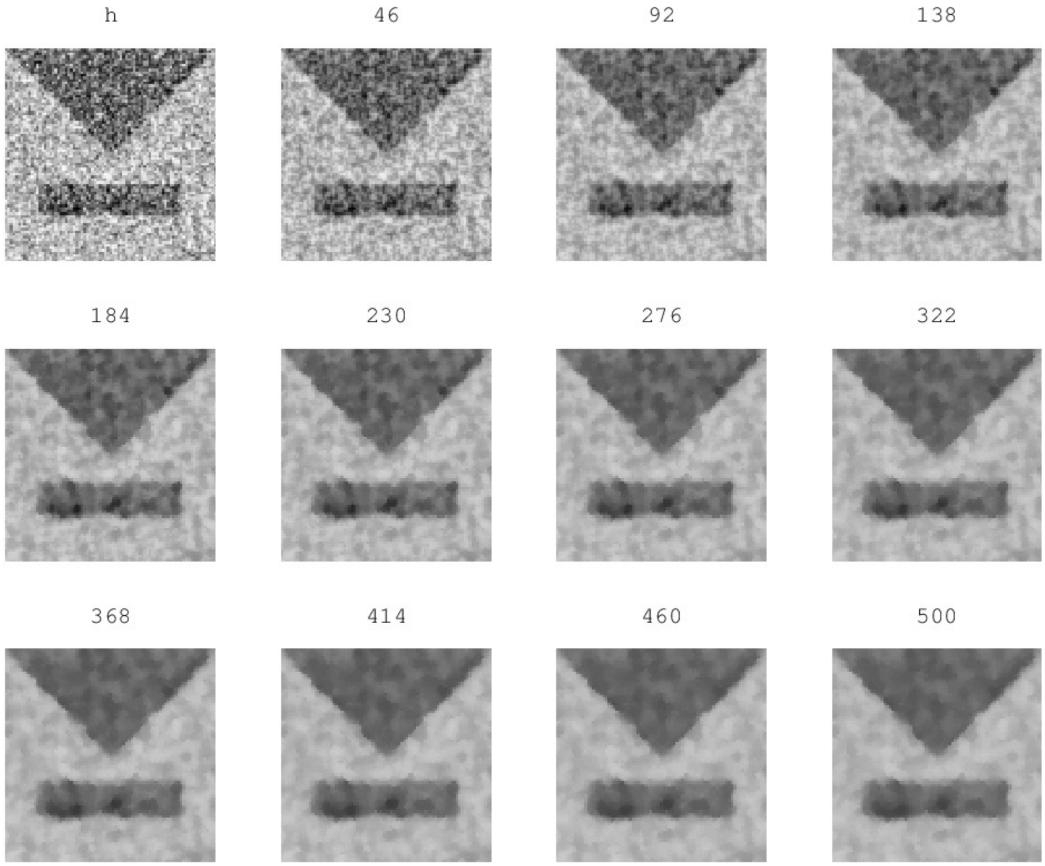

Figure 10. Change of h, using the modification 2 (same parameters)

