Region-based color correction

Dieu-Sang Ly, Serge Beucher and Michel Bilodeau

January 14, 2014

1 Introduction

Color correction refers to modifying the color of an input image so that it is similar to the color of a reference image. They can be classified into two categories: one based on pixel correspondences and the other based on the color statistics of two images. Methods that employ pixel correspondences rely on correspondence accuracy where as statistical methods do not require exact correspondences. There are two main types of statistical approaches: histogram matching [3] and color distribution modification. The first one assumes that both images are acquired under the same illumination and from the same viewpoint. This strong assumption rarely holds, especially in color correction application. The second one utilizes the color transfer function [7]. The global color correction, where the parameters of the color transfer are computed from the entire input and reference images, can produce a correct result uniquely when two images observe almost exactly the same regions under the same illumination. In this work, we address and solve the problem of color correction of two images captured under different conditions or from different cameras. We suggest a local color correction method, where different color transfer functions are computed from and applied to individual regions of the input image.

2 Region-based color correction

The region-based color correction consists of three tasks: segment both input and reference images into regions, search for region correspondences between these images and apply the color transfer to the input image.

2.1 Region segmentation

The original image is segmented into regions using watershed transformation [1]. The idea is to consider a gray-scale image as a topographic relief and to flood this relief from different sources until they start to merge. This results in watershed lines separating different catchment basins. In addition, predefined markers can be used as flooding sources to control the segmentation, e.g. to avoid over segmentation. The marker-controlled watershed segmentation is composed of the following steps:

1. Computation of segmentation criterion and markers
   In order to partition the image into homogeneous regions, we can use the image gradient as the segmentation criterion (or the topographic relief mentioned above) since the gradient value is low within a homogeneous region and high at its boundary. The markers should be inside the regions, hence can be computed from the local minima of the gradient image or by applying a threshold to the gradient image. Note

that we compute the gradient from color image, i.e. the maximum of the gradients of all color channels, which preserves region boundaries better than the gradient from gray-level image, as illustrated in figure 1.

2. Marker-controlled watershed segmentation
   The image gradient and markers are provided to watershed segmentation. If the resulting regions are more numerous than expected, we can run an additional region fusion step, which is summarized in algorithm 1. If the color difference between two adjacent regions is inferior to a given threshold, the BoundaryEliminator operator eliminates their inner boundary and keeps their outer boundaries with other regions in order to avoid incorrect boundary elimination and region fusion. Figure 2 shows the input and reference images together with their watershed segmentation and region fusion. These images are captured by two cameras with different photometric parameters.

Algorithm 1 Region fusion. Each region $\mathbf{R}_i$ is assigned a mean color, computed by averaging the color of all pixels belonging to that region in L$\alpha\beta$ space, and an isMergedInto indicator, i.e. the index of the region that $\mathbf{R}_i$ is merged into. isMergedInto($\mathbf{R}_i$) $= -1$ if $\mathbf{R}_i$ is not merged into any region. The color difference between two regions is the Euclidean distance between their mean color in L$\alpha\beta$ space.

```
N ← number of initial regions
isMergedInto(R_i) ← −1 where i = 1...N
thres ← threshold to merge 2 regions
for i ← 1 to N − 1 do
  for j ← i + 1 to N do
    if R_i and R_j are adjacent and their color difference is less than thres then
      Retrieve the biggest region that R_i and R_j have been possibly merged to
      if R_i and R_j have not been merged to any region then
        BoundaryEliminator(R_i,R_j)
        isMergedInto(R_j) ← i
      else if R_i has not been merged to any region and R_j has been merged to R_l then
        BoundaryEliminator(R_i,R_l)
        isMergedInto(R_i) ← l
      else if R_i has been merged to R_k and R_j has not been merged to any region then
        BoundaryEliminator(R_k,R_j)
        isMergedInto(R_j) ← k
      else if R_i and R_k have been merged to the same region R_k then
        BoundaryEliminator(R_k,R_j)
      else if R_i and R_k have been merged to two different regions R_k and R_l then
        BoundaryEliminator(R_k,R_l)
        isMergedInto(R_j) ← k
        isMergedInto(R_l) ← k
      end if
    end if
  end for
end for
```

