Table of contents

Chromatic Aberration

Chromatic Aberration (CA) is an optical defect present in camera lenses due to the differences in the path taken by light rays. When travelling through the different lenses constituting the objective, the light rays of various colors will be refracted differently, according to their wavelength. A red (620-750nm) ray will be less refracted than a blue (450-495nm) ray.

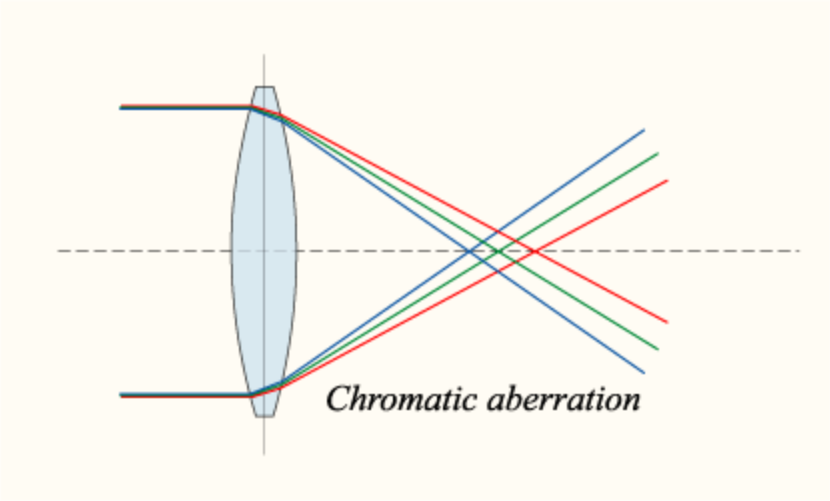


Figure 1: Refractions for Red, Green and Blue rays. Source: Wikipedia.

Because of this, the camera sensor will not be aligned with focal plane of the lenses for every color (we consider Red, Green and Blue only). If we suppose that the focus has been setup such that the green focus plane is aligned with the sensor plane, the red and blue color will slightly be out of focus, or translated.

CA results in fringes of magenta or cyan colors as a color channel (R, G or B) is scaled up/down or translated relative to the others. A picture can suffer from two types of CA: axial and transverse.

Axial Chromatic Aberration (ACA)

ACA is the result of the lens assembly having distinct focal planes for each color. The focal points are aligned with the optical axis, thus the name Axial CA. It causes the R and G channels of the image to be out of focus, considering that the G channel is in focus. It is difficult to correct in post-processing since a loss of information occurred at the capture of the image.

ACA occurs particularly at small apertures.

Figure 2: Example of Axial CA. Green and purple fringing are visible at the limit of the in-focus area. Source: Wikipedia

The optical assembly in Figure 1 will exhibits Axial CA.

Transverse Chromatic Aberration (TCA)

A lens subject to TCA has a unique focal plane for the R, G and B channels, but the focus point is at different positions on the focal plane. The result is different magnifications for each color channel. For example, the Red channel would be zoomed in in comparison to the Green channel, and the Blue channel zoomed out. No information is lost in this image as all the image is in focus. Scaling up or down the channels independently usually corrects TCA.

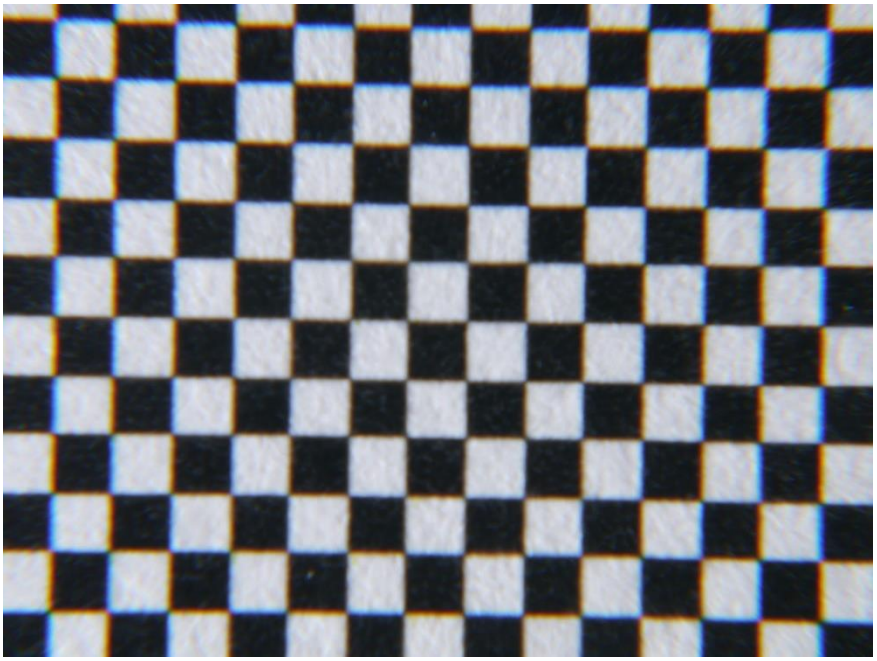


Figure 3: Example of Transverse CA. R is scaled up radially, B is scaled down radially.

ACA and TCA can concurrently affect an image and renders CA correction very hard. The objective of this project is to correct both types of CA with the same technique.

PDE approach to CA correction

By decomposing an image in 3 color channels R, G and B, and using the G channel as a reference, the idea is to find the offset (in x and y) that each pixel underwent due to CA.

In order to attract image features of the R and B channel to the corresponding feature in the G layer, two approaches are possible. First, one can simply compare the R intensity of a pixel with its G intensity. Secondly, because CA is present in high contrast areas of the image, one can use the edges (R and G gradient) for the comparison.

