

Dehazing using the Dark Channel Prior and Guided Filter

Thomas Chabal & Martin Graive

École des Ponts ParisTech

February 2021

Abstract

Visibility in the atmosphere decreases with distance from the observer, and this can in some cases seriously affect the quality of photographs. Recovering a normal visibility in an image is an important challenge for image edition, but the challenge is difficult due to the absence of ground truth images corresponding to the haze-free ones. Following [3], we evaluated the local intensity of this “atmospheric perspective” and implemented a method to remove this haze and correct the image. We refined it thanks to Guided Image Filtering [4] and accelerated it with a Fast Guided Filter [5], making the method work for real-time applications. We fully reimplemented all the methods, and open-sourced this code¹.

1 Introduction

An important problem in image restoration is the removal of haze in images acquired in rainy or foggy weather. Indeed, the high amount of raindrops in the air reduces the transmission of light by absorbing photons and energy, which impacts the light beams that get to a camera sensor. This reduction of transmission depends on the depth of objects in an image, making those far away from the camera almost invisible in the numerical image as the distance the light beams cover through the haze is high. Yet, if the details did not completely disappear, they can be partially retrieved with exact methods. In this way, we can enhance a picture and make previously indistinguishable elements become clearly visible.

Haze removal has applications in fields where restoring images is important, such as photography for consumer needs, but this can also be useful in domains where exactness is critical, such as for judicial investigations, as the presented methods of these papers are exact, *i.e.* they do not hallucinate details as a deep neural network would.

2 Related Work

Haze removal is difficult because the haze depends on the (unknown) depth information. The problem is under-constrained if the input is only a single haze image. A lot of methods have been proposed to better constrain the problem by using multiple images or additional sources of information.

Polarization based methods [12, 13] use several images taken with different degrees of polarization to remove the haze. In [9, 8, 11], information is added by taking multiples pictures of

¹Code available at <https://github.com/thomaschabal/HazeRemoval>

the same scene but in different weather conditions. Naturally, depth-based methods like [6, 10] use a depth map provided by the user (or computed with the aid of user-made segmentation). Yet, these papers have heavy requirements which are often not achievable in practice and make these methods useless for most applications.

New methods [1, 14] make use of strong priors or assumptions. In [14], the author assumes that the haze-free image has higher contrast than the hazed one. Dehazing is then performed by maximizing the local contrast. Fattal, in [1], estimates the albedo of the scene and the medium transmission, assuming that the transmission and surface shading are uncorrelated. These assumptions may fail in practice, which drives us to explore other relevant methods.

3 Method

3.1 The model

This study mainly relies on two research papers on haze-removal: one using dark channel prior [3], which computes a transmission map to remove the haze, and the other about Guided Image filtering method [4] to refine this transmission map.

Haze in images is due to the presence of raindrops in the air, which absorb the light rays coming to the photo sensor. This absorption can be modeled using a simple physical understanding of the phenomenon. We introduce the notations I for the image acquired by the camera, A for the atmospheric light (*i.e.* the color of the haze), J for the radiance, that is the quantity of light that was emitted before its partial absorption by raindrops, and t the transmission factor of the light. We then define the following relation which lays the foundations of all our work, using x to refer to a pixel:

$$\boxed{\forall x, I(x) = J(x)t(x) + A(1 - t(x))}$$

We remark that t indeed models the transmission of light consistently with our human experience. If t is low ($t \approx 0$), which is equivalent to a major absorption of light and therefore a little amount of light that gets to the camera, then the image is mostly made of atmospheric light: $I(x) \approx A$. Conversely, if t is close to 1, which occurs when there is very little haze and all the light is transmitted through the air, the image is mainly the true radiance: $I(x) \approx J(x)$.

The goal of this study is to recover J only from I , estimating with the same method the atmospheric light A and the transmission map t , which can also be understood as a depth map. This does not demand to provide other information or images and uses only simple assumptions such as local constancy of transmission in patches or the dark channel prior presented in section 3.3.

3.2 Estimating the atmospheric light

In the method [3] used for the general pipeline of haze removal, He et al. exploit the dark channel prior. Under this assumption, most non-sky pixels in an image have at least one color channel whose intensity is very low (actually at least one pixel per considered patch). This assumption is valid in plenty of natural images but may fail for saturated images or images with dark objects or shadows where large parts are very dark and the method is not relevant anymore. It is expressed, with J the radiance, c the different color channels and $\Omega(x)$ a local patch around pixel x , as:

$$\min_{c \in \{r,g,b\}} \min_{y \in \Omega(x)} J^c(y) \approx 0$$

Using this property, we can therefore estimate the color of the haze A (or atmospheric light): by selecting the top 0.1% lightest dark channel pixels and then keeping only the one with highest intensity among them, we obtain a very good approximation of the haze's color. This is notably due to the empirical observation that bright dark channels concern mainly sky or haze pixels.