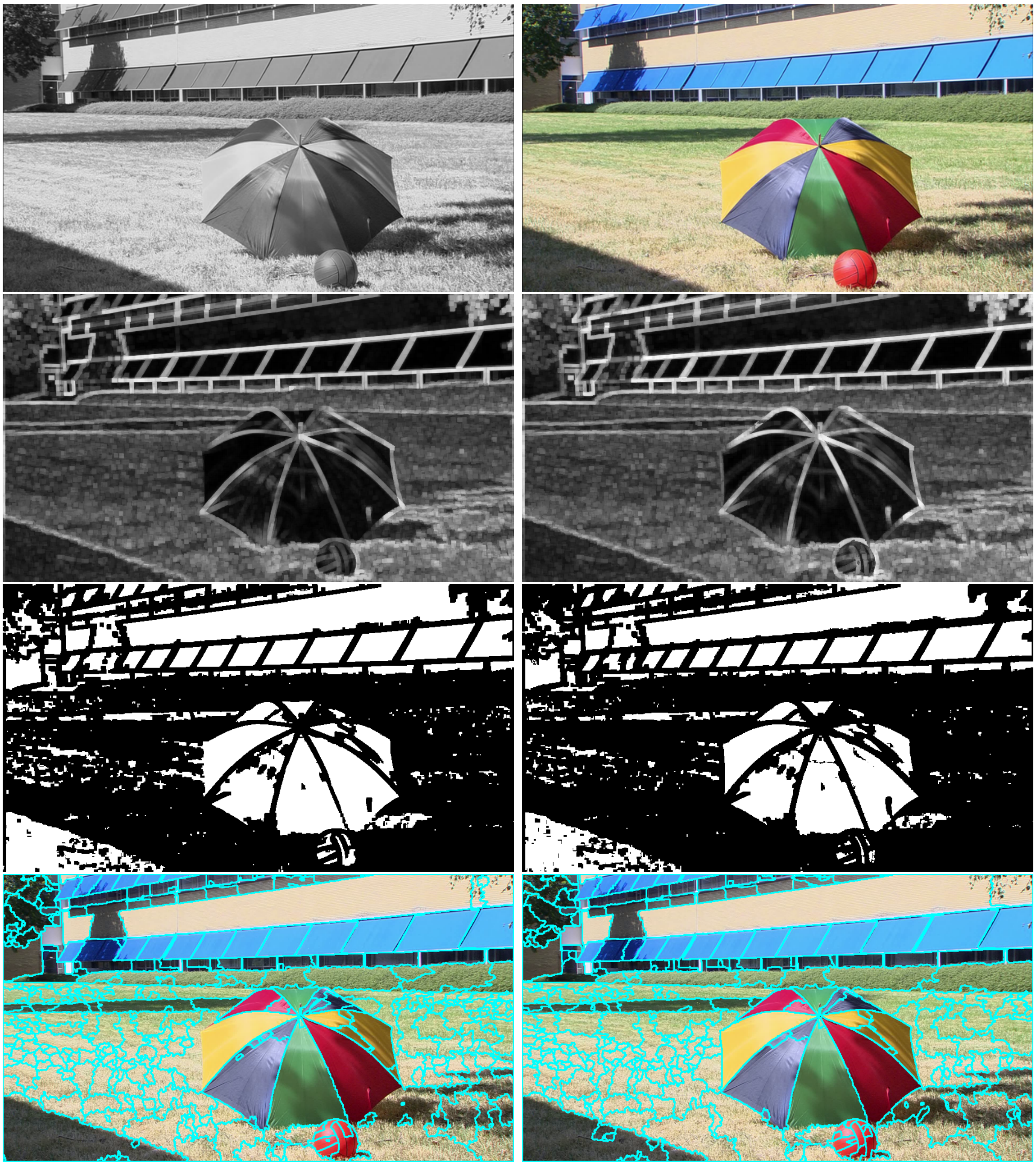

Figure 1: Watershed segmentation from the image gradient and markers. Row 1: input image in gray-level (left) and color (right). Row 2: image gradient from gray-level (left) and color (right) images. Row 3: markers computed from the image gradient and a threshold, post-processing such as opening/closing has been used to refine markers. Row 4: watershed segmentation. Color gradient preserves region boundaries better than gray-level gradient. For instance, the outer edges of the umbrella with low contrast remains in the segmentation with the color gradient but disappears with the gray-level gradient.

Figure 2: Region segmentation and fusion. Row 1: reference (left) and input (right) images. Row 2: watershed segmentation. Row 3: region fusion. The watershed line is one-pixel large, but we increase their thickness for better visualisation.