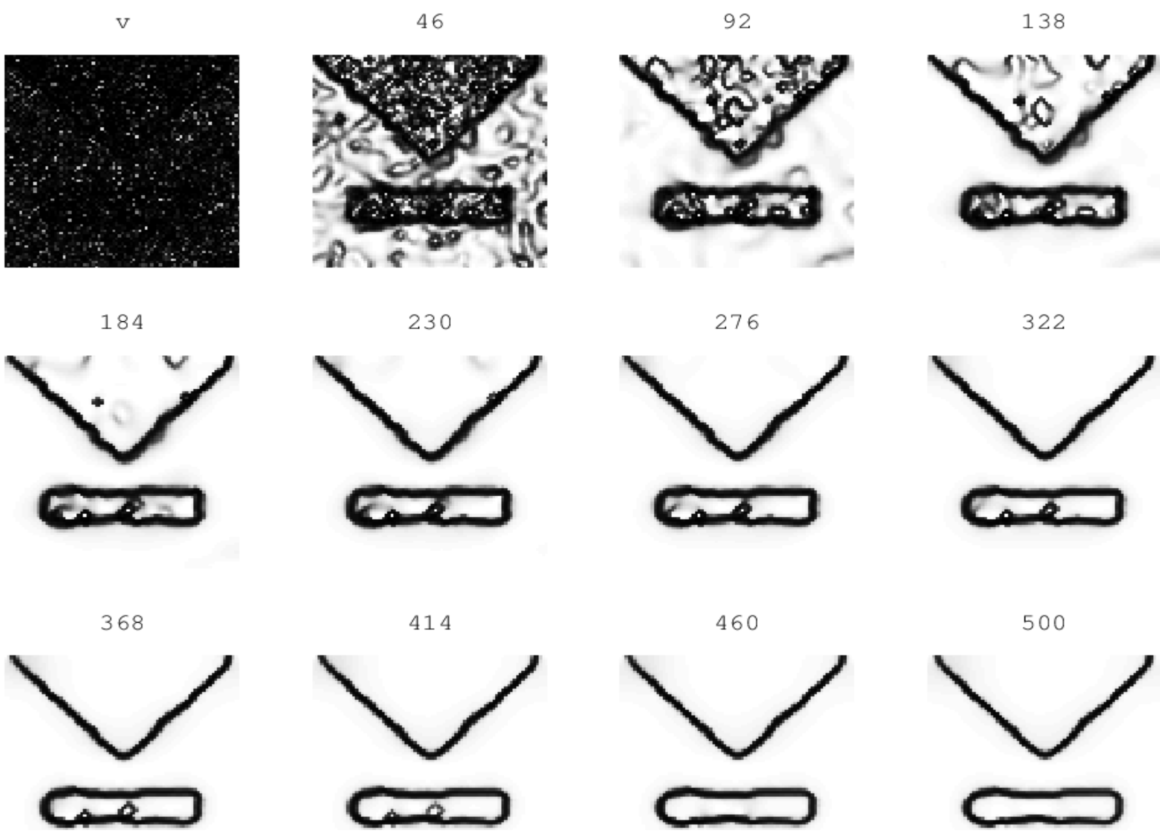

Figure 11. Change of v, using the modification 2 (same parameters)

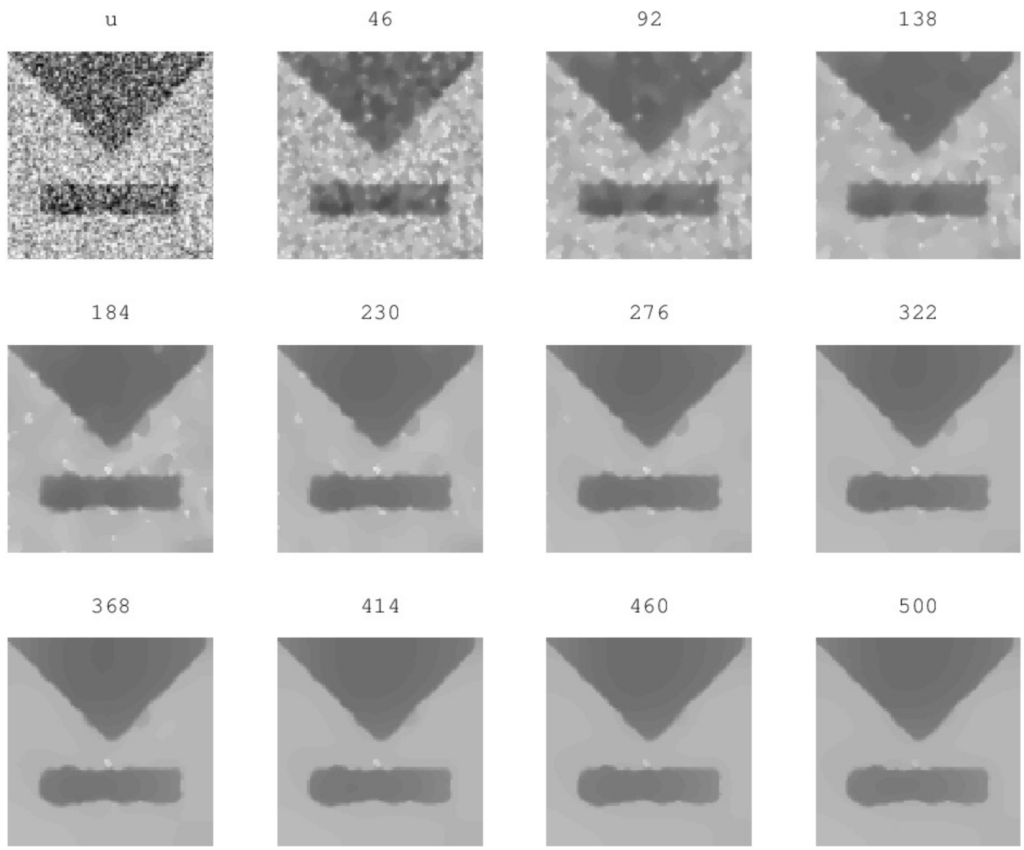

Figure 12. Change of u, using the modification 2 (same parameters but h is initialized as 1-g)