3.3 Estimating the transmission

The transmission map recovery also takes advantage of this dark prior channel assumption. We consider a patch $\Omega(x)$ of the image centered at pixel x and first deal with only one color channel. Here, we make another important assumption: we suppose that the transmission is constant equal to $t(x)$ in this patch. Using the previous model, we have that:

$$\forall y \in \Omega(x), I^c(y) = J^c(y)t(x) + A^c(1 - t(x))$$

Normalizing this with $A^c > 0$ (which is always the case in practice) and looking at the minimal intensity over the three color channels and over the patch, we obtain:

$$\min_c \min_{y \in \Omega(x)} \left(\frac{I^c(y)}{A^c} \right) = (1 - t(x)) + t(x) \min_c \min_{y \in \Omega(x)} \left(\frac{J^c(y)}{A^c} \right)$$

We then invoke the dark channel prior which assesses that the minimal intensity of the radiance over these color channels is zero, which corresponds to a haze-free image. Therefore, the renormalization of this value by the atmospheric light remains zero, which conducts to:

$$\min_c \min_{y \in \Omega(x)} \left(\frac{I^c(y)}{A^c} \right) = 1 - t(x)$$

Hence the following expression of the transmission t :

$$t(x) = 1 - \min_c \min_{y \in \Omega(x)} \left(\frac{I^c(y)}{A^c} \right)$$

This function is a good estimate of the actual depth map (see figure 1), and that explains why keeping some haze in the dehazed image is good to convey a feeling of depth and such that the resulting image does not look too unnatural. In this way, the authors introduce some constant ω close to 1 to attenuate the haze removal, which eventually gives the transmission:

$$\boxed{\forall x, t(x) = 1 - \omega \min_c \min_{y \in \Omega(x)} \left(\frac{I^c(y)}{A^c} \right).}$$

3.4 Computing the radiance

Once we obtained A and t , reordering the terms of the model equation gives:

$$\forall x, t(x)(J(x) - A) = I(x) - A.$$

Yet, this transmission t can be very small ($t \approx 0$) in some areas, making it impossible to recover the radiance J (the value would explode otherwise). Thus, we define a minimum value t_0 for the transmission before retrieving the radiance as:

$$\boxed{\forall x, J(x) = \frac{I(x) - A}{\max(t(x), t_0)} + A}$$

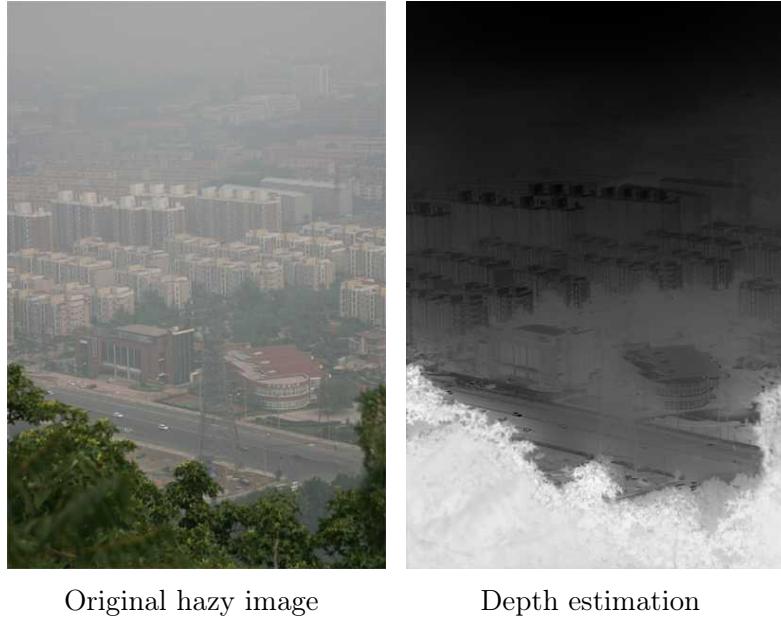


Figure 1: Transmission as an estimate of the depth map

The last step is usually to boost the exposure of the resulting image, which is usually rather dim. We used gamma correction (with a per-case user-defined value) to make the image lighter.

Yet, the resulting radiance is often of very bad quality, with bright artifacts around all edges of the image. This is visible on figure 2 and must be attributed to gross transmission map.

3.5 Refining the transmission

The transmission map as computed in section 3.3 displays block effects because of the minimum over a patch Ω . We intend to refine the transmission because it is of the highest importance in the algorithm.

3.5.1 Soft matting

The method proposed in the original article [3] to improve the transmission map is soft matting. Indeed, the relation that defines I as a t -convex combination of J and A is a matting problem. Therefore, following [7], the new transmission minimizes the cost function

$$E : \mathbf{t} \in [0, 1]^n \mapsto \underbrace{\mathbf{t}^\top L \mathbf{t}}_{\text{alpha}} + \underbrace{\lambda \|\mathbf{t} - \tilde{\mathbf{t}}\|_2^2}_{\text{data}}$$

where $\tilde{\mathbf{t}}$ is the vector form of the previously computed transmission and L is called the Laplacian matrix. The Laplacian matting matrix is a sparse $n \times n$ (with n the number of pixels in the image) symmetric positive definite matrix. It is an adjustment of the graph Laplacian matrix $L = D - W$ used for spectral segmentation.

Therefore the first term of the cost function is the alpha-matting term, that smooths the transmission around the edges of objects. The second term keeps the resulting transmission close to the patch-grained estimation. A hyper-parameter to tune is the strength of regularization λ , see section 4.

