

# GUI-Aided NIR and Color Image Blending

Andrea Guidi, Radhakrishna Achanta, Clement Fredembach, Sabine Süsstrunk

*School of Computer and Communication Sciences*

*Ecole Polytechnique Federale de Lausanne (EPFL), Switzerland*

[andrea.guidi, radhakrishna.achanta, clement.fredembach, sabine.sustrunk]@epfl.ch

**Abstract**—A near-infrared (NIR) image of a scene contains different information than the corresponding visible spectrum (RGB) image, due to physical phenomena related to scattering, absorption, and reflectance of the different radiation bands. It can be desirable to combine the most relevant image content from both NIR and RGB to improve image quality. In this paper, we present two schemes of combining NIR and RGB information to obtain visually better quality images. The first is an automatic approach that performs segment-based NIR-blending by local and global entropy maximization. The second uses a GUI-aided approach where the user chooses the segments and manually controls blending. To minimize artifacts at the segment boundaries, three different boundary smoothing methods are introduced and compared in terms of speed and quality of output.

## I. INTRODUCTION

Near-infrared (NIR) is used in many imaging tasks. As opposed to X-ray or Thermal Infrared images, the scene content of an NIR image is similar to what is perceived in the visible spectrum. Much research has been devoted to security applications. Images or videos can be captured in the dark with acceptable quality [1]. Similarly, by fitting an NIR source on a camera, e.g., by using flash light fitted with a filter, one can obtain photographs that are almost invariant to ambient light sources, an approach which has been used in face recognition [2]. Another field that uses NIR information is remote sensing, where it is often employed in conjunction with middle- or far-infrared to better distinguish object classes, such as water, constructed areas, and in particular vegetation [3]. NIR imaging has also been employed for in-depth analysis of tree types and health [4].

Near-infrared spectral responses can also be used to discriminate between samples that are visually indistinguishable. Among these are analyses of counterfeited products, such as cigars or drugs [5], disambiguating black metamers [6], as well as discriminating painted surfaces [7]. In all these studies, the visible spectra of considered objects or surfaces are almost identical. NIR images nonetheless capture radiation beyond the visible spectrum, and as such, their color interpretation is difficult to ascertain. This is one of the primary reasons why NIR applications are generally very task specific.

The goal of this paper is to explore the idea of combining the “best” of both NIR and RGB image content to obtain better quality images. NIR information based color image quality improvement has been previously done in [8] where haze is



(a) RGB image



(b) NIR image



(c) GUI output

Fig. 1. 1(a) Input RGB image. 1(b) Input near-infrared image. 1(c) Output obtained by using our GUI-aided method. Note that by combining RGB and NIR information, we can obtain an image richer in detail.

removed from visual (VS) images.

We first show results of combining VS and NIR information using an automatic method that relies on maximizing entropy in the output image. Motivated by the fact that “better visual quality” is highly user- and task-dependent, we also present

a GUI-aided approach of improving image quality using NIR and visible sensor responses. The GUI allows users to select pre-segmented regions in images where they can control the ratio of NIR to visible information content.

At the end of a segment based blending process (automatic or GUI-aided), artifacts may appear at the edges of segments. We demonstrate three different approaches of automatically removing these artifacts: Gaussian blurring, matting, or Bilateral filtering. These three approaches are compared for speed and visual quality.

The rest of the paper is organized as follows: Section II briefly explains how NIR and visible images of a scene are obtained, Section III explains the segment-wise automatic blending technique, Section IV explains the same for GUI-aided (manual) blending, Section V describes three methods of boundary artifact removal, and finally Section VI concludes the paper.

## II. NIR IMAGE ACQUISITION AND BLENDING

Camera sensors are made of silicon, sensitive to radiation from 200 nm to 1100 nm. The 400-700 nm range is called the visible band, as it corresponds to the radiation spectrum the human visual system is sensitive to, while the 700-1100 range belongs to the NIR band. The ultraviolet part of the spectrum (200-400 nm) is generally filtered out by the camera's optical elements. A filter, called hot mirror, protects the sensors from incoming NIR radiation. By replacing the hot mirror by a piece of plain glass, the camera is enabled to capture both visible and NIR [9]. NIR and visible images of the same scene are obtained by manually swapping an NIR and visible blocking filter on the lens between each shot.