2.2 Region matching

The first solution to region matching is based on region similarity, i.e. match regions using their color, location, area or other characteristics. On one hand, using color as matching criterion requires that the color of two regions be sufficiently similar, hence is not relevant in color correction application. On the other hand, location and area criteria can fail in case of complex transformation between two images, for example important zooming or translation. Figure 3 shows the region matching between two segmented images in figure 2 using their color and location. First, the region color is calculated from the average of the color of all region pixels in $L\alpha\beta$ space. The region location is assumed to be at the region centroid. Next, given a threshold of region location, we search, in the neighbourhood of each region of the input image, the region of the reference image having the closest color. As the depth of the scene is important in these two images, the regions distant from the cameras have a significant translation between two images whereas the regions close to the cameras does not. Consequently, the correspondences are erroneous when we use a single threshold of region location. Searching region correspondences based on location criterion is not robust to image transformation. Moreover, the illumination of these two images is so different, therefore using the color in region matching produces incorrect result.

Figure 3: Region matching based on the color and location of regions

The second solution to region pairing assumes known image registration [6]. Each region in the segmented input image is projected to the unsegmented reference image using the image transformation and the overlaid region is considered as its match. The first drawback of this method is that it requires coarsely registered images. Otherwise, it is impossible to map all regions of one image to the other. It is hard to handle accurate transformations all over the image due to the extrapolation problem [4], which means that a transformation can correctly map the image region straddled by points used to compute that transformation and is less accurate with distance from this region. Figure 4 illustrates the extrapolation problem, where it is not possible to find a single transformation valid for the entire image. The second drawback of this method is that the region projection will result in wrong matches in case of occultation, see figure 5. In other words, if a region of the input image is projected to an occult region of the reference image, the computation of the color of this occult region is incorrect. As a consequence, the parameters of the color transfer computed from this region are wrong.

