Soft Segmentation for Comparative Image Editing

Anna Regina Corbo Luiz Velho

Instituto Nacional de Matemática Pura e Aplicada
Visgraf Laboratory
$\{corbo, lvelho\}$@impa.br

Abstract

We present a new tool for comparative image editing of HDR and LDR images that preserves edges by doing a soft segmentation. The user selects with strokes different regions to be modified and has control over all the regions simultaneously to make adjustments in parameters such as saturation, contrast and exposure. The final mask that maps those regions preserves the most important transformation areas and the smoothness of transition regions.

1. Introduction

The use of digital cameras and image editing applications has gained popularity, therefore the demand for advanced tools has increased. In this context, one of the most common editing operations consists in the modification of tonal parameters such as saturation, contrast, color temperature, or exposure over some specific region of the image. Available tecniques just make tonal value adjustments globally and to make local adjustments it is necessary to carefully select regions or hand-paint layer masks.

Local adjustments require the segmentation of the region of interest in the image. There are many segmentation algorithms in the literature, but in most of them the image segmentation is based only on edge-preserving methods.

Our work is closely inspired by the work of Lischinski *et al.* [2] and Levin *et al.* [1]. They developed a tool for local adjustments of tonal values where the user indicates the region of interest by drawing some strokes and indicates how and what are the parameters to be adjusted. The effects are interpolated to the entire image, but only regions similar the original one are changed.

Here, we extend this method for handing multiple regions in an image based on a Partition of Unity segmentation. These areas are treated as unit parts where the sum of all the transformations is one. Our method preserves region characteristics while allowing the user to do different local adjustments of tonal values over multiple regions in parallel by just changing the parameter values.

The tool has produced good results for HDR radiance maps and for LDR ordinary images, where multiple regions were simultaneously adjusted without loosing the global smoothness. There, after choosing an HDR radiance map or an ordinary LDR image, the user indicate with strokes the different areas in the image that will be modified and, for each selected area, adjusts the parameter values as desired until satisfactory results. Figure 1 shows the workflow of our tool described in section 2 and a example result.

2. Comparative Image Editing

2.1. Soft Segmentation and Local Editing

When a stroke is done, it means that the region next to it will be modified. If a pixel x is similar to the pixels $\bar{x}$ marked then pixel x will be selected too. The similarity criteria are based on an appropriate metric.

If f is the scalar function that specify how the parameter P will be adjusted in each pixel, it is desirable that f is close to user specifications and that f is as smooth as possible. In these cases, f can be described as a quadratic functional expressed as:

$$f = \text{argmin}_f \{\sum_x w(x)(f(x)-g(x))^2+\lambda\sum_x h(\nabla f, \nabla L)\}$$

Minimizing this functional is the same as solving the linear system $Af = b$, where $b_i = w_i g_i$, and

$$A_{ij} = \begin{cases} -\lambda(\|L_i - L_j\|^\alpha + \varepsilon)^{-1}, j \in N_4(i); \\ w_i - \sum_{k\in N_4(i)} A_{ik}, i = j; \\ 0. \quad \text{otherwise.} \end{cases}$$

Here, the indices i, j represent pixels i and j in the image and $N_4(i)$ are the 4-neighbors of pixel i; λ, α, ε are constants; L_i is the log-luminace channel in pixel i; g_i is the value of the user specification of adjustment in pixel i and w_i is the weight function calculate in pixel i, according to the choosen metric.

2.2. Editing over Multiple Regions

In our approach, for each stroke s a mask M_s is calculated using the method described in section 2.1. The mask that allows the image multiple control is done by a interpolation $\bar{M}$ as above.

Taking an image as a unit, each region stroked will be a complementary unit part. If two stokes are done, then two masks, M_1 and M_2, are calculated. The interpolation mask is done by a sum of two masks M_1' and M_2', where for each pixel x,

$$M_1'(x) = M_1(x)$$

$$M_2'(x) = (1 - M_1)(x) \cdot M_2(x)$$

i.e, M_2' is a subset of the inverse of M_1'. In this way, where the larger M_1' adjustments were done, pratically no change will be made by M_2'. Then the most important adjusts will be conserved.

The value of a pixel x in the interpolation mask M_{12} is given by

$$M_{12}(x) = M_1'(x) + M_2'(x)$$

This method preserves the local adjustments, keeping the smoothness of the original transitions areas and can be used in the same way for n different selected regions, where:

$$\sum_{i=1}^{n} M_i' \approx 1.$$

When the user adjusts a region R, the corresponding mask M_R is adjusted and so is the global mask $\bar{M}$. This operations preserves the main adjustments made by the other masks. The result is a soft partition of unit, preserving weights defined by the original strokes. The results may differ according to the order of mask insertion in the partition of unit structure. This feature allows the user to define which are the dominant adjustments to be made in the image. Moreover, different options of structure can be precomputed, making possible for users to apply the masks in different orders and then choose the one which has the best results.

3. Conclusion

The presented technique is based on a stroke-based interface where the user selects areas of interest and a simple method to calculate the soft layer-masks of adjustments. Then a simple interpolation is done, supposing that for each main change there is a region in image that is predominant.

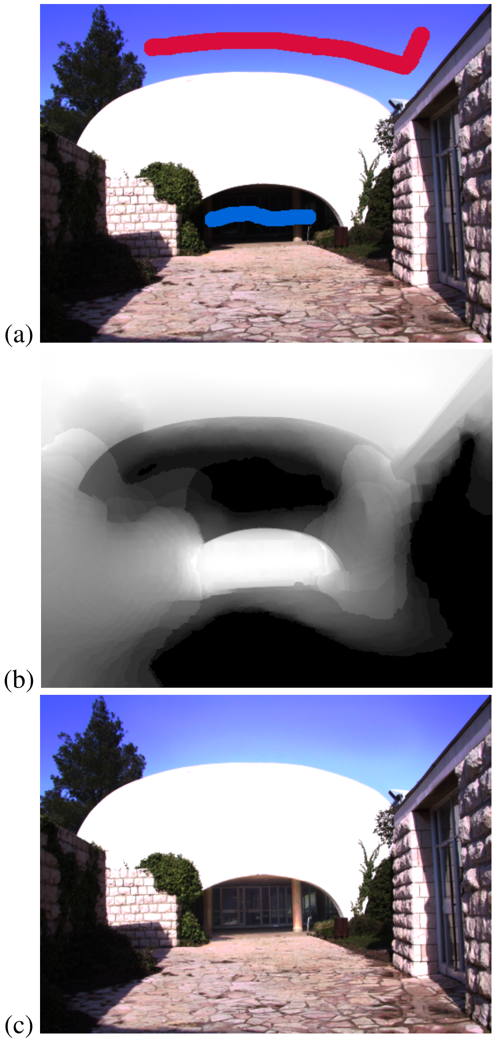

Figure 1. a) Original image with strokes; b) Combination mask; c) Result. (*Synagogue* image by The Hebrew University)

In future work, we would like to test other kinds of interpolation such as a non-linear mean, where the pixels with larger weight could be more representative than pixels in transition regions. Moreover, we would like to create a interface where the user will be able to see in real time a preview of his adjustments while stroking.

References

[1] A. Levin, D. Lischinski, and Y. Weiss. Colorization using optimization. *ACM Transactions on Graphics*, 23(3):689–694, 2004.

[2] D. Lischinski, Z. Farbman, M. Uyttendaele, and R. Szeliski. Interactive local adjustment of tonal values. *Proceedings of SIGGRAPH 2006*, 25:646–653, 2006.