The global minimizer of the coercive and convex cost function E is solution of the equation

$$\nabla E(\mathbf{t}) = 0 \iff (L + \lambda I_n) \mathbf{t} = \lambda \tilde{\mathbf{t}}.$$

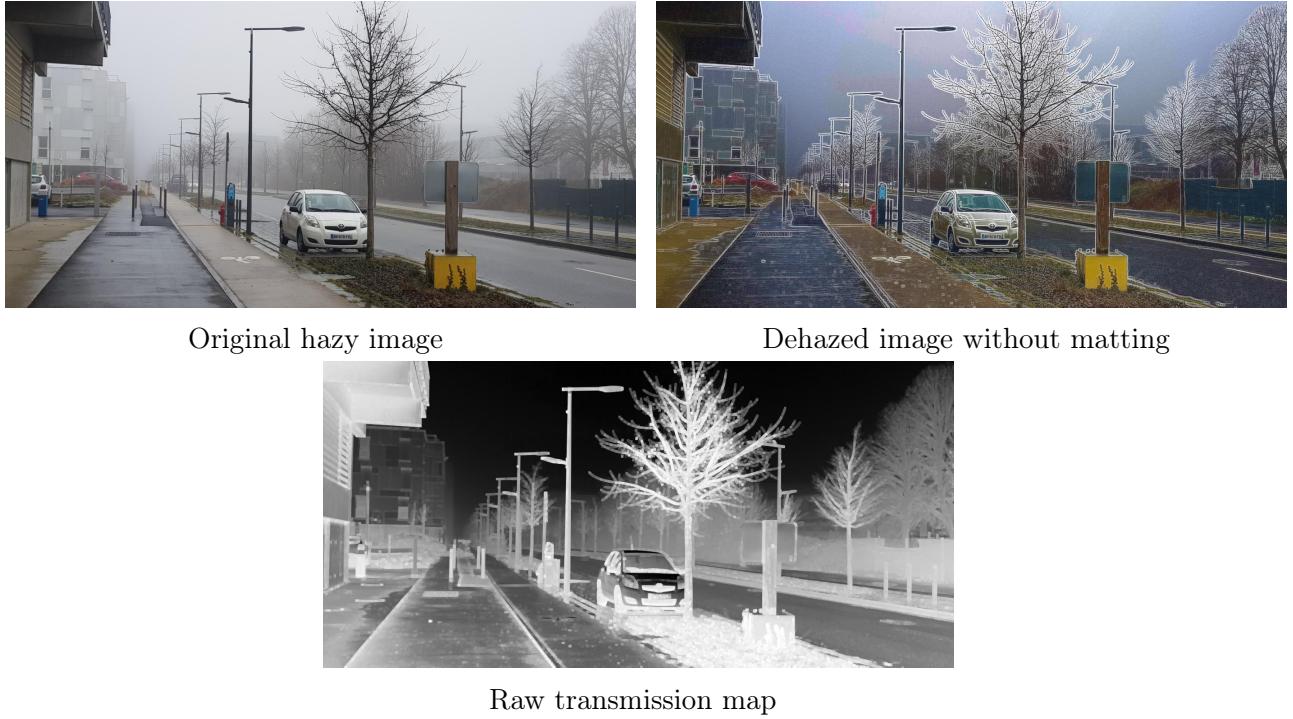


Figure 2: Radiance recovered after dehazing the image without any transmission matting. The edges are marked with large white lines, while the transmission map is of very low quality.

In our implementation we decided to solve this sparse linear system using SciPy’s sparse linear algebra submodule [15]. Specifically we used conjugate gradient iterations, as in [3]. The preconditioner for $L + \lambda I_n$ was $\text{diag}(L + \lambda I_n)^{-1} \approx (L + \lambda I_n)^{-1}$.

3.5.2 Guided image filtering

Sadly, soft matting is very slow and takes several minutes to compute on large images. This drove us towards the use of another method: Guided Image Filtering [4]. Through this research, He et al. linked a guide image I and a high-quality output q with a linear relation. Yet, a global linear equation would not be a solution as the refinement may be different in various areas. We thus look for local linear equations on windows ω_k centered at pixel k for all k in the image with:

$$\forall i \in \omega_k, q_i = a_k I_i + b_k$$

In that way, we define an energy function of these a_k and b_k , which we minimize with respect to these two parameters. The goal is to get an output q close to the input image p , so to reduce the L^2 distance $(a_k I_i + b_k - p_i)^2$ for $i \in \omega_k$, while controlling a_k . The idea behind this control is to limit the gradient of the output $\nabla q = a \nabla I$, and this is done with a penalization term ϵa_k^2 . The overall energy is then:

$$\forall k, E(a_k, b_k) = \sum_{i \in \omega_k} [(a_k I_i + b_k - p_i)^2 + \epsilon a_k^2]$$

The optimal parameters a_k and b_k are simply found out by differentiating this quadratic function and setting it equal to 0, the solution being unique as the energy is convex. The first equation of guided filtering gives the expression of q_i on a specific window (for a given $i \in \omega_k$), so in order to finally recover the true value of q_i we must average these linear equations over all the windows that include the considered pixel. This eventually gives the following formula

of guided filtering, with \bar{a}_i and \bar{b}_i being the mean values of a_k and b_k over the windows ω_k that include pixel i , μ_k and Σ_k the mean and covariance matrix of the RGB guide image I in window ω_k , I_3 the identity matrix of dimension 3, \bar{p}_k the mean value of input image p in ω_k and $|\omega_k|$ the number of pixels in window ω_k :

$$\forall i, q_i = \sum_{k: i \in \omega_k} \frac{a_k^T I_i + b_k}{|\omega_k|} = \bar{a}_i^T I_i + \bar{b}_i \quad \text{with} \quad \begin{cases} a_k &= (\Sigma_k + \epsilon I_3)^{-1} \frac{1}{|\omega_k|} \sum_{i \in \omega_k} [I_i p_i - \mu_k \bar{p}_k] \\ b_k &= \bar{p}_k - a_k^T \mu_k \end{cases}$$

This formula is valid for general guided filtering. For haze removal specifically, the guide image I is the initial RGB image full of haze and the input p is the raw transmission map to be refined, which is 1-channel.