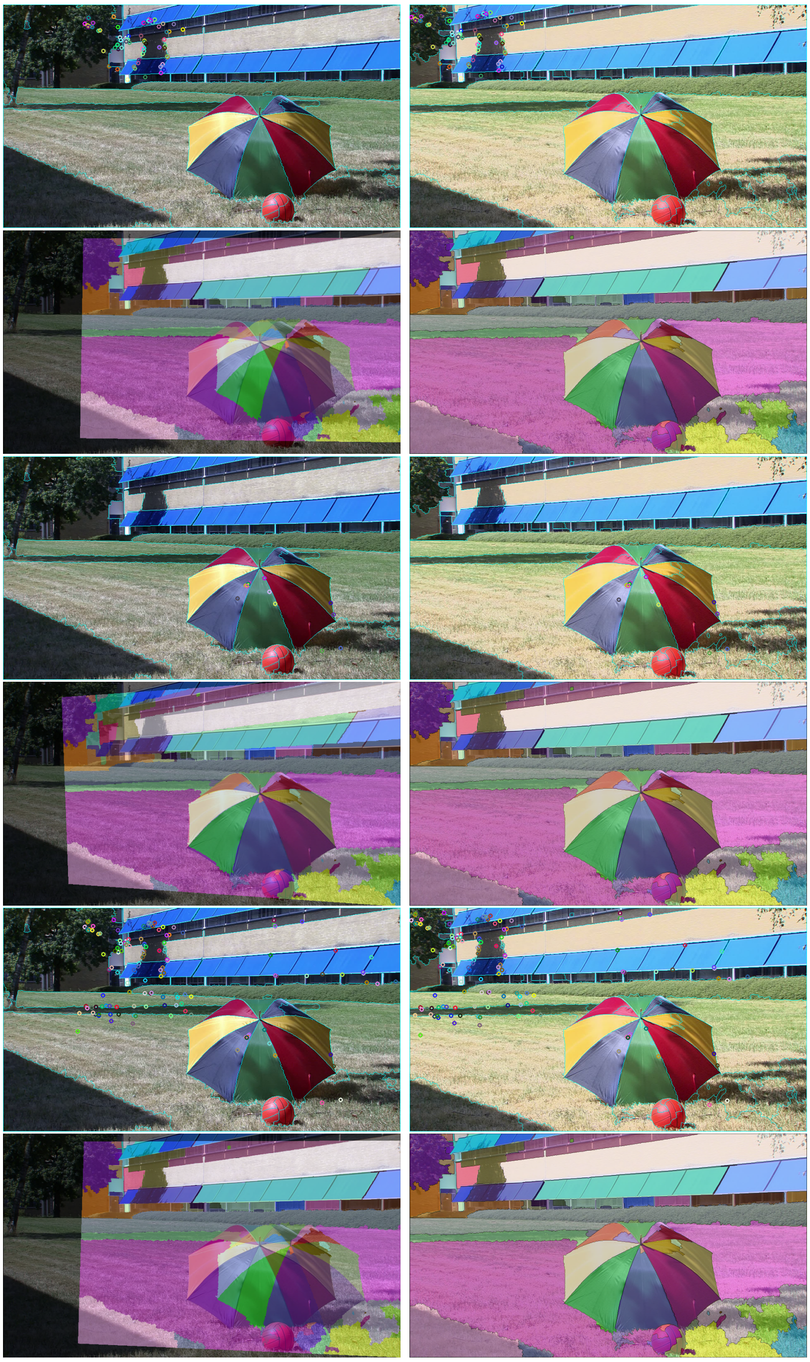

Figure 4: Region matching based on known image transformation. Row 1: point correspondences used to compute the image transformation. Row 2: regions in the input image (right) are projected to the reference image (left) to look for their region matches. Similarly, another set of point correspondences (row 3) results in another image transformation which is used to project the regions of the input image to the reference image (row 4). It can be seen that the transformation calculated from a set of points can accurately project only the regions containing these points. The transformation is not accurate for regions far from these points. Even if point correspondences are distributed over the images (row 5), the transformation still can handle only regions straddled by the most accurate point correspondences (row 6).

Figure 5: Example of region occultation. The region corresponding to the green cup in the right image may be projected to an occult region in the left image which contains the green cup and part of the yellow cup.

As the challenge is to search region correspondences having dissimilar colors, we seek for an approach of region pairing using a criterion robust to color variation. In this work, we propose a method of region matching leveraged by image point features as follows

1. The input and reference images are both segmented into regions.

2. Point correspondences between these images are estimated by a robust approach of point feature detection and matching.

3. Each point is assigned to the region that it belongs to. Multiple points may locate in the same region. We discard points lying on the region borders (or the watershed lines).

4. Two regions are matched if they are straddled by matched points. An example is illustrated in table 1. Regions $\mathbf{R}_i$ and $\mathbf{R'}_m$ in the reference and input images respectively are matched as they contain matched points $\mathbf{p}_1$ and $\mathbf{p'}_1$.

5. Finally, we merge regions in case of one-to-multiple matching, which may happen when a region in one image corresponds to several adjacent regions in the other image. For example, $\mathbf{R}_i$ and $\mathbf{R}_k$ are matched to $\mathbf{R'}_m$, hence $\mathbf{R}_i$ and $\mathbf{R}_k$ are merged. Similarly, $\mathbf{R'}_n$ and $\mathbf{R'}_q$ are merged as they are matched to one region $\mathbf{R}_j$.

Reference image		Input image	
Regions	Point correspondences	Point correspondences	Regions
$\mathbf{R}_i$	$\mathbf{p}_1$	$\mathbf{p'}_1$	$\mathbf{R'}_m$
$\mathbf{R}_j$	$\mathbf{p}_2$	$\mathbf{p'}_2$	$\mathbf{R'}_n$
$\mathbf{R}_k$	$\mathbf{p}_3$	$\mathbf{p'}_3$	$\mathbf{R'}_m$
...	...	...	...
$\mathbf{R}_j$	$\mathbf{p}_M$	$\mathbf{p'}_M$	$\mathbf{R'}_q$