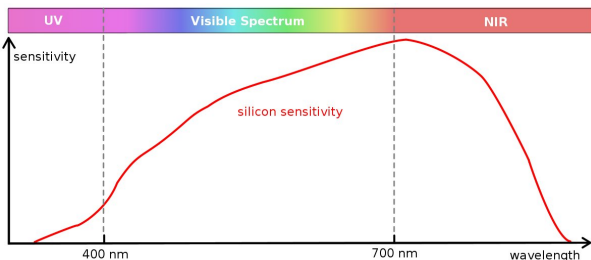


Fig. 2. Graph of silicon sensitivity. An interesting amount of information can be retrieved in the NIR spectrum.

To blend visible and NIR images, we first transform the RGB image into an opponent color space to obtain an intensity-chrominance image representation. The one channel NIR image contains intensity data. For both automatic as well as manual approaches, NIR-blending is performed at image segment level. We pre-segment the image using the mean-shift algorithm [10], [11], [12]. The output of the intensity image  $I_o$  is a weighted sum of the color image intensity  $I_{RGB}$  and  $I_{NIR}$ . The parameter  $\alpha$  decides the ratio between the two:

$$I_o = \alpha I_{RGB} + (1 - \alpha) I_{NIR} \quad (1)$$

## III. AUTOMATIC BLENDING OF NIR AND VS INFORMATION



Fig. 3. A comparison of automatic blending with GUI-aided blending. 3(a) is the original RGB image. 3(b) and 3(c) are the results of automatic blending of NIR and VS inputs. Note that neither the automated global processing nor the automated local processing are able to differentiate between the different objects of the image. This justifies the use of GUI-aided method, which takes advantage of user input to blend images.

In this section we present an automatic approach of blending NIR and VS intensity information. This is done both globally, i.e. by using a single  $\alpha$  value for the entire image, and locally by using a different value of  $\alpha$  for each

segment of the image (pre-segmented by mean-shift<sup>1</sup> [12] segmentation algorithm).

In the case of whole image blending, the image is considered to be the only segment. A set of  $\alpha$  values is defined, and for each value the corresponding blended fragment is computed. The entropy is calculated for every result and the one with the highest entropy is kept to form the output image. This operation is repeated for every segment so that at the end the whole image has been processed. Fig. 3(b) shows the result obtained by globally processing an image. We can see more detail in the shadow area of the object, nevertheless the result looks unnatural. Fig. 3(c) shows the result obtained by locally processing the image. In some cases, adjacent segments of the same object are processed with different  $\alpha$  values. This introduces an undesired edge in the resulting image. Using GUI-aided method (Fig. 3(d)) we can avoid this undesired result by choosing a more suitable  $\alpha$  value.

#### A. Entropy calculation

To compute the entropy of the image, the intensity is quantized into fixed numbers of levels. Entropy is computed as:

$$H = - \sum_{x \in X} p(x) \cdot \log_2 p(x) \quad (2)$$

where  $X = [1, nb]$ ,  $nb$  defines the number of bins used for the quantization,  $x$  is a selected bin, and  $p(x)$  is obtained as:

$$p(x) = \frac{|x|}{\sum_{x \in X} |x|} \quad (3)$$

The number of bins is obtained using Scott's Rule [13], which defines the bin range as follows:

$$\text{bin range} = 3.5 \cdot \text{std}(\text{samples}) N^{1/3} \quad (4)$$

$N$  is the number of samples to be quantized, which corresponds to the number of pixels in the segment.

## IV. GUI-AIDED NIR AND VS IMAGE BLENDING

As seen in the previous section, automatic blending, whether applied globally or locally, is often unsatisfactory since there is no easy way to automatically judge the semantic relevance of the output, which may differ from user to user or task to task. The degree of blending is best decided manually. For this reason, we propose a GUI-aided approach of performing segment based blending. The GUI we propose is shown in Fig. 4. The GUI allows the user to specify a segment and choose the value of  $\alpha$  to use for the NIR-blending.