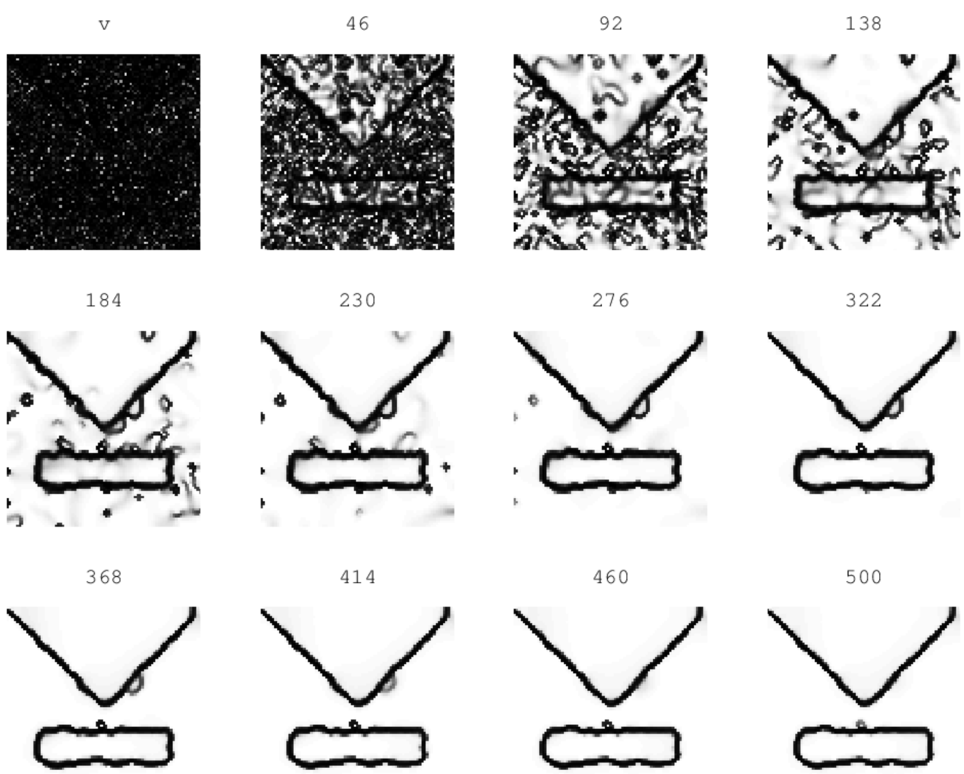

Figure 13. Change of v, using the modification 2 (same parameters but h is initialized as 1-g)

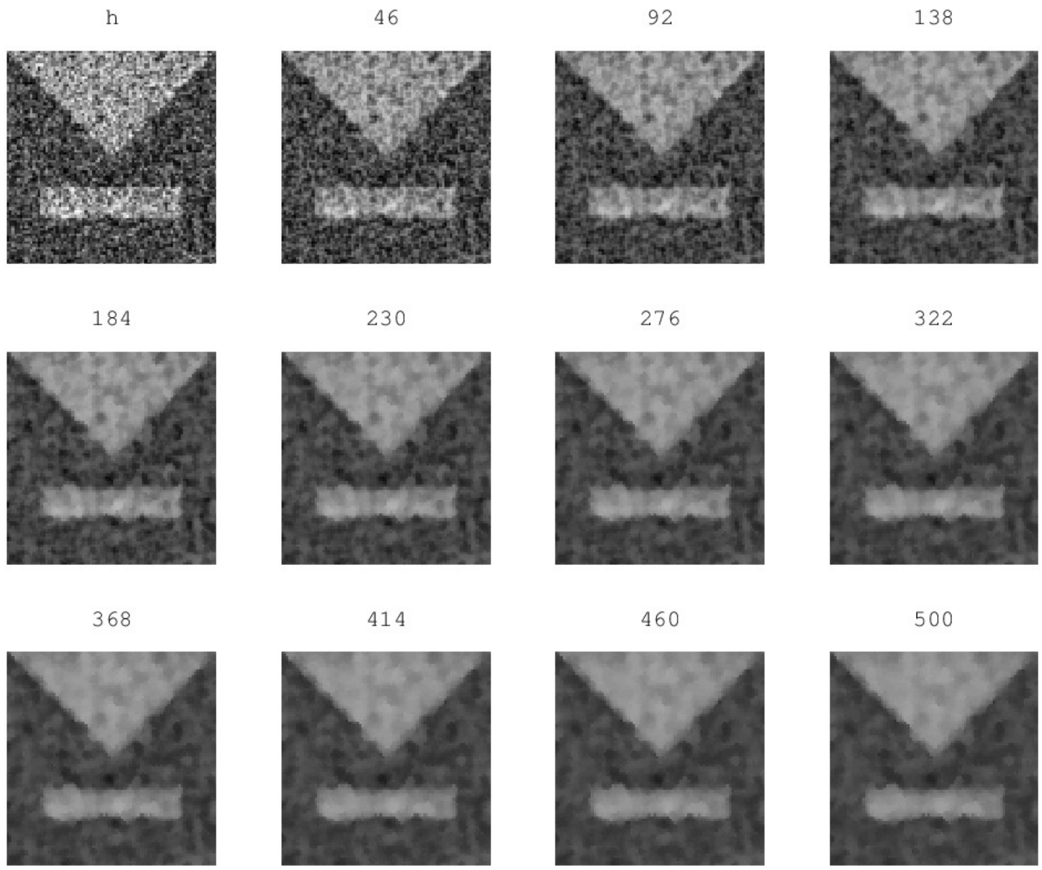

Figure 14. Change of h, using the modification 2 (same parameters but h is initialized as 1-g)

The reason why u dilates is that, since this time h is initialized as 1-g, jitters or spiky regions (pointing outside) in the edges becomes actually colder than the more regular regions on the edges. Therefore, those regions diffuse faster, filling up the gaps between the spikes, causing dilation (Notice the bar which becomes slightly larger than the original image. Also notice that in Fig 9 the size of the bar is slightly smaller, since in the case that h initialized as g, spiky regions diffuse faster as predicted in subsection 2.2.)