Table 1: Region matching leveraged by point correspondences

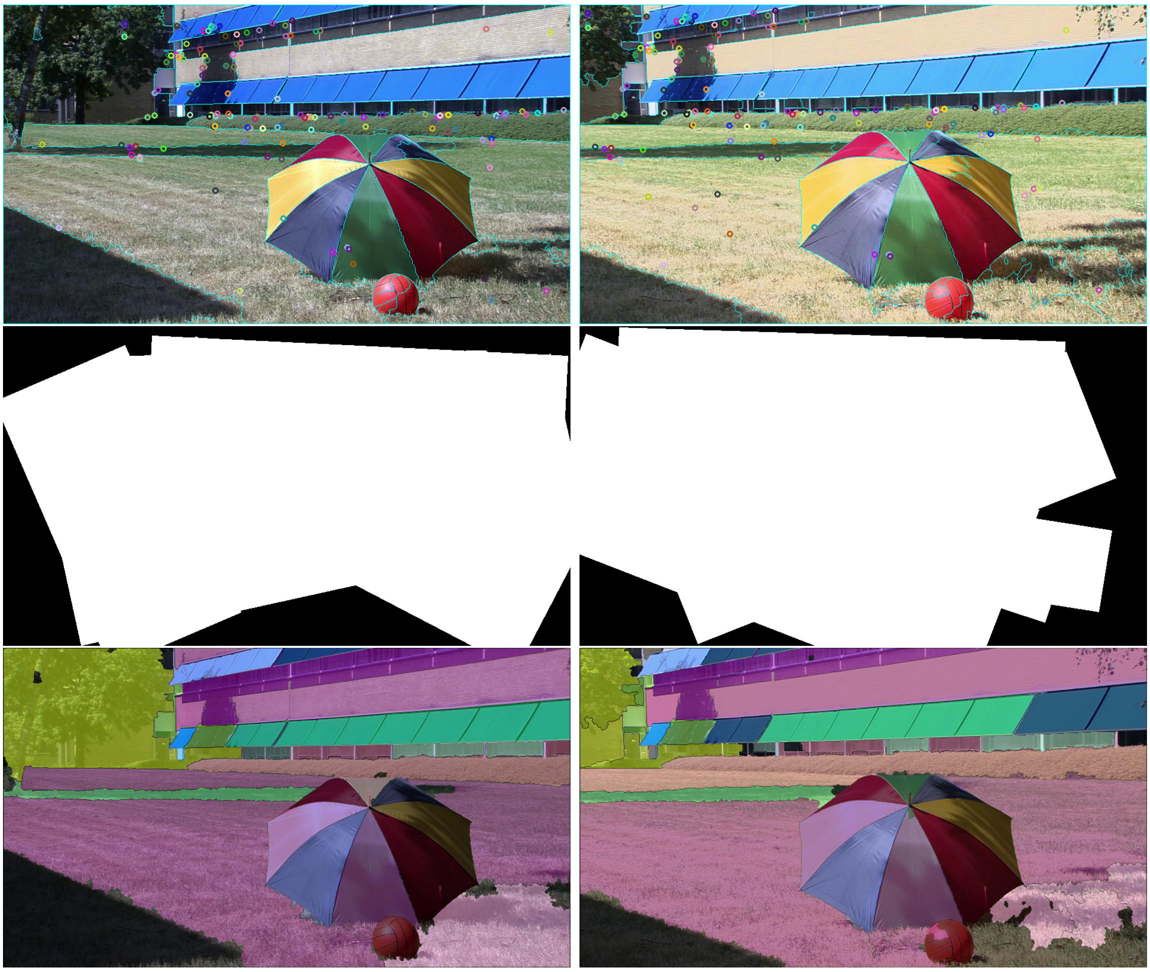

Figure 6: Region matching leveraged by point correspondences. Row 1: point correspondences by incremental tiling. Row 2: the final masks bounding these points. Row 3: region correspondences in same color.

Point correspondences between the input and reference images are estimated through the next steps

1. Detect SIFT features and extract their descriptors using SIFT (Scale-Invariant Feature Transform) [5]. SIFT features are well known for the robustness to illumination change and several basic image transformations.

2. Compute initial matches using Brute-Force matcher: for each feature in one image, find the closest among all features in the other image using the distance between SIFT descriptors.

3. Select good matches among the initial matches using a threshold of the distance between a pair of descriptors.

4. Refine the good matches by an iterative method such as RANSAC [2] with the fitting model being the homography between two images. Note that this step is used to obtain the inliers and the output model is discarded.

If we apply the previous steps once, the point correspondences are the best matches fitted to the best homography, therefore it is not guaranteed that they are spatially distributed all over the image, as shown in figure 4. In order to overcome this limited distribution, we implemented an incremental tiling approach: after one set of point correspondences is found, we mask the image part straddled by these points in both images and search for point correspondences within the unmasked image part. This mask can be computed by a rotated rectangle or a convex hull bounding a set of point. The point matching with mask is repeated until the final mask covers most of the image, 80% in our case.

Figure 6 illustrates the result of point matching by the incremental tiling technique and the region matching leveraged by these points.