3.5.3 Fast guided filter

Going further in the project, we had a look at the Fast Guided Filter proposed by [5]. This paper proposes to accelerate the refinement step by reusing the guided filtering presented above, but this time using downsampled inputs, *i.e.* guide image and raw transmission map, and eventually upscale the resulting transmission. This downsampling is performed with a scale factor s defined by the user. It appears that using a downscale factor of 4 can make the computation around 10 times faster than simple guided filtering. The new pipeline is the following one, denoting by I the guide image, p the raw transmission map, \bar{a} and \bar{b} the guided filtering coefficients and q the refined map:

$$(I, p) \xrightarrow[\text{factor } s]{\text{Downscale}} (I', p') \xrightarrow[\text{Compute linear parameters}]{\text{Guided filtering}} (\bar{a}', \bar{b}') \xrightarrow[\text{factor } s]{\text{Upscale}} (\bar{a}, \bar{b}) \xrightarrow[\substack{\text{End of guided filtering} \\ q = \bar{a}^T I + \bar{b}}]{\text{}} q$$

The speedup is explainable by the complexity of the guided filtering. Indeed, the method is linear with the number of pixels in the guide image and the transmission map ($O(N)$ with N pixels). Thus, for a square image of size $N = n \times n$, the guided filter has complexity $O(n^2)$. If we downscale it by a scale factor s in each direction, we then have a complexity of $O(\frac{n^2}{s^2}) = O(\frac{N}{s^2})$, which is a significant reduction for $s = 4$. In addition, we observe that the upscaled map is close to the map obtained with simple guided filter and the results are very similar, such that the gain in computation time is not balanced by a loss in quality.

4 Experiments

The models previously described require few parameters. After reimplementing all the methods from scratch, we studied the impact of all the parameters on the result in order to pick the best ones.

4.1 Haze removal method

First, the haze removal method itself introduces a few parameters to adjust manually.

Haze-conservation parameter ω The effect of parameter ω is shown in figure 3. With $\omega = 1$, the perspective seems unnatural and the image is very noisy. A too little value for ω , however, is fruitless. Thus, setting this parameter a little below 1 allows a little bit of haze to be kept in the final image to make distant objects feel distant. In the article [3], the authors used the value $\omega = 0.95$, which we also chose after leading our study on its impact.