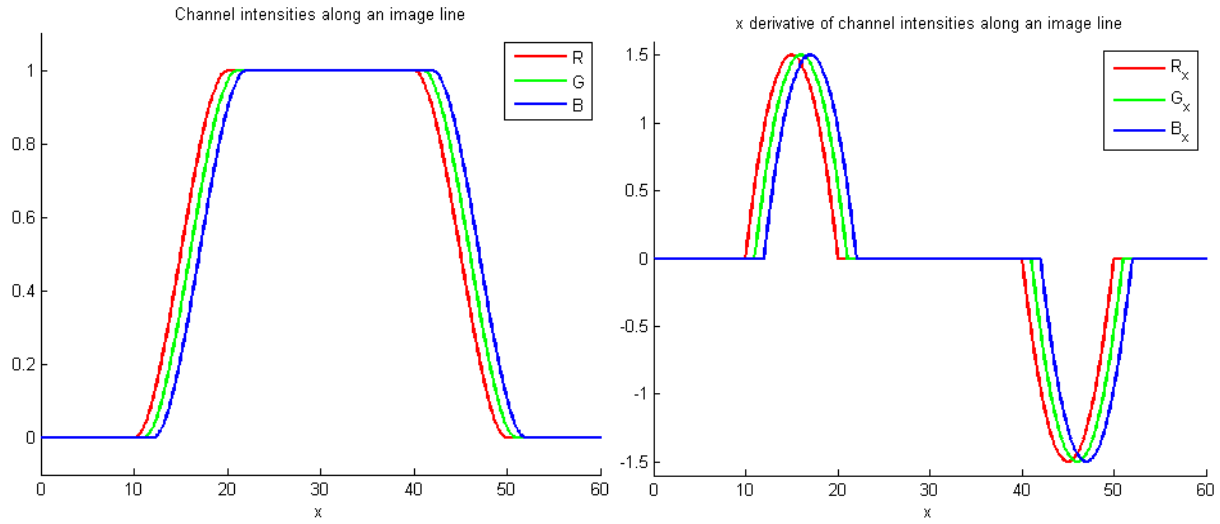


Figure 4: Channel intensities and x-derivatives for a black/white/black transition.

Intensity-based energy

Energy definition

We denote the 3 color channel intensities $R(x,y)$, $G(x,y)$ and $B(x,y)$ and define them as functions from $\Omega = [\![1,\ M]\!] \times [\![1,\ N]\!]$ to $[0,1]$. Where M, N are respectively the width and length of the image.

I used a variational gradient descent technique by defining an energy penalizing differences in G-intensity from one pixel at (x,y), namely $G(x,y)$ with the R-intensity of an offset pixel at $(x+u, y+v)$, namely $R(x+u, y+v)$. We suppose here the effect of the CA independent for the R and B channel.

Where we defined $u: \Omega \to \mathbb{Z}$ and $v: \Omega \to \mathbb{Z}$. Respectively the x-offset and y-offset due to CA.

Let's define the energy functional:

$$E_I(u,v) = \frac{1}{2}\int_\Omega |R(x+u,y+v) - G(x,y)|^2\, dxdy + \frac{1}{2}\int_\Omega \|\nabla u\|^2\, dxdy + \frac{1}{2}\int_\Omega \|\nabla v\|^2\, dxdy$$

The first part of this energy penalizes following the rule stated above. The second part ensures that the x and y offsets stays smooth (low gradient). Weighting coefficients $\alpha \geq 0, \beta \geq 0$, are added to allow for adjustments between the two parts:

$$E_I(u,v) = \frac{\alpha}{2}\int_\Omega |R(x+u,y+v) - G(x,y)|^2\, dxdy + \frac{\beta}{2}\left[\int_\Omega \|\nabla u\|^2\, dxdy + \int_\Omega \|\nabla v\|^2\, dxdy\right] \qquad (1a)$$

Looking at the left plot of Figure 4, we can see that if we offset positively the red channel by one pixel, the first part of the energy is minimized. After minimizing for R, we employ the same technique for the B channel.

$$E_I(u,v) = \frac{\alpha}{2}\int_\Omega |B(x+u,y+v) - G(x,y)|^2\, dxdy + \frac{\beta}{2}\left[\int_\Omega \|\nabla u\|^2\, dxdy + \int_\Omega \|\nabla v\|^2\, dxdy\right] \qquad (1b)$$

If we offset negatively the blue channel, we minimize the first part of the energy.

Gradient descent PDE

In order to determine the optimal values for u and v for all pixels, I employed a gradient descent approach.

$$\begin{cases} \nabla_u E_I(u,v) = L_u - \dfrac{\partial}{\partial x}L_{u_x} - \dfrac{\partial}{\partial y}L_{u_y} \\ \nabla_v E_I(u,v) = L_v - \dfrac{\partial}{\partial x}L_{v_x} - \dfrac{\partial}{\partial y}L_{v_y} \end{cases}$$

Where

$$E_I(u,v) = \int_\Omega L(u,v,\nabla u,\nabla v,x,y)\, dxdy$$

$$L(u,v,\nabla u,\nabla v,x,y) = \frac{\alpha}{2}|R(x+u,y+v) - G(x,y)|^2 + \frac{\beta}{2}(\|\nabla u\|^2 + \|\nabla v\|^2)$$

We have:

$$\begin{aligned} L_u &= \alpha\big(R(x+u,y+v) - G(x,y)\big)R_x(x+u,y+v) \\ L_v &= \alpha\big(R(x+u,y+v) - G(x,y)\big)R_y(x+u,y+v) \\ L_{u_x} &= \beta u_x, \quad L_{u_y} = \beta u_y, \quad L_{v_x} = \beta v_x, \quad L_{v_y} = \beta v_y \end{aligned}$$

$$\begin{cases} \nabla_u E_I(u,v) = \alpha\big(R(x+u,y+v) - G(x,y)\big)R_x(x+u,y+v) - \beta\Delta u \\ \nabla_v E_I(u,v) = \alpha\big(R(x+u,y+v) - G(x,y)\big)R_y(x+u,y+v) - \beta\Delta v \end{cases} \qquad (2)$$

The evolution of u and v go directly against the gradient (steepest descent):

$$\begin{cases} u_t = -\alpha\big(R(x+u,y+v) - G(x,y)\big)R_x(x+u,y+v) + \beta\Delta u \\ v_t = -\alpha\big(R(x+u,y+v) - G(x,y)\big)R_y(x+u,y+v) + \beta\Delta v \end{cases} \quad (3)$$

Both functions are evolved simultaneously.

Local linearization for discretization

These gradient descent PDEs are highly nonlinear as the relation between u_t (resp. v_t) and u (resp. v) is dependent on the data R issued from the image.

