

# Project 2: Anisotropic Diffusion

Manasi Datar

## 1. Overview

In this project we explore the process of anisotropic diffusion through the work of Perona & Malik. This work introduces a slight modification to the conventional diffusion process by modeling the flux as a function of edge-strength in the image, thereby giving us "anisotropy". This modification is significant since it allows the diffusion process to have knowledge of the region boundaries and thus preserve them. In fact, we saw in class that such a process may even enhance region boundaries in effect !

We also explore the construction of a nonlinear (anisotropic) scale-space and comment on its usefulness.

The following sections discuss the process of isotropic diffusion and introduce the change required to obtain the anisotropic version. We also look at results of applying anisotropic diffusion on multiple images and inspect the nonlinear scale-space with an example. We conclude with a few remarks relating back to the lecture by James Fishbaugh, and comments on the implementation.

## 2. Isotropic Diffusion

The diffusion equation is a general case of the heat equation that describes the density changes in a material undergoing diffusion over time. Isotropic diffusion, in image processing parlance, is an instance of the heat equation as a partial differential equation (PDE), given as:

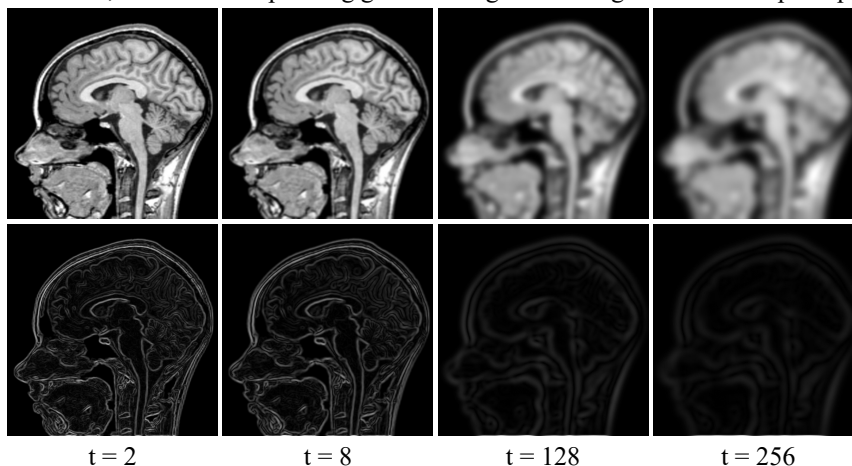
$$\frac{\partial I}{\partial t} = \nabla^2 I = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}$$

where,  $I$  is the image and  $t$  is the time of evolution.

Investigation of the above equation gives an intuition that solving this for an image is equivalent to convolution with some Gaussian kernel. This fact enables us to look at isotropic diffusion as a means of constructing a linear scale-space. As described in the project description, one way to solve the heat equation in the discrete domain is the Forward Time Central Space (FTCS) method. The FTCS solution basically adds the result of a discrete convolution (with a local gradient window) back to the original image after scaling it with a certain factor. This can be elaborated as:

$$I_{i,j}^{t+1} = I_{i,j}^t + \lambda [I_{i-1,j}^t + I_{i+1,j}^t + I_{i,j-1}^t + I_{i,j+1}^t - 4I_{i,j}^t]$$

The stability of such a solution depends on the value of  $\lambda$  and Perona-Malik suggest the range  $[0,0.25]$  for a stable solution. I have chosen to use a constant value of  $\lambda=0.25$  for all my results here. The following figure shows the results of isotropic diffusion, and the corresponding gradient magnitude images for an example input:



We can notice that while the diffusion process blurs the image considerably as the number of iterations increases, the

edge information progressively degrades as well. This is not a desirable effect and exactly the drawback that Perona & Malik address in their work.

### 3. Anisotropic Diffusion

Perona & Malik introduce the flux function as a means to constrain the diffusion process to contiguous homogeneous regions, but not cross region boundaries. The heat equation (after appropriate expansion of terms) is thus modified to:

$$\frac{\partial I}{\partial t} = c(x, y, t) \Delta I + \nabla c \cdot \nabla I$$

where  $c$  is the proposed flux function which controls the rate of diffusion at any point in the image.

A choice of  $c$  such that it follows the gradient magnitude at the point enables us to restrain the diffusion process as we approach region boundaries. As we approach edges in the image, the flux function may trigger inverse diffusion and actually enhance the edges !