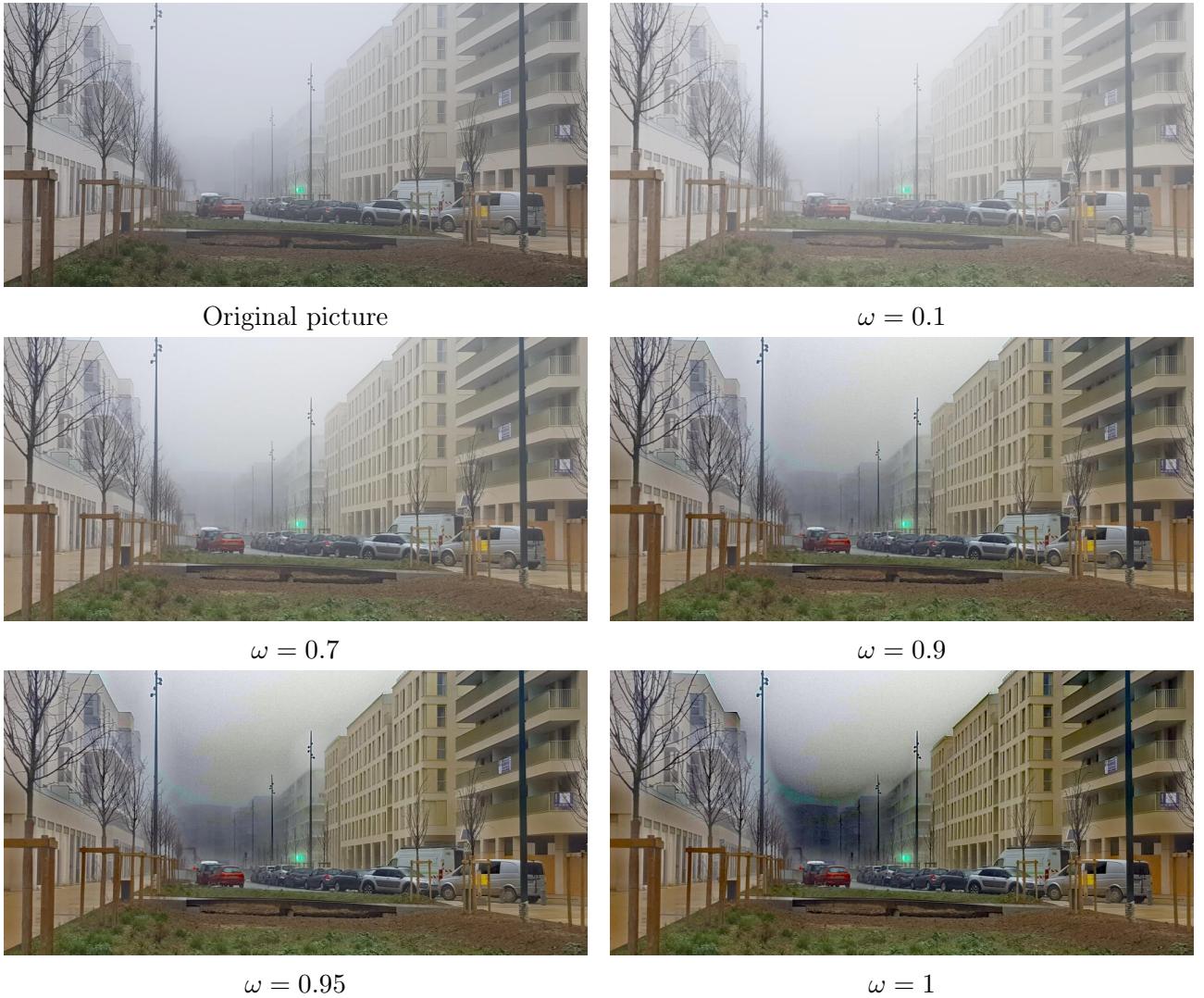


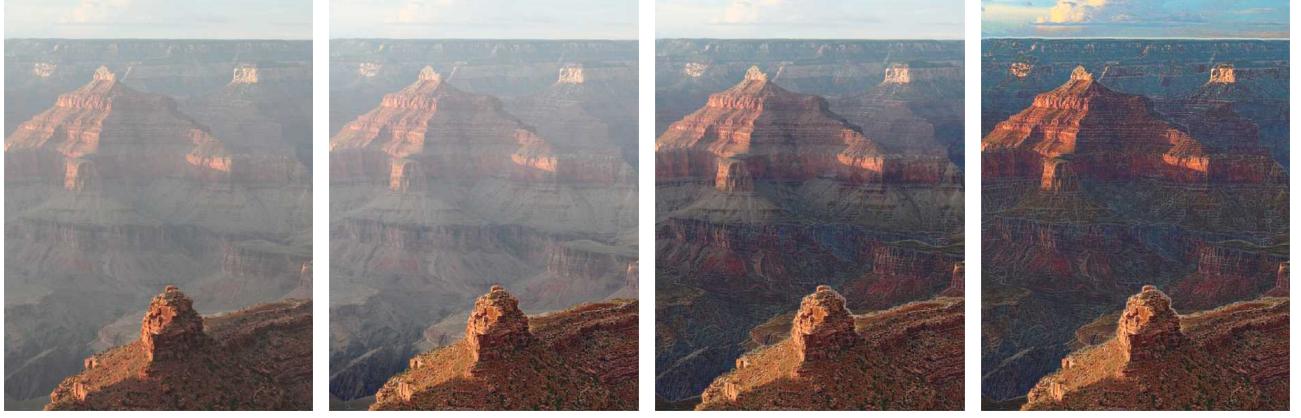
Figure 3: Impact of the parameter ω limiting the haze removal

Threshold t_0 As seen on figure 4, changing the value of the threshold actually changes the depth at which the haze stops being removed. The smaller t_0 , the farther away the farthest haze remnants. The article used $t_0 = 0.1$ and so did we after visualizing the results of figure 4.

Patch size The patch size used for the computation of the raw transmission map has severe impact on the final result. Remember that we consider the transmission as constant on such patches, so they can't be too large for images with important haze. Yet, having too small patch sizes gives very low-quality results as visible on figure 5 for a patch size of 1 or 2. With intermediate patch sizes, artifacts become visible around edges, even more pronounced around thin structures like tree branches. Given these observations, we decided to take a patch size of 5 for the following studies.

4.2 Soft-matting

Soft-matting requires a regularization parameter to constrain the matted transmission to coarser one evaluated using the dark channel prior. As seen on figure 6, the smaller the regularization parameter, the smoother the resulting image gets. On the other hand, the larger the parameter λ is, the more the output resembles the “vanilla” radiance.



Original picture

$t_0 = 0.7$

$t_0 = 0.5$

$t_0 = 0.1$

Figure 4: Impact of t_0 , the minimum value for transmission, on the resulting image.



Original picture

Patch size 1

Patch size 2

Patch size 5

Patch size 10

Patch size 25

Figure 5: Impact of the patch size for first transmission estimation

As for the Laplacian matrix, it depends on:

- a regularization parameter ε which controls the smoothness of the matte (and may improve convergence). We set it to 10^{-7} , in keeping with the defaults of [2];



$$\lambda = 0.1$$



$$\lambda = 10^{-3}$$

Figure 6: Impact of the regularization parameter of soft matting

- a radius r , different from the size of the patches Ω , and usually set to 1 (*i.e.* only adjacent pixels) to reduce spatial overhead, that controls the propagation of information in the image. In keeping with the graph Laplacian analogy, the radius refers to the denseness, of the graph associated with the image: the broader the radius, the more degree to each node.

4.3 Guided Image Filtering

Guided image filtering is characterized by two parameters as seen in section 3.5.2: the penalty ϵ and the window size.

Penalty term ϵ We observed on figure 7 that a too small penalty yields blurry and noisy results and no information have been recovered from the hazy image. Conversely, a too important regularization will give a very high contrast which even seems unreal, all the dark areas becoming black. We thus decided to pick $\epsilon = 5.10^{-3}$ which is a good trade-off.

Window size The ideal window size depends considerably on the input size: the larger the input guide image and transmission map, the larger the window size. We remarked that a window size of 30 was appropriate for images of maximum size 600 while not being too long to compute (the computation time depends on the window size). We then defined a simple rule to automate the choice of window size: $\text{window_size} = 15 \left\lceil \frac{\sqrt{\text{max_size}}}{10} \right\rceil$ (with this rule, an image with size 600 will have a window of size 30).