On the hypothesis that the image is smooth enough, we can approximate these PDEs at the first order by:

$$\begin{cases} u_t = -\alpha\left(R(x,y) + uR_x(x,y) + vR_y(x,y)\right)R_x(x+u,y+v) + \beta\Delta u \\ v_t = -\alpha\left(R(x,y) + uR_x(x,y) + vR_y(x,y)\right)R_y(x+u,y+v) + \beta\Delta v \end{cases}$$

Or,

$$\begin{pmatrix} u \\ v \end{pmatrix}_t = -\alpha\begin{pmatrix} R_x(x,y)R_x(x+u,y+v) & R_y(x,y)R_x(x+u,y+v) \\ R_x(x,y)R_y(x+u,y+v) & R_y(x,y)R_y(x+u,y+v) \end{pmatrix}\begin{pmatrix} u \\ v \end{pmatrix} + \beta\begin{pmatrix} \Delta u \\ \Delta v \end{pmatrix} \quad (4)$$

This PDE is locally behaving like a 2D reaction-diffusion PDE.

Gradient-based energy

Energy definition

The previous method has the disadvantage of working very well for grayscale images (where $R = G = B$ before CA alteration), but may not converge for color images as no offset (u, v) will satisfy $R(x+u, y+v) = G(x,y)$.

Using the gradient of the color channels allows the algorithm to ignore the absolute intensities while still being able to find features (edges, color gradient, …).

A similar energy to (1) can be defined, but penalizing the norm of the gradient difference:

$$E_G(u,v) = \frac{\alpha}{2}\int_\Omega \|\nabla R(x+u,y+v) - \nabla G(x,y)\|^2\, dxdy + \frac{\beta}{2}\left[\int_\Omega \|\nabla u\|^2\, dxdy + \int_\Omega \|\nabla v\|^2\, dxdy\right] \quad (5)$$

As in the intensity-based energy, a second pass for the B channel is computed independently from the first one for R.

Gradient descent PDE

In order to determine the optimal values for u and v for all pixels, I employed a gradient descent approach.

$$\begin{cases} \nabla_u E_G(u,v) = L_u - \dfrac{\partial}{\partial x} L_{u_x} - \dfrac{\partial}{\partial y} L_{u_y} \\ \nabla_v E_G(u,v) = L_v - \dfrac{\partial}{\partial x} L_{v_x} - \dfrac{\partial}{\partial y} L_{v_y} \end{cases}$$

Where

$$E_G(u,v) = \int_\Omega L(u,v,\nabla u,\nabla v,x,y)\, dxdy$$

$$L(u,v,\nabla u,\nabla v,x,y) = \frac{\alpha}{2}\|\nabla R(x+u,y+v) - \nabla G(x,y)\|^2 + \frac{\beta}{2}(\|\nabla u\|^2 + \|\nabla v\|^2)$$

We have:

$$\begin{aligned} L_u &= \alpha\left((R_x - G_x)R_{xx} + (R_y - G_y)R_{xy}\right) \\ L_v &= \alpha\left((R_x - G_x)R_{yx} + (R_y - G_y)R_{yy}\right) \\ L_{u_x} &= \beta u_x, \quad L_{u_y} = \beta u_y, \quad L_{v_x} = \beta v_x, \quad L_{v_y} = \beta v_y \end{aligned}$$

$$\begin{cases} \nabla_u E_I(u,v) = \alpha\big(R(x+u,y+v) - G(x,y)\big)R_x(x+u,y+v) - \beta\Delta u \\ \nabla_v E_I(u,v) = \alpha\big(R(x+u,y+v) - G(x,y)\big)R_y(x+u,y+v) - \beta\Delta v \end{cases} \quad (2)$$

Where derivatives of R are evaluated at the point $(x+u, y+v)$ and the derivative of G at (x,y).

The evolution of u and v go directly against the gradient (steepest descent):

$$\begin{cases} u_t = -\alpha\left((R_x - G_x)R_{xx} + (R_y - G_y)R_{xy}\right) + \beta\Delta u \\ v_t = -\alpha\left((R_x - G_x)R_{yx} + (R_y - G_y)R_{yy}\right) + \beta\Delta v \end{cases} \quad (6)$$

Both functions are evolved simultaneously.

Lack of time forced me to abandon this method in order to focus on the intensity-based approach.

PDE discretization

Recall the local linearization of the intensity-based energy gradient descent:

$$\begin{pmatrix} u \\ v \end{pmatrix}_t = -\alpha \begin{pmatrix} R_x(x,y)R_x(x+u,y+v) & R_y(x,y)R_x(x+u,y+v) \\ R_x(x,y)R_y(x+u,y+v) & R_y(x,y)R_y(x+u,y+v) \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} + \beta \begin{pmatrix} \Delta u \\ \Delta v \end{pmatrix}$$

Discretization scheme

The discretization scheme I used is Forward Time Central Space (FTCS). This choice is justified by the presence of both a reaction and diffusion term. Both terms are stable using this scheme, as long as their CFL condition are respected.

Forward Time

This is the discretization scheme used:

$$u(x,y,t+\Delta t) \approx u(x,y,t) + \Delta t \times u_t(x,y,t)$$

$$v(x,y,t+\Delta t) \approx v(x,y,t) + \Delta t \times v_t(x,y,t)$$

Where Δt is the time step and Δt satisfies the PDEs CFL conditions.

Central Space

Several space derivatives have to be discretized. Central space is used. We define $\Delta x = \Delta y = 1$ as the spatial grid spacing, in pixels. For a function $f(x,y)$, we have:

$$f_x(x,y) \approx \frac{f(x+\Delta x,y) - f(x-\Delta x,y)}{2\Delta x}$$

$$f_y(x,y) \approx \frac{f(x,y+\Delta y) - f(x,y-\Delta y)}{2\Delta y}$$

Second order derivatives:

$$f_{xx}(x,y) = \frac{f(x+\Delta x,y) - 2f(x,y,t) + f(x-\Delta x,y)}{\Delta x^2}$$

$$f_{yy}(x,y) = \frac{f(x,y+\Delta y) - 2f(x,y,t) + f(x,y-\Delta y)}{\Delta y^2}$$

Second order crossed derivatives are not useful in our case.

Stability and CFL conditions

Given the complexity of the PDEs and its high dependency on the image data, I chose to separately consider the reaction term and the diffusion term. If both CFL conditions are satisfied, the CFL condition for our PDE will also be satisfied (under the data smoothness hypothesis).