Perona & Malik suggest the following two flux functions:

$$c(\|\nabla I\|) = e^{-(\|\nabla I\|/K)^2}$$

$$c(\|\nabla I\|) = \frac{1}{1 + \left(\frac{\|\nabla I\|}{K}\right)^2}$$

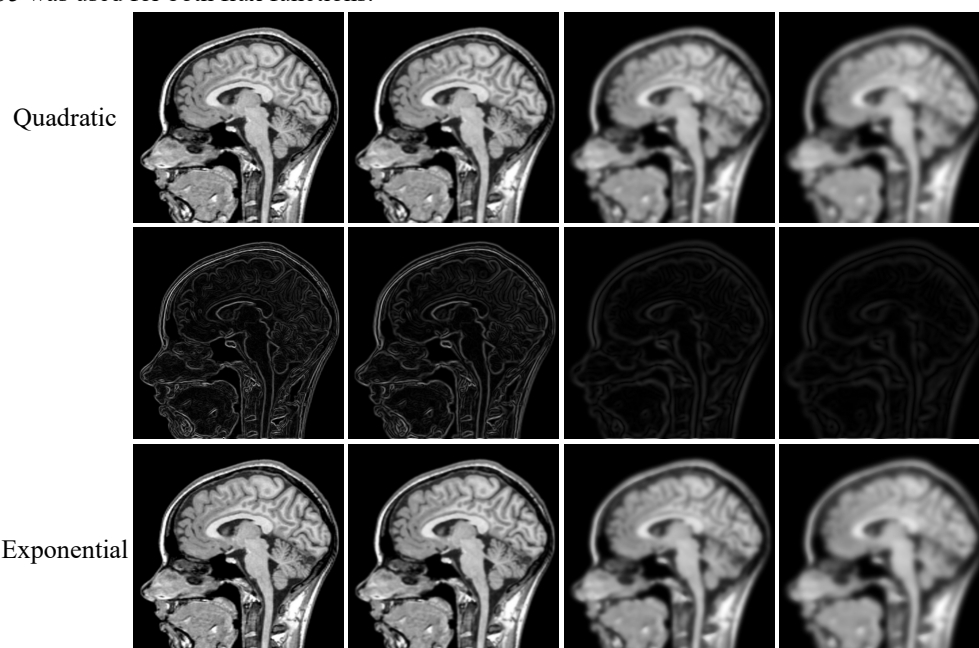
The flux functions offer a trade-off between edge-preservation and blurring (smoothing) homogeneous regions. Both the functions are governed by the free parameter  $\kappa$  which determines the edge-strength to consider as a valid region boundary. Intuitively, a large value of  $\kappa$  will lead back into an isotropic-like solution.

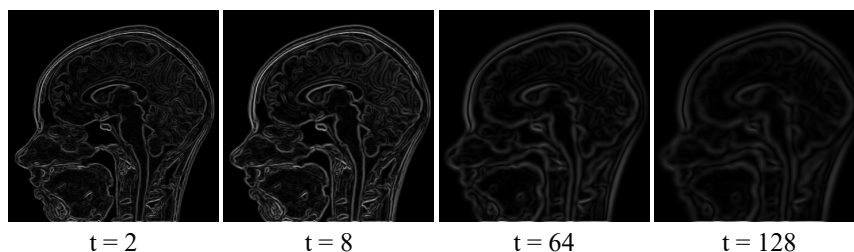
We will experiment with both the flux functions in this report.

A discrete numerical solution can be derived for the anisotropic case using the FTCS method as follows:

$$I_{i,j}^{t+1} = I_{i,j}^t + \lambda [c_N \cdot \nabla_N I + c_S \cdot \nabla_S I + c_E \cdot \nabla_E I + c_W \cdot \nabla_W I]_{i,j}^t$$

where  $\{N, S, W, E\}$  correspond to the pixel above, below, left and right of the pixel under consideration  $(i, j)$ . Another nice property is that  $c$  is based on the gradient magnitude, thus it does not matter if we take forward or backward gradients ! The following table displays the results of the anisotropic diffusion process for the same example image as above.  $\kappa=0.35$  was used for both flux functions.