## V. BOUNDARY ARTIFACT REMOVAL

Segment based blending, though a better approach than global blending, suffers from boundary artifacts. This is because the intensity information at the boundaries may differ enough to introduce a visible edge. We experiment with three different approaches of removing these artifacts, and compare

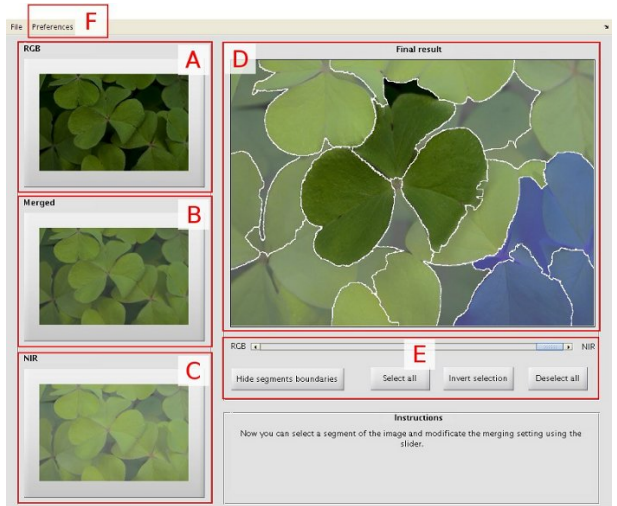


Fig. 4. Screenshot of the proposed GUI. A is the image obtained by global processing with  $\alpha = 0$ . B is the image obtained by global processing with  $\alpha = 0.5$ . C is the image obtained by global processing with  $\alpha = 1$ . The user chooses the starting image (A,B or C) by clicking on it. The chosen image is segmented and loaded in area D. One or more segments are selected by clicking on them or using buttons in panel E. Selected segments are highlighted in blue. The slider in panel E allows to change the  $\alpha$  value for the selected segments. Additional options to influence the segmentation operation and the artifacts removal method are available in the menu F.

them for their pros and cons. Fig. 7 shows an example of processing done on an image using the three different approaches.



Fig. 5. An example of segment boundary artifacts.

#### A. Alpha mask

We define the "alpha mask" to be a two dimensional matrix  $M$  such that  $M_{i,j}$  is the value of  $\alpha$  corresponding to the image pixel  $i,j$ . Applying the automated method or letting the user choose the desired value of  $\alpha$ , we eventually obtain a completely defined alpha mask. This results in a patchwork where values are grouped in areas corresponding to segments and change only along segment edges.

Boundary artifacts are possible when the variation between two adjacent alpha mask values is high. A large difference between  $I_{RGB}$  and  $I_{NIR}$  introduces a visible edge that is particularly disturbing in low-textured smooth regions.

<sup>1</sup><http://www.caip.rutgers.edu/riul/research/code/EDISON/index.html>

To minimize these artifacts, we tested three different ways to process the alpha mask. All the methods modify the values of  $M$  along the boundaries between segments. Fig. 6 shows how an alpha mask may appear before and after the artifact removal correction.



Fig. 6. The alpha mask of a blended image can appear like a patchwork as a consequence of segmentation. Applying an artifact removal blurs the boundaries. The degree of blurring is not necessarily the same over the entire alpha mask.

### B. Gaussian blurring

The most basic approach is to filter the alpha mask using a Gaussian filter (the GUI allows the user to set the standard deviation of the gaussian kernel). This approach is fast but it performs uniform smoothing all along the boundary, smoothing edges even where they are desirable (Fig. 7(b)).

### C. Bilateral filtering

To overcome the limitations of the simple Gaussian blurring we experiment with edge-adaptive blurring using bilateral filtering<sup>2</sup> [14]. The range and space standard deviation values can be modified in the GUI. Predictably, Bilateral filtering gives a much better output than simple Gaussian blurring, as seen in Fig 7(c).

### D. Matting

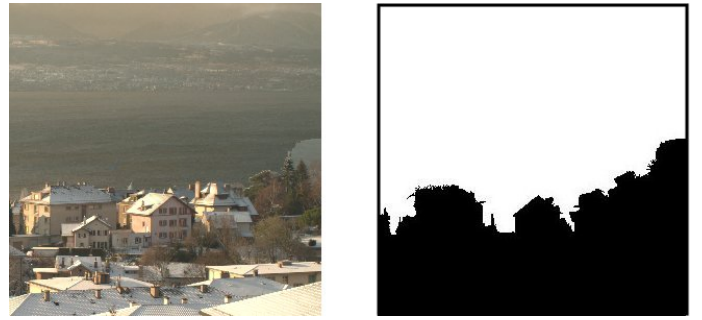
The third method is Levin et al.'s matting technique<sup>3</sup> [15]. It takes as input an image  $I$ , which is assumed to be a composite of a foreground image  $F$  and a background image  $B$ . The intensity of the  $i$ -th pixel is assumed to be a linear combination of the corresponding foreground and background intensity where  $\alpha_i$  is the foreground opacity of the pixel.

$$I_i = \alpha_i F_i + (1 - \alpha_i) B_i \quad (5)$$

All quantities on the right hand side of the compositing equation 5 are unknown. The problem is strictly under-determined and consequently some assumptions have to be made to compute them. Both the foreground  $F$  and the background  $B$  are assumed to be approximately constant over a small window around each pixel. Also, initial conditions specifying background and foreground regions are set.