Diffusion term

$$u_t = \beta \Delta u, \quad v_t = \beta \Delta v$$

These PDEs are two identical 2D linear heat equations. Their CFL condition is:

$$\beta \Delta t \leq \frac{\Delta x^2}{4}, \qquad \beta \geq 0$$

Reaction term

$$\begin{pmatrix} u \\ v \end{pmatrix}_t = -\alpha \underbrace{\begin{pmatrix} R_x(x,y)R_x(x+u,y+v) & R_y(x,y)R_x(x+u,y+v) \\ R_x(x,y)R_y(x+u,y+v) & R_y(x,y)R_y(x+u,y+v) \end{pmatrix}}_{M} \begin{pmatrix} u \\ v \end{pmatrix}$$

Both PDEs are linked via the matrix M. The CFL condition for a 1D reaction PDE of reaction coefficient $\lambda \leq 0$ is:

$$|\lambda| \Delta t \leq 2$$

Here, the reaction is 2D, and happens in the directions of M's eigenvectors, with its eigenvalues for coefficients.

Define $eig(M) = \{\lambda_1, \lambda_2\}$.

The CFL conditions are:

$$\alpha \lambda_1 \Delta t \leq 2, \qquad \lambda_1 \geq 0$$

$$\alpha \lambda_2 \Delta t \leq 2, \qquad \lambda_2 \geq 0$$

Final CFL condition

The choice of Δt is made by evaluating satisfying these 3 CFL conditions for all pixels in the image:

$$\beta \Delta t \leq \frac{\Delta x^2}{4}$$

$$\alpha \lambda_1 \Delta t \leq 2$$

$$\alpha \lambda_2 \Delta t \leq 2$$

Results

Test images

I used two test images displaying severe CA. Both images are pictures of a black and white subject and principally grayscale. The first image is subject to Axial CA:

Figure 5: Test image 1 (250x250px). Presents severe Axial CA.

The second image is affected by Transverse CA:

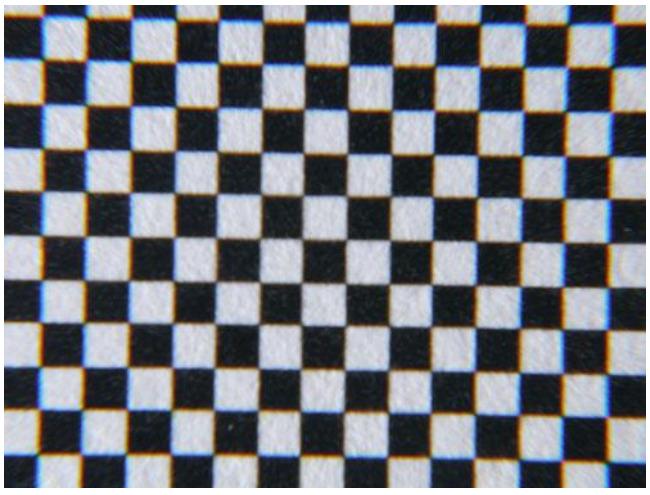


Figure 6: Test image 2 (500x376px). Presents severe Transverse CA.

Score comparison

In order to compare the efficiency of different methods, I defined a score function for grayscale images.

$$score(R,G,B) = \int_{\Omega} (|R(x,y) - G(x,y)|^2 + |G(x,y) - B(x,y)|^2 + |B(x,y) - R(x,y)|^2)\, dxdy$$

While I implemented only one method of CA correction, we will still use this function as it is representative of the picture quality CA-wise. The lower the score, the less affected by CA the image is.

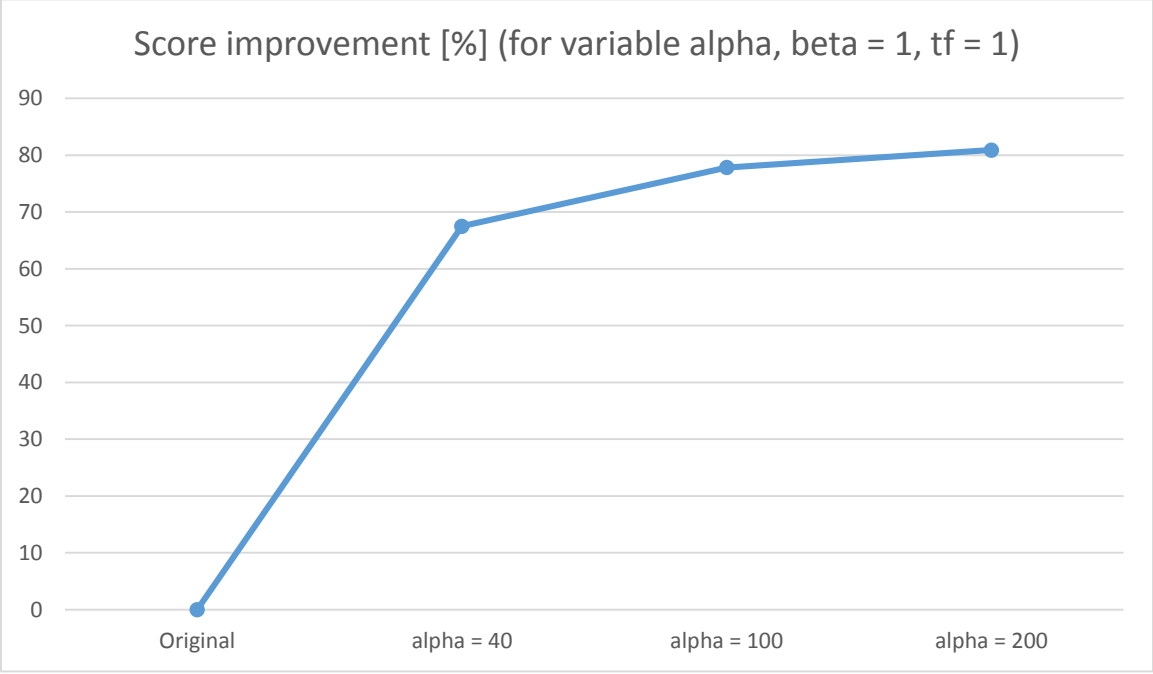
The score of the original image is used as a baseline to compare the algorithm efficiency between the two test images.

Score original test image 1	786
Score original test image 2	10139

Results against Transverse CA

TCA can be easily corrected via independent scaling of the color channels as it does not cause information loss.