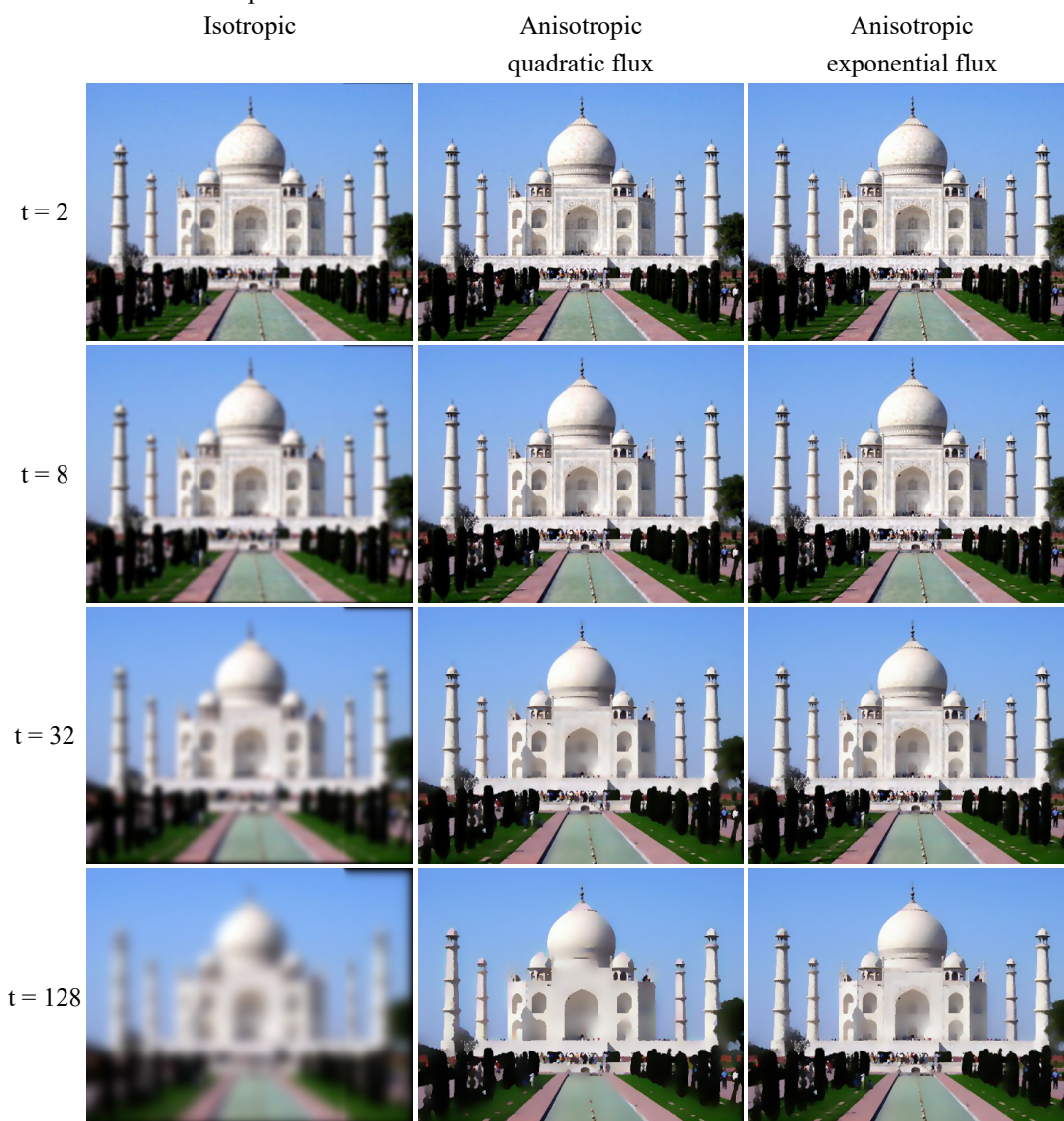


As seen in the above table, the flux functions behave differently as the diffusion progresses and can lead to interesting choices based on the application at hand.

## 4. "Anisotropic" Scale Space

Similar to the intuition that isotropic diffusion is akin to Gaussian blurring, we can imagine that anisotropic diffusion is similar to blurring by a weighted kernel. However, this kernel is not as straightforward as the Gaussian due to the free parameter  $\kappa$ . We can think of creating a nonlinear scale space based on such a kernel, but the ideal scale space will have to be determined using heuristics.

The following table tries to give a sense of what the nonlinear scale spaces would look like. The gradient magnitude maps are also included for comparison.



$\kappa$  was set to 0.05 for this experiment.