5 Evaluation

We performed haze removal using all the techniques described previously on a host of personal images that we shot for this project. A sample of these results is visible on figures 8 and 9. These pictures originally were very hazy, that even makes it hard to see the factory in the middle of image 8. When computing haze removal without any form of matting, we obtain images where information has been recovered but large artifacts appear as bright edges for every structure, as if an edge filter had been applied. When applying soft matting as recommended by [3], we obtain very contrasted images where some haze is still visible and the factory remains partially invisible. When eventually computing guided filter and fast guided filter, the result is more pleasant on the street image 9, while making every element clearly visible on image 8.

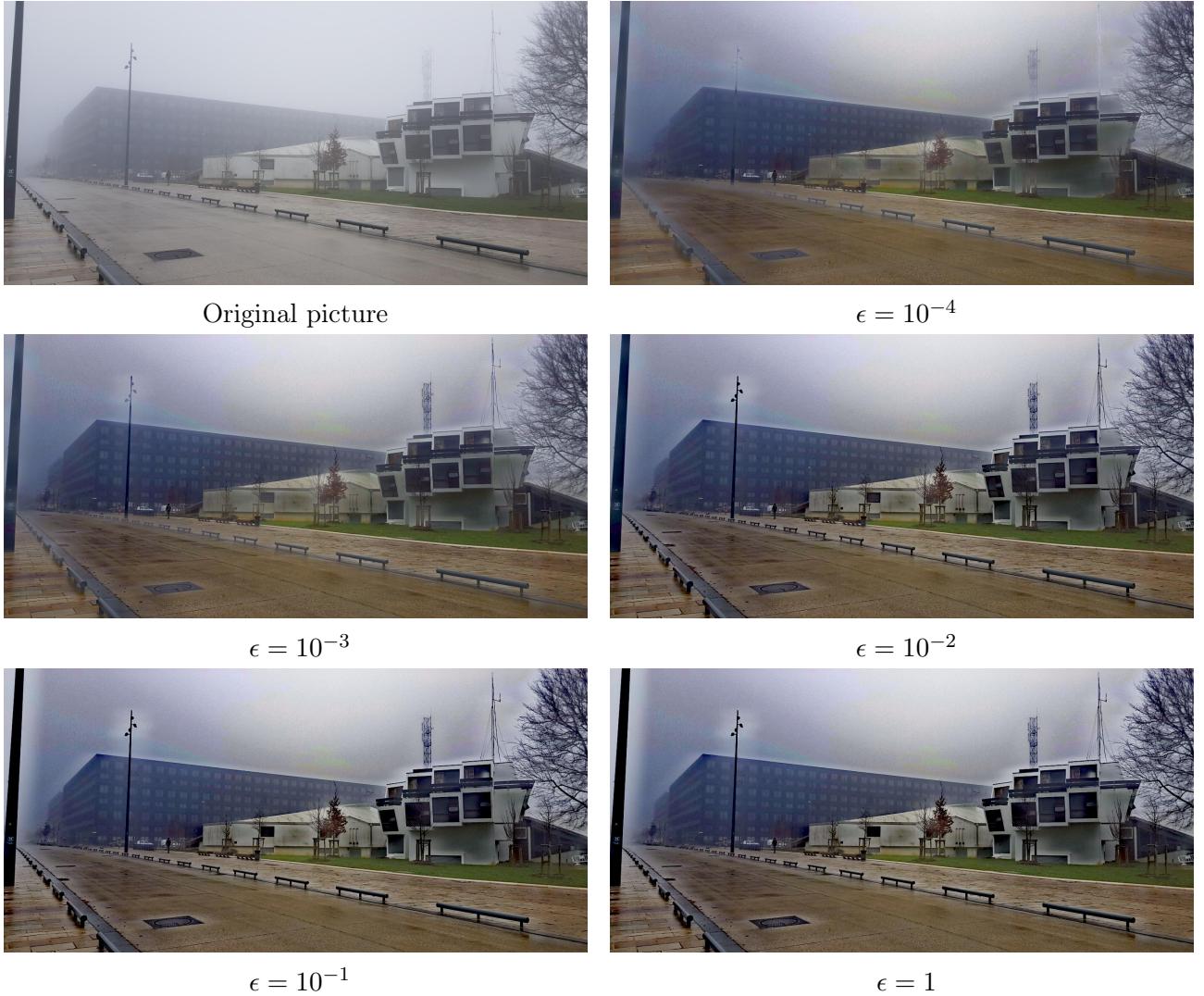


Figure 7: Impact of ϵ , penalty term in the guided filter, on the resulting image

Yet, while (fast) guided filter efficiently removes haze, it also emphasizes noise in the sky parts. This is particularly visible in images where the sky occupies a large part, like in figures 3 and 10, which afterwards contain colored artifacts or even have huge color differences between the sky and other patches (especially on the first example of figure 10). Moreover, the impact of patches around edges is major, generating bright areas around important edges such as on figure 9. This can be explained by the important gradient on these areas that are not handled well by the guided filter nor by the dark channel prior which is not respected in these patches.

Looking at the computational cost of these methods, we notice clear differences. The original article [3] used soft matting to refine the transmission map. Yet, this algorithm is by far the hungriest one of all those implemented here. The guided filtering tactic from [4] is an improvement in terms of speed, but the method is even more interesting when being optimized with the fast guided filter [5]. The latter yields similar results in terms of quality while being around 10 times faster. All these computation times are summarized in table 1.

No quantitative measure of the performances can be done as there is no reference image for these methods. This study only relies on qualitative evaluation.

Overall, the best trade-off between great visual result and low computation time is the fast guided filter: with it, we reach excellent results while having a very low computation time, which nearly makes the method adapted for real-time applications.