Efficiency



Intuitively enough, the higher α, the more the difference in intensities is penalized, and the better correction we get. Unfortunately, it come with a high cost in processing time. The case $\alpha = 40$ is

processed in less than 30 seconds, while it takes 10 minutes when $\alpha = 200$. This is due to the CFL condition for the reaction term being dependent on α.

In all cases, the CA correction is highly visible.

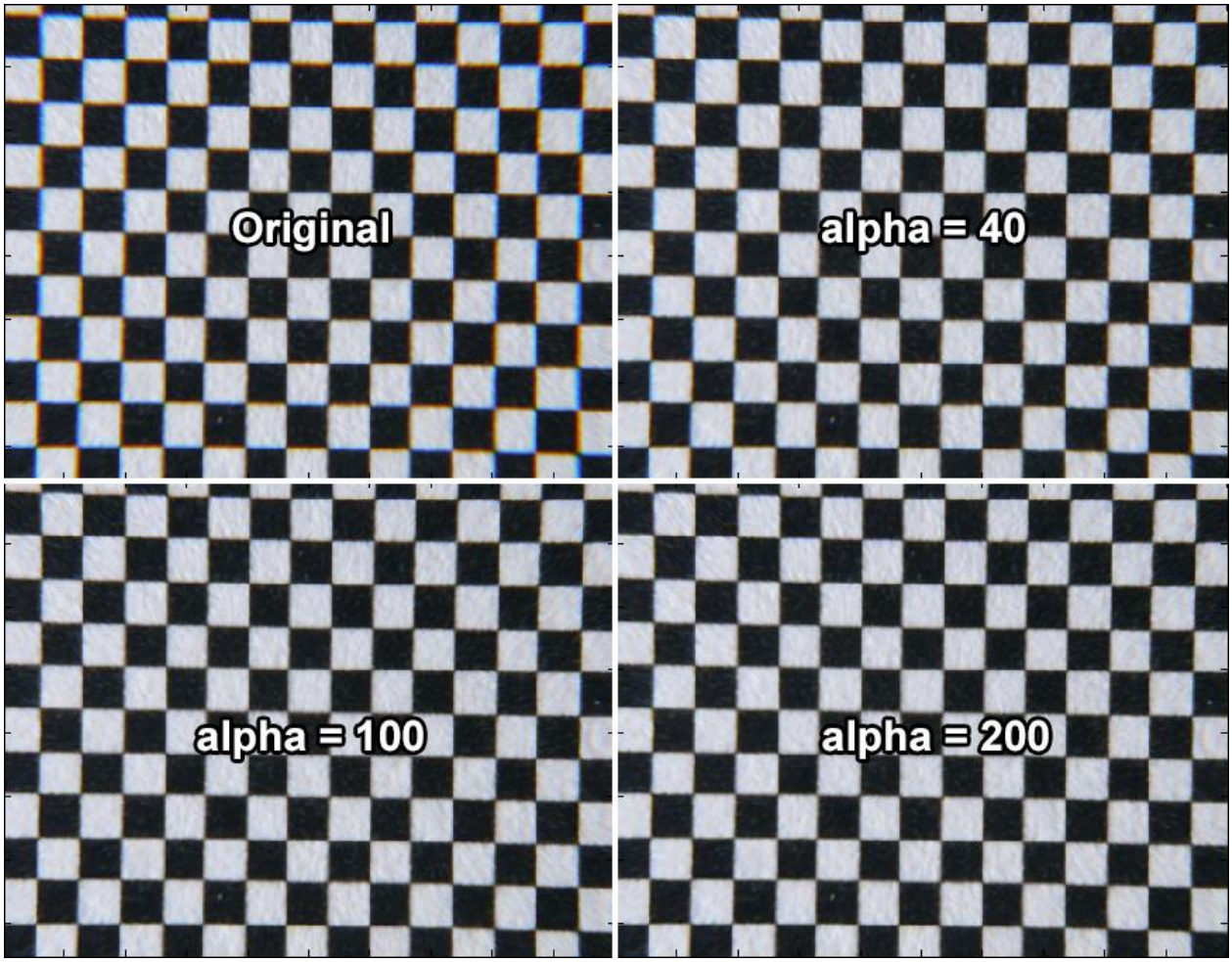


Figure 7: Comparison of the CA correction for alpha = 40, 100, 200.

For $\alpha = 100$ and higher, the CA is almost indiscernible.

No image distortion has been introduced by our algorithm at any value of α.

Energy descent

As a validation of the algorithm working as it should, I plotted the energy descent against the simulated time.