The above table highlights the advantages of the nonlinear scale-space. We can see that the isotropic diffusion process quickly gets rid of essential details including edge information, whereas the anisotropic version works well with both the flux functions.

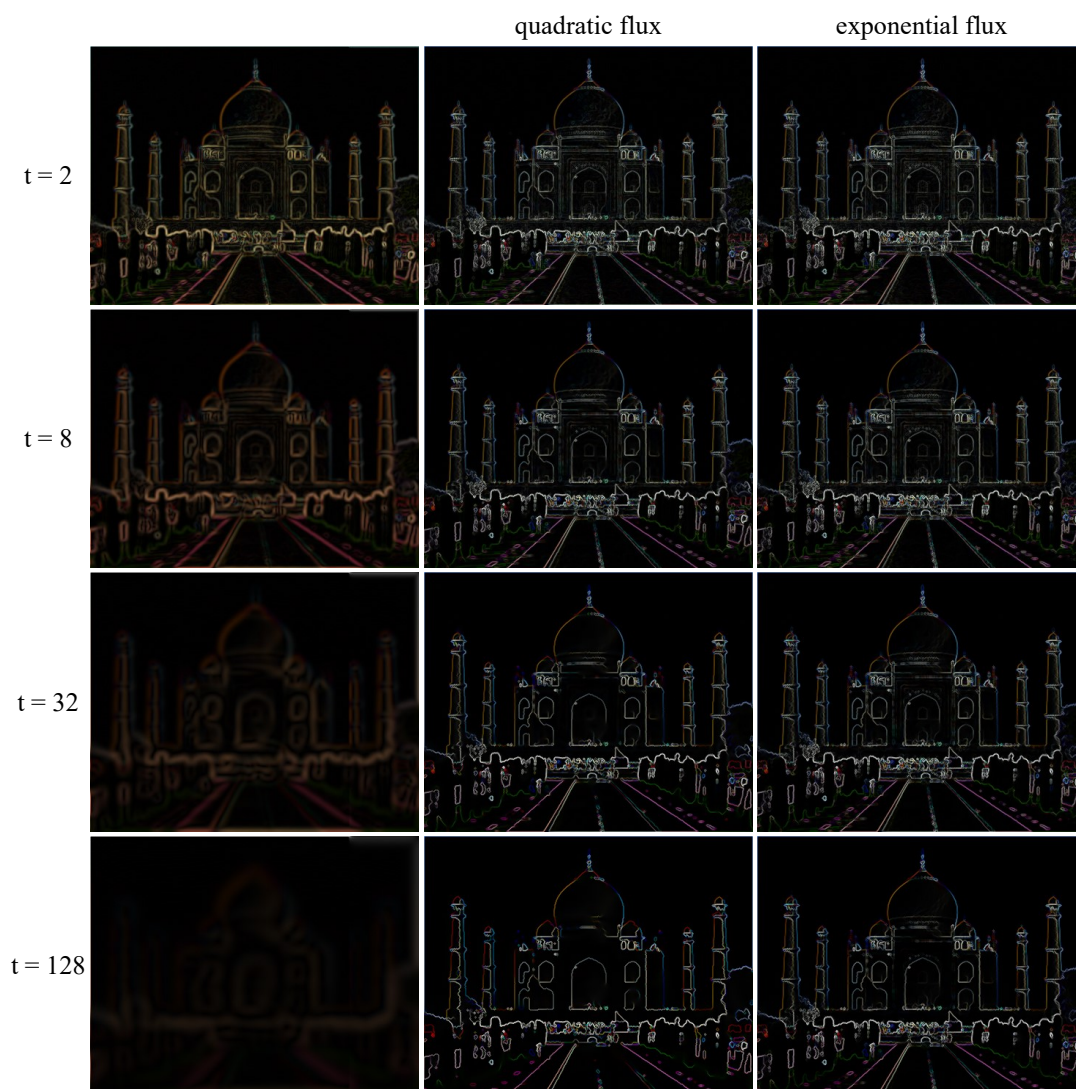
The table below includes the corresponding gradient magnitude images to further improve the understanding of the process.