<sup>2</sup><http://people.csail.mit.edu/jiawen/software/bilateralFilter.m>

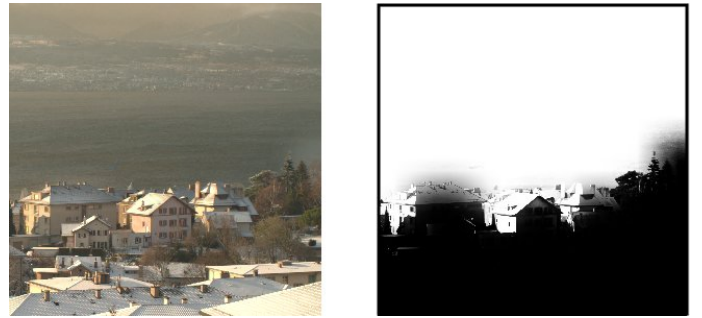
<sup>3</sup><http://people.csail.mit.edu/alevin/matting.tar.gz>



(a) Output without any processing to remove segment boundary artifacts



(b) Segment boundary artifact removal by uniform Gaussian filtering



(c) Segment boundary artifact removal by Bilateral filtering



(d) Segment boundary artifact removal by matting

Fig. 7. Output images and weight masks with different segment boundary processing.

We apply the matting technique to smooth boundaries by specifying a chosen segment as foreground, the rest of the image as background, and leaving regions close to the boundary as unspecified. The width of the unspecified region

can be modified by the user in the GUI. Fig. 8 shows a region subdivision. The procedure is then repeated for each segment.

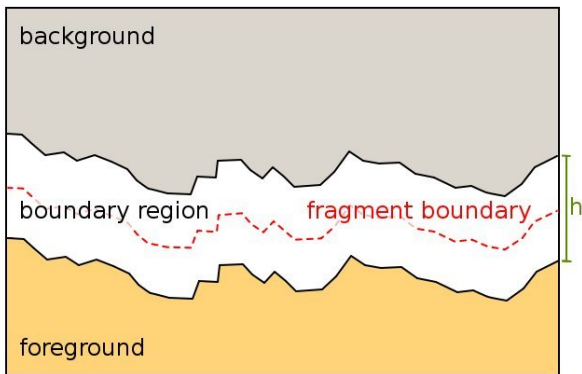


Fig. 8. Inputs regions for the matting artifact removal algorithms. "Foreground" corresponds to the user-chosen segment. "Background" corresponds to the non-chosen segments. The boundary region is the region where the algorithm is applied. The value "h" defines the width of the region. This parameter is a menu item of the GUI and can be set by the user.

### E. Performance comparison

The Gaussian blurring approach, which is computationally the most efficient, is also the most naïve in smoothing boundaries. The Bilateral filtering and the matting approach output far better results. Matting can perhaps offer the best quality output, but is the slowest of the three. In terms of general quality of the output and speed, the best compromise is offered by using the Bilateral filter. A comparison of speeds is shown in Fig. 9.

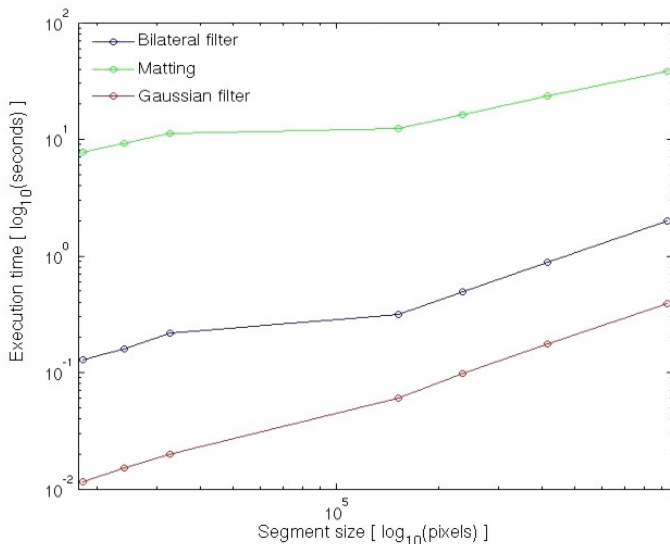


Fig. 9. Time performances of the different methods to process a fragment. The time is measured for different fragment sizes. Results are obtained using a computer equipped with 4 GByte RAM and an Intel Core2 Duo processor with core frequency of 2.66 GHz. Note that the matting method is approximately two orders of magnitude slower than Bilateral filtering, which in turn is one order of magnitude slower than Gaussian filtering.