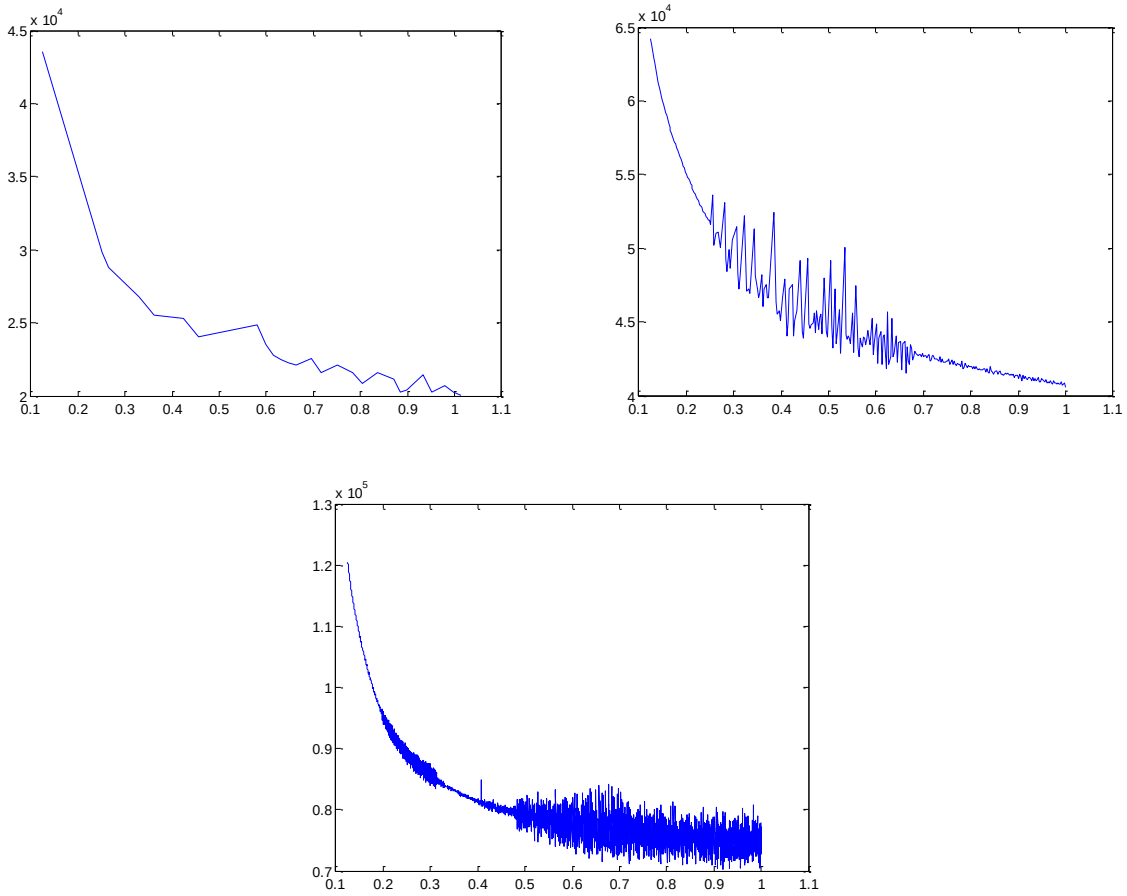
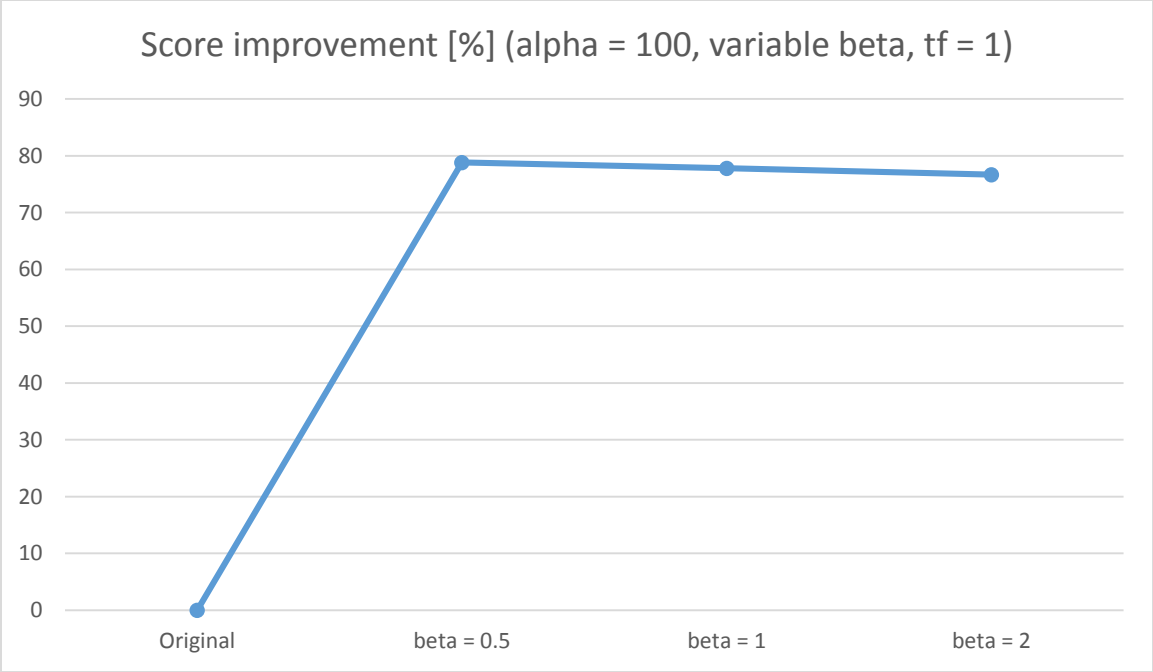


Figure 8: Energy for alpha = 40, 100, 200.

Diffusion coefficient effect

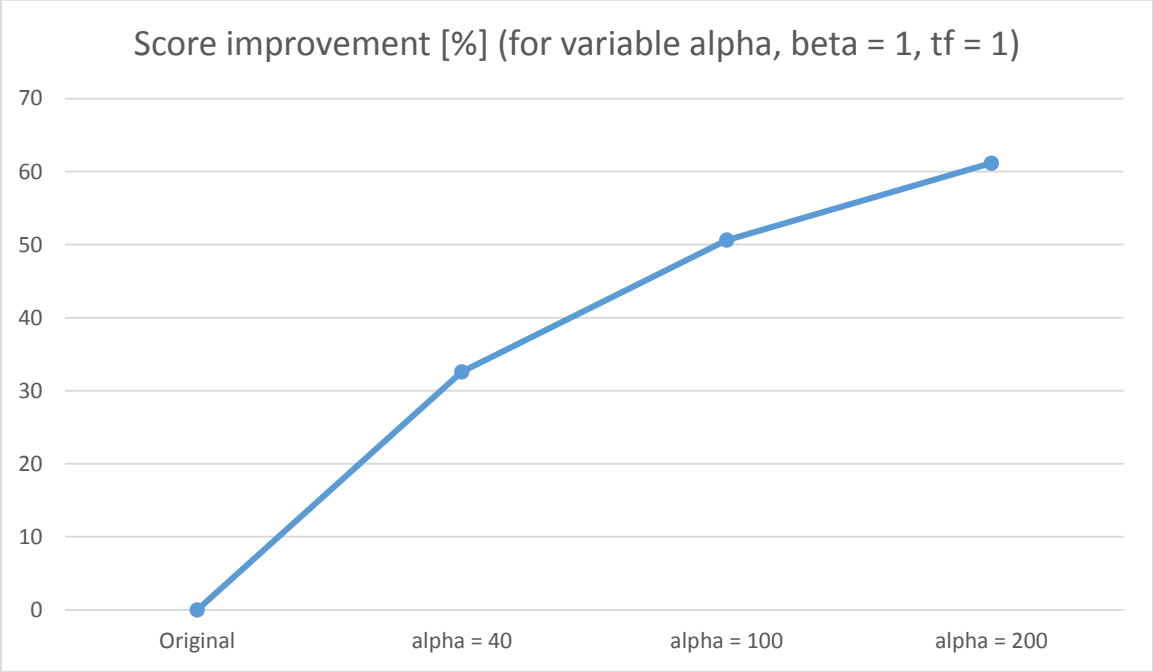
Varying β, the diffusion coefficient has a very limited effect. Increasing its value reduces the score improvement. CA is present only near high contrast edges, thus the offsets u and v are inherently discontinuous. The diffusion term fights this discontinuity and reduces the algorithm efficiency near the edges.