Matting method	None	Soft-matting	Guided Image Filtering	Fast Guided Filter
Computation time (s) Image size: 291x600	0.07	5.5	3.1	0.5
Computation time (s) Image size: 680x1400	0.4	28.3	23.3	0.9
Computation time (s) Image size: 972x2000	0.8	54.4	76.0	1.5

Table 1: Computation times for all the matting methods studied in this paper on images of various sizes.

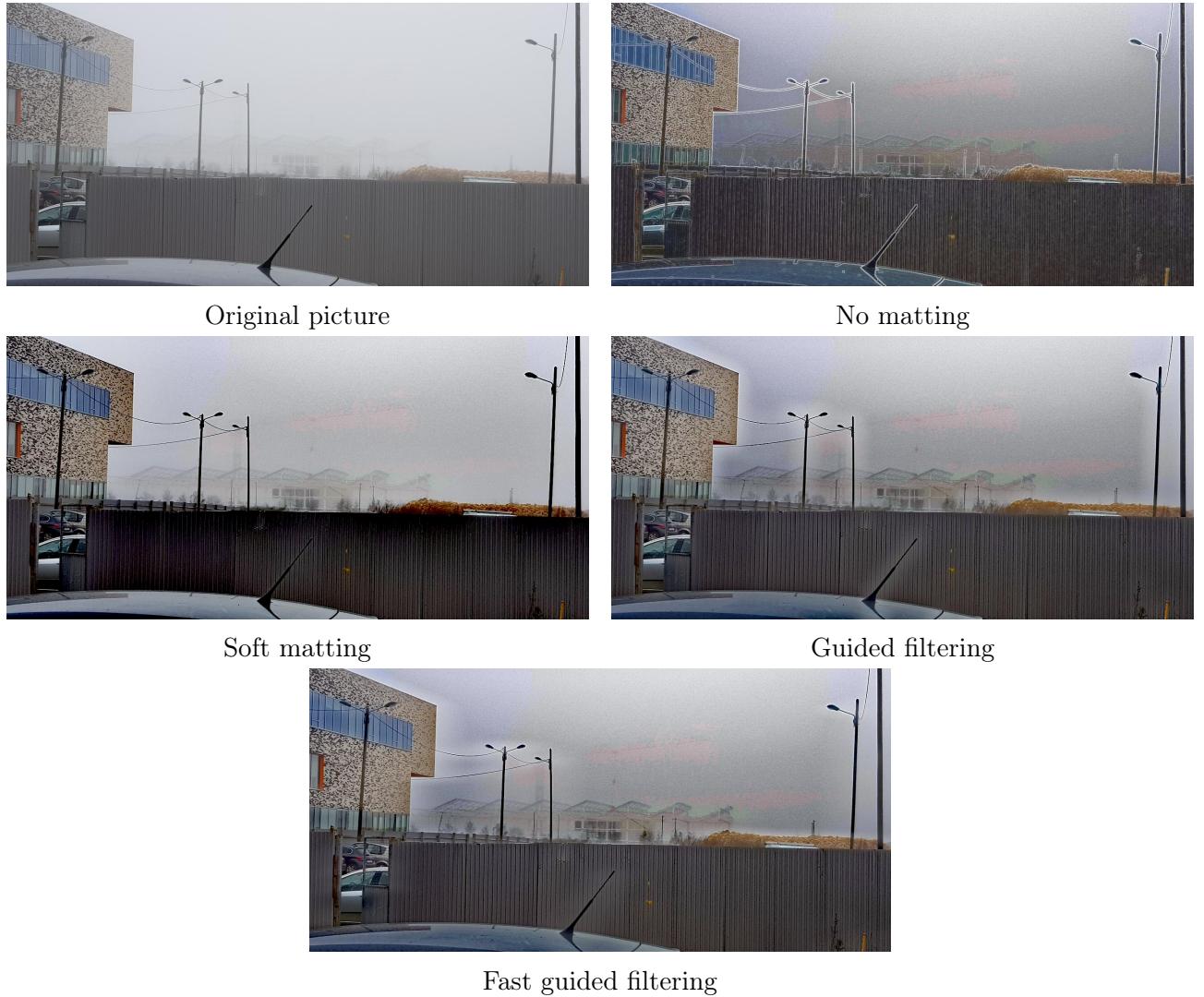


Figure 8: Haze removal with various matting methods

6 Conclusion

As part of this project, we explored the replacement of soft-matting by guided image filtering and fast guided filter in the refinement of transmission map for haze-removal. All the algorithms were reimplemented from scratch and evaluations were lead on images of our own, taken under hazy conditions.