## VI. CONCLUSION

We present two ways of blending NIR and visible information to improve visual image quality - automatically, by globally and locally maximizing entropy, and GUI-aided, where users can manually choose regions and their blending ratios. Since we perform segment-based blending, both approaches result in boundary artifacts. Therefore, we also propose three ways of removing boundary artifacts and compare their results. Our work shows that NIR information and VS information can be combined to obtain images with few artifacts and good visual quality. The limitation of course is that at present, the easiest way to obtain NIR and VS images is to capture two separate images. This may no longer be a limitation when the use of sensors that simultaneously capture both NIR and VS information [16] becomes widespread.

## REFERENCES

- [1] X. M. Xiaopeng Zhang, Terence Sim, "Enhancing photographs with near infra-red images," *IEEE Proceedings of Computer Vision and Pattern Recognition*, pp. 1–8, 2008.
- [2] S. Li, R. Chu, S. Liao, and L. Zhang, "Illumination invariant face recognition using near-infrared images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 4, pp. 627–639, April 2007.
- [3] D. Lu, E. Moran, and M. Batistella, "Linear mixture model applied to amazonian vegetation classification," *Remote Sensing of Environment*, vol. 14, pp. 456 – 469, 2003.
- [4] N. M. Milton, B. A. Eisworth, and C. M. Ager, "Effect of phosphorus deficiency on spectral reflectance and morphology of soybean plants," *Remote sensing of environment*, vol. 36, pp. 121–127, 1991.
- [5] D. A. Burns and E. W. Ciurczak, *Handbook of Near-Infrared Analysis*. CRC Press, 1999.
- [6] M. Blue and S. Perkowitz, "Reflectivity of common materials in the submillimeter region," *IEEE Transactions on Microwave Theory and Techniques*, vol. 25, no. 6, pp. 491–493, Jun 1977.
- [7] M. Vilaseca, J. Pujol, M. Arjona, and F. M. Martinez-Verdu, "Color visualization system for near-infrared multispectral images," *The Journal of imaging science and technology*, vol. 49, pp. 246–255, 2005.
- [8] L. Schaul, C. Fredembach, and S. Susstrunk, "Color Image Dehazing using the Near-Infrared," in *IEEE International Conference on Image Processing 2009*, 2009.
- [9] C. Fredembach and S. Susstrunk, "Colouring the near infrared," *Proceedings of the IS&T/SID 16th Color Imaging Conference*, pp. 176–182, 2008.
- [10] P. M. Dorin Comanicu, "Mean shift: A robust approach toward feature space analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, pp. 603–619, 2002.
- [11] B. G. Peter Meer, "Edge detection with embedded confidence," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, pp. 1352–61 365, 2001.
- [12] C. Christoudias, B. Georgescu, and P. Meer, "Synergism in low level vision," *IEEE Conference on Pattern Recognition*, vol. 4, pp. 150–155, 2002.
- [13] G. Finlayson, M. Drew, and C. Lu, "Intrinsic images by entropy minimization," in *European Conference on Computer Vision*, 2004.
- [14] S. Paris, P. Kornprobst, J. Tumblin, and F. Durand, "A gentle introduction to bilateral filtering and its applications," 2007.
- [15] A. Levin, D. Lischinski, and Y. Weiss, "A closed-form solution to natural image matting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 2, pp. 228–242, 2008.
- [16] Y. M. Lu, C. Fredembach, M. Vetterli, and S. Susstrunk, "Designing color filter arrays for the joint capture of visible and near-infrared images," in *International Conference on Image Processing*, 2009.



(a) Input RGB image



(b) Input RGB image



(c) Input NIR image



(d) Input NIR image



(e) Output obtained using the GUI



(f) Output obtained using the GUI

Fig. 10. Other examples of inputs and output of the proposed method.