Results against Axial CA

ACA is usually hard to correct algorithmically because of the loss of information it causes.

Efficiency



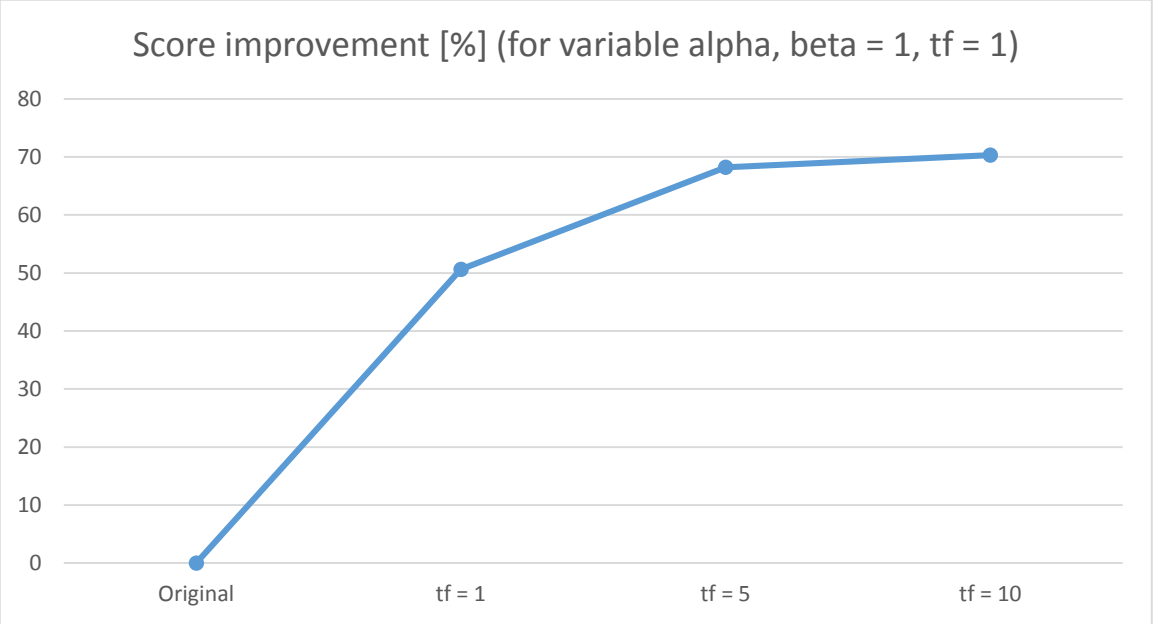
On this graph, we see that the improvement in image quality is not as high as for the TCA test image. This could be due to the impossibility to reconstruct the data lost in the blur of the R and B channels in the original image (red and blue are out-of-focus).

Even when $\alpha = 200$, the color contamination (especially blue) due to this blur is still visible.

Figure 9: Comparison of the CA correction for alpha = 40, 100, 200.

1s of virtual time did not seem to be enough to get a result as clean as for TCA correction. So I tried with $t_f = 1, 5, 10s$.

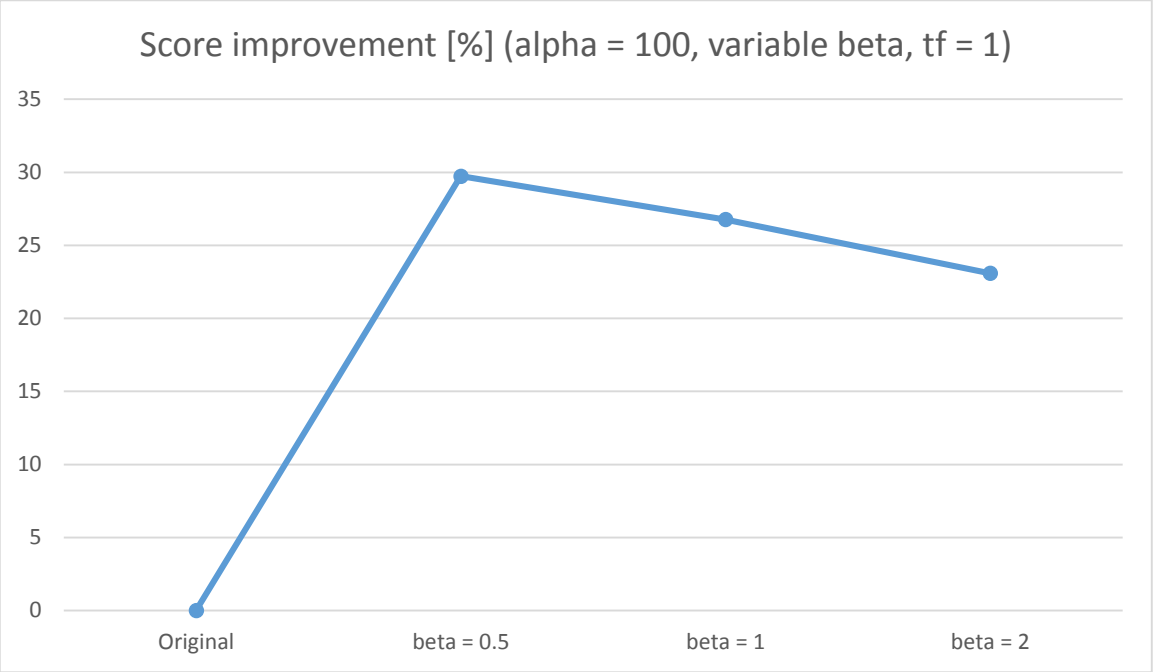
Increasing t_f



As I expected, the higher the final virtual time, the better the result. 70% improvement in image quality over the original image is around what we got for TCA, but required more than 20 minutes of computation on a smaller image. Most of the fringing disappeared but color contamination is still marginally present.