2.3 Color transfer

The general color transfer function was introduced by Reinhard et al. [7] to scale and offset the color distribution of an input image towards a reference image.

$$\mathbf{C}_{\mathrm{o}} = \mu_{\mathrm{r}} + \frac{\sigma_{\mathrm{r}}}{\sigma_{\mathrm{i}}}(\mathbf{C}_{\mathrm{i}} - \mu_{\mathrm{i}}) \tag{1}$$

where:

- $\mathbf{C}_{\mathrm{i}}$: input image
- $\mathbf{C}_{\mathrm{o}}$: output of color transfer
- $(\mu_{\mathrm{i}}, \sigma_{\mathrm{i}})$ and $(\mu_{\mathrm{r}}, \sigma_{\mathrm{r}})$: (mean, standard deviation) of the input and reference images

If we apply a single transfer function to every pixel of the input image, which means that μ_{i}, σ_{i}, μ_{r} and σ_{r} are computed from the entire input and reference images, then the color correction is considered as global, see figure 8. The color transfer is applied separately to each channel of the input image in L$\alpha\beta$ space. It can be seen that the blue color of the umbrella and the outdoor blinds and the yellow color of the grass, the umbrella and the building wall are not well corrected. The reason is that a global color transfer cannot correctly map two images having so different colors.

Similarly, we apply the color transfer to each segmented region of the input image, that refers to region-based color correction. For each pair of regions k found in the region matching step, we compute the color distribution μ_{i}^k, σ_{i}^k, μ_{r}^k and σ_{r}^k, which are the parameters of the color transfer between these two regions. Given N region matches between the input and reference images, the color correction is a combination of N local color transfers. In addition, in order to ensure a smooth color shading across the color-corrected image, each local color transfer is weighted by an influence mask $\mathbf{IM}^k$, which measures the similarity of each pixel of the input image and the mean color of the region in consideration.

The influence mask of a region k having the mean color μ_{i}^k is generated as follows

1. The Euclidean distance between every pixel of the input image $\mathbf{C}_{\mathrm{i}}$ and the mean color of region μ_{i}^k in L$\alpha\beta$ space
$$\mathbf{D} = ||\mathbf{C}_{\mathrm{i}} - \mu_{\mathrm{i}}^k|| \tag{2}$$
2. In order to normalize the maximum of $\mathbf{D}$ which varies from different regions, we introduce the following distance bounded by 0 and 1
$$\mathbf{P} = 1 - \frac{\mathbf{D}}{\max(\mathbf{D})} \tag{3}$$
An element of $\mathbf{P}$ approaches 1 when the color of the corresponding pixel in $\mathbf{C}_{\mathrm{i}}$ is close to μ_{i}^k and 0 vice versa.

3. The influence mask
$$\mathbf{IM} = e^{a\mathbf{P}^b} \tag{4}$$
where $a = 10$ and $b = 2$ in our case.

Figure 7 shows the influence masks calculated for three example regions.

Finally, the region-based color correction combines the color transfer functions of N region correspondences weighted by N influence masks

$$\mathbf{C}_\mathrm{o} = \frac{\sum_{k=1}^{N}(\mu_\mathrm{r}^k + \frac{\sigma_\mathrm{r}^k}{\sigma_\mathrm{i}^k}(\mathbf{C}_\mathrm{i} - \mu_\mathrm{i}^k)) \times \mathbf{IM}^k}{\sum_{k=1}^{N}\mathbf{IM}^k} \tag{5}$$

The result of region-based color correction is illustrated in figure 8.

3 Conclusions

We have proposed an approach of color correction based on region segmentation and matching. Region segmentation is achieved by the fast and effective watershed transform and optionally region fusion. Then, segmented regions are paired using point feature correspondences. Finally, the color transfer weighted by masks is applied to modified the color distribution of the image. This method is applicable to images captured by cameras with different view points and photometric parameters.

Future works will include the evaluation of the proposed color correction technique and possibly the improvement of region pairing by the propagation of the segmentation of the input image towards the reference image.

References

[1] Serge Beucher and Christian Lantuejoul. Use of watersheds in contour detection. In *International Workshop on Image Processing*, 1979.

[2] Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, June 1981.

[3] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing (3rd Edition)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2006.

[4] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, New York, NY, USA, 2 edition, 2003.

[5] David G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the International Conference on Computer Vision-Volume 2 - Volume 2*, ICCV '99, pages 1150–, Washington, DC, USA, 1999. IEEE Computer Society.

[6] Miguel Oliveira, Angel Domingo Sappa, and Vítor Santos. Unsupervised local color correction for coarsely registered images. In *CVPR*, pages 201–208, 2011.

[7] Erik Reinhard, Michael Ashikhmin, Bruce Gooch, and Peter Shirley. Color transfer between images. *IEEE Comput. Graph. Appl.*, 21(5):34–41, September 2001.