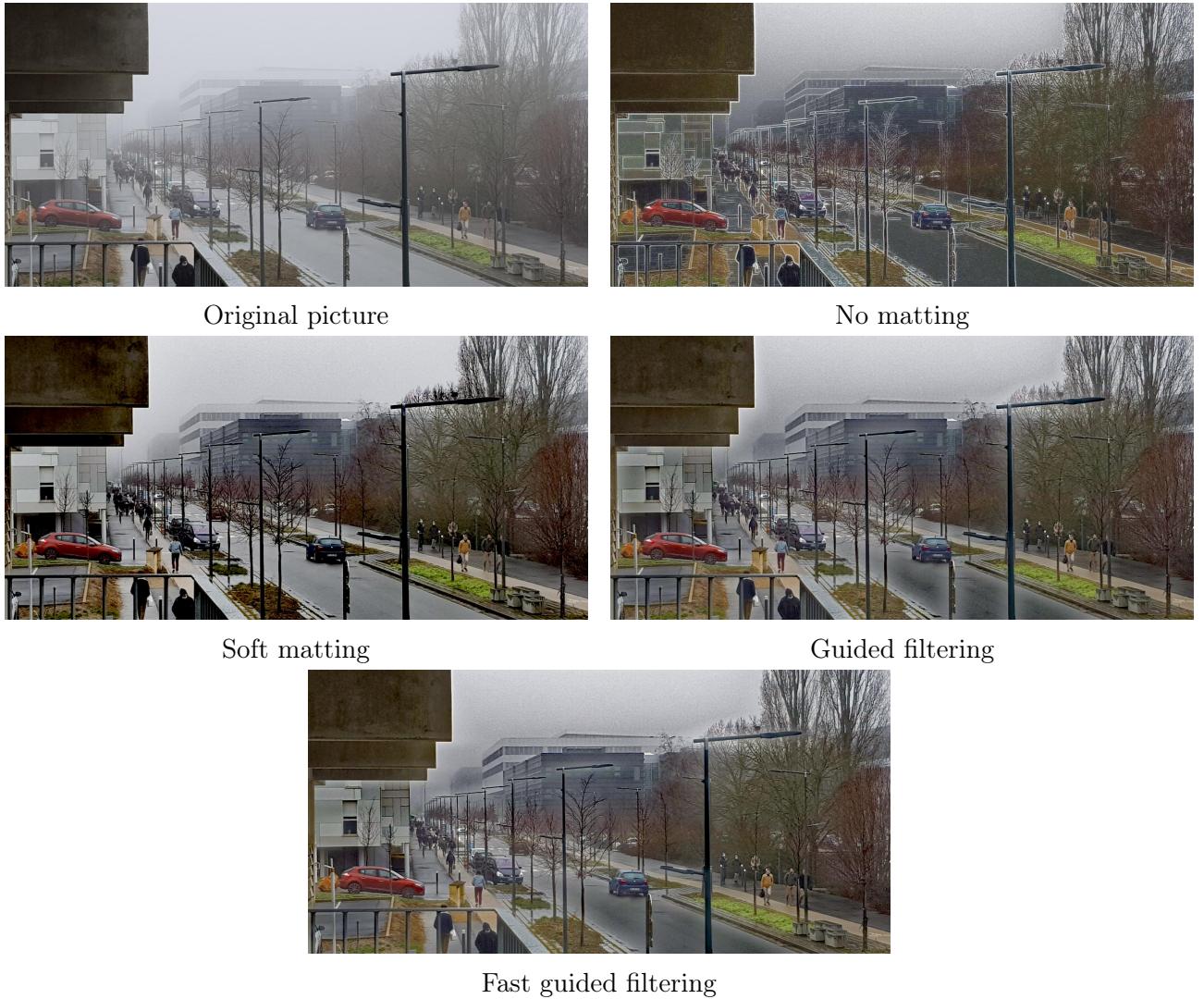


Figure 9: Haze removal with various matting methods



Figure 10: Failure cases of haze removal in images where there is a large part devoted to the sky

We obtained excellent results for haze removal on large images using the fast guided filter and were able to perform dehazing in no time.

If this project focused on image dehazing, the same technique can be used for other classical computer vision problems such as detail enhancement, HDR compression, flash-no flash image denoising or matting feathering.

References

- [1] R. Fattal. Single image dehazing. *ACM SIGGRAPH 2008 papers*, 2008.
- [2] Thomas Germer, Tobias Uelwer, Stefan Conrad, and Stefan Harmeling. Pymatting: A python library for alpha matting. *Journal of Open Source Software*, 5(54):2481, 2020.
- [3] K. He, J. Sun, and X. Tang. Single image haze removal using dark channel prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(12):2341–2353, 2011.
- [4] K. He, J. Sun, and X. Tang. Guided image filtering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(6):1397–1409, 2013.
- [5] Kaiming He and Jian Sun. Fast guided filter. *ArXiv*, abs/1505.00996, 2015.
- [6] Johannes Kopf, Boris Neubert, Billy Chen, Michael Cohen, Daniel Cohen-Or, Oliver Deussen, Matt Uyttendaele, and Dani Lischinski. Deep photo: Model-based photograph enhancement and viewing. *ACM Transactions on Graphics (TOG)*, 27:116, 12 2008.
- [7] A. Levin, D. Lischinski, and Y. Weiss. A closed form solution to natural image matting. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’06)*, volume 1, pages 61–68, 2006.
- [8] S. G. Narasimhan and S. K. Nayar. Contrast restoration of weather degraded images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(6):713–724, 2003.
- [9] Srinivasa Narasimhan and Shree Nayar. Chromatic framework for vision in bad weather. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, volume 1, pages 598 – 605 vol.1, 02 2000.
- [10] Srinivasa Narasimhan and Shree Nayar. Interactive (de)weathering of an image using physical models. *IEEE Workshop on Color and Photometric Methods in Computer Vision*, 10, 12 2015.
- [11] S. Nayar and S. Narasimhan. Vision in bad weather. *Proceedings of the Seventh IEEE International Conference on Computer Vision*, 2:820–827 vol.2, 1999.
- [12] Y. Y. Schechner, S. G. Narasimhan, and S. K. Nayar. Instant dehazing of images using polarization. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, volume 1, pages I–I, 2001.
- [13] S. Shwartz, E. Namer, and Y. Y. Schechner. Blind haze separation. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’06)*, volume 2, pages 1984–1991, 2006.
- [14] R. Tan. Visibility in bad weather from a single image. *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

- [15] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.