Isotropic

Anisotropic

Anisotropic



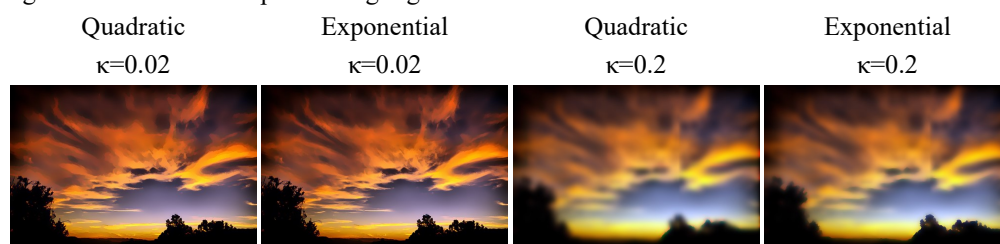


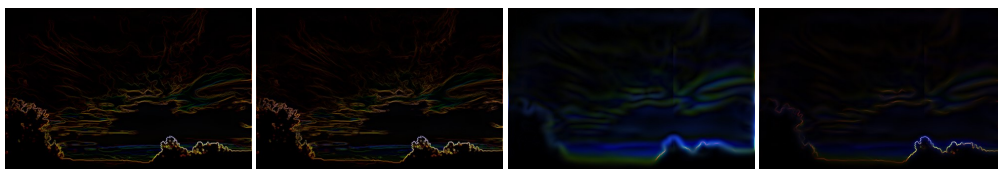
The gradient magnitude images display the effect of the lux function more clearly. As we traverse along the scale-space, gradient magnitude decays faster due to the isotropic diffusion process. This is evidence of the fact that this process has no information about the edges present. On the other hand, the edges are preserved upto much higher scales in the anisotropic process. Also noticeable is the fact that stronger edges last longer than weaker edges. The effect of inverse diffusion is seen at higher scales. Specially in the case of the exponential flux function which favors edge preservation over smoothing. This is evident in the progressive sharpening of the vertical edges along the minarets and also the steeple atop the Taj Mahal.

## 5. Results

Along with the number of iterations, the free parameter  $\kappa$  is very influential in the application of the diffusion process. As we can see from the results below,  $\kappa$  can be used to provide knowledge about the granularity of regions (or edge-strengths) we wish to preserve. As such, the choice of parameters for any application using this process (e.g. segmentation) would be dependent on the images at hand.

The following table shows an example that highlights this intuition.





In the above table, we can see how the choice of  $\kappa$  affects the result. A higher value of  $\kappa$  implies that only wider edges are preserved, while edges denoting detail (intuitively, those at finer scales) are smoothed away. The above table also indicates the opposite nature of the two flux functions. The gradient magnitude images obtained using the Quadratic flux function preserve lesser detail along the edges, but provide smoother regions, as compared to the results obtained using the Exponential flux function using the same number of iterations and  $\kappa$  value. This fact can be useful in the appropriate choice of the flux function for an actual application. One such application could be the denoising of MRI images. An example of the knee with the optimal smoothing is shown below:



The result above was generated using anisotropic diffusion with an Exponential flux function,  $t=16$  and  $\kappa=0.2$ . This result highlights the possible applications of anisotropic smoothing, as also mentioned in the Ultrasound speckle noise cancellation filter discussed in class.

## 6. Conclusion

Various experiments with nonlinear scale-space suggest that it can be a powerful tool for various applications. However, linear (Gaussian) scale-spaces are not far behind. The inherent properties of the Gaussian kernel make it convenient to use and easy to understand. On the other hand, nonlinear scale-spaces allow for the inclusion of knowledge about local criteria such as edge-strength to smooth homogeneous regions, while preserving the boundaries.

One way to improve the process of anisotropic diffusion would be to automate the process of computing  $\kappa$  values. Since this value is related to the edge-strength, the obvious choice seems to be to examine the density function of the image gradient. We could look at peaks in this function to identify regions at various levels of coarseness to be preserved, in some sense a level-of-detail (LOD) type approach. Another naive, yet probably effective method would be to simply look at the histogram and choose the value of  $\kappa$  based on the edge-strength we wish to preserve.

Nonetheless, even without these improvements, it is clear that anisotropic diffusion and nonlinear scale-space is a valuable tool in the image processing toolkit.

## 7. Implementation Notes

Continuing my exploration of the VISPack library, I implemented the above methods using C++ and VISPack image processing operations. The [code](#) can be compiled using CMake on multiple compilers. I tried to be as efficient as possible in the implementation of the actual diffusion equation solution, but lost out on user-friendliness in the process. As such, image I/O and working with the various formats turned out to be quite tedious. This will definitely figure more prominently in my next submission.