Figure 10: Final result for alpha = 1, beta = 1, tf = 10.

Diffusion coefficient effect



Similarly to TCA, and for the same reasons, increasing β has a negative effect on the image quality improvement. Here the change is more drastic as the improvement was already low.

Results on other images

I then tried this CA correction method against other images, closer to real-life pictures.

Face

An extreme case of Axial CA. $\alpha = 50,\ \beta = 0.5,\ t_f = 60s$.

Figure 11: Original image, corrected image after 60s.

Original score	14102
Score after 60s	5966 (−57.7%)

While the original image is severely affected by CA, this method allowed to clean the image by removing more than 50% of the CA after 60s.

Wooden beam

Purple fringing. $\alpha = 50,\ \beta = 0.5,\ t_f = 10s$

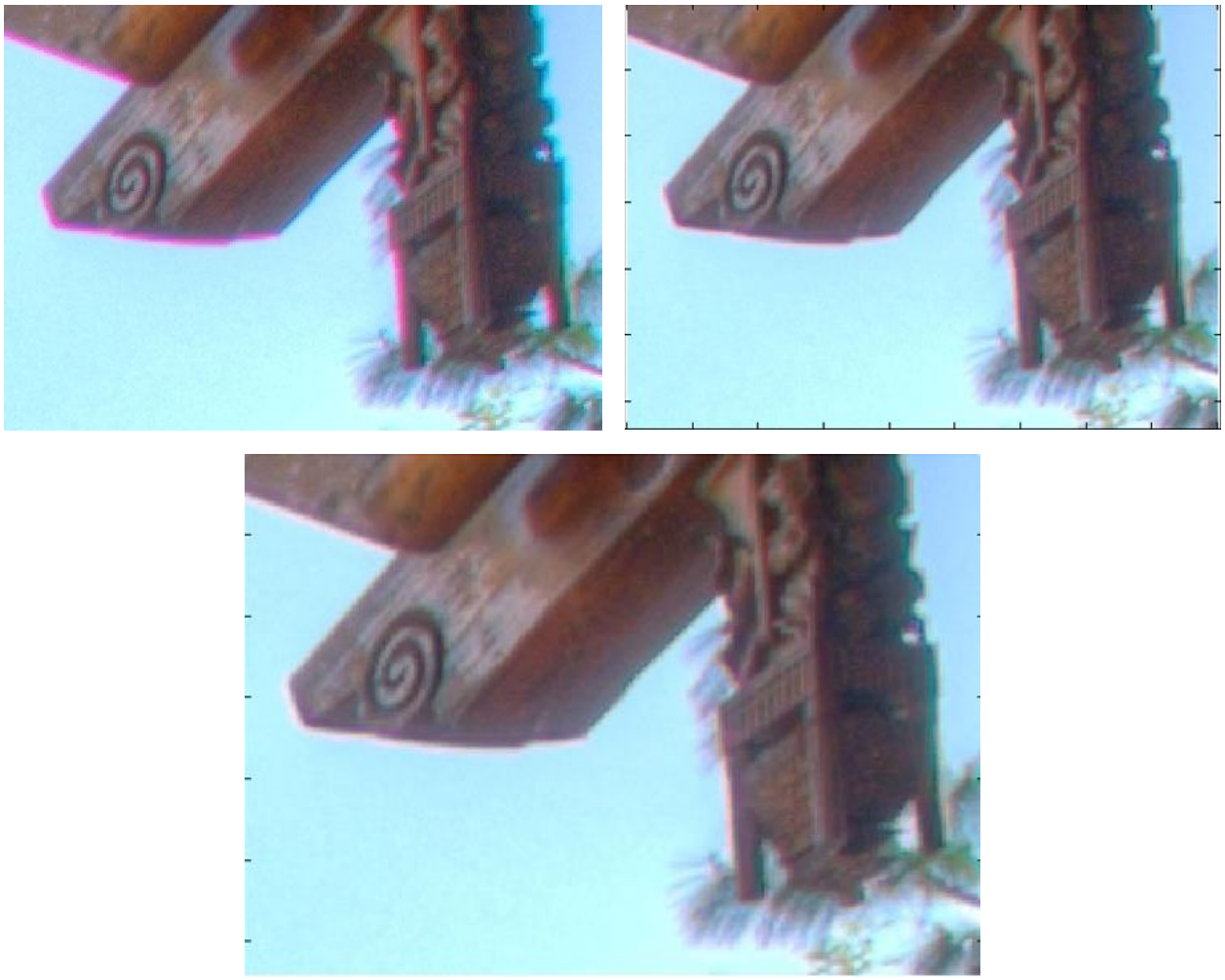


Figure 12: Original image, corrected image after 10s, corrected image after 20s.

Original score	13025
Score after 10s	10357(−20%)
Score after 20s	9863 (−24%)

This image is a typical case of what is called purple fringing. I can be caused by both TCA and ACA and is defined by purple borders on high contrast objects.

After 10s of simulated time, a 20% score improvement has been reached, but more than 50% of the CA is removed. This is normal because this image contains a lot of blue and brown, and the score function penalizes non gray colors. After 20s, purple fringing is almost completely removed.

While CA has been reduced visibly, no visible image deterioration has been caused by the algorithm.

Conclusions and ameliorations

As I demonstrated, the intensity-based energy gradient descent gives good results for grayscale images. Unfortunately, it comes with a very limiting downside: it is extremely slow for large images and large α.

This problem is inherent to the gradient descent scheme on an image. Optimization and vectorization of the code, or implementation in a faster, compiled language such as C++ would also help improve processing time.

The CFL condition of the PDE is certainly greatly conservative as I opted for a linearization and used the worst-case scenario. Finding a tighter CFL condition will increase the time steps and allow the same image to be treated in a shorter time. More importantly, the hypothesis that the data is smooth enough to approximate at first order doesn't hold true for well-focused pictures, and probably also for most real-life images.

The sensitivity to color intensities can be attenuated with the gradient-based energy. While this energy does not directly depends on the absolute color intensities, it is not totally immunized to them. One could design an energy that only takes into account the gradient direction (normalized gradient).

While it is slow and proved to work only with grayscale images, this method is able to reconstruct images affected by Axial CA, usually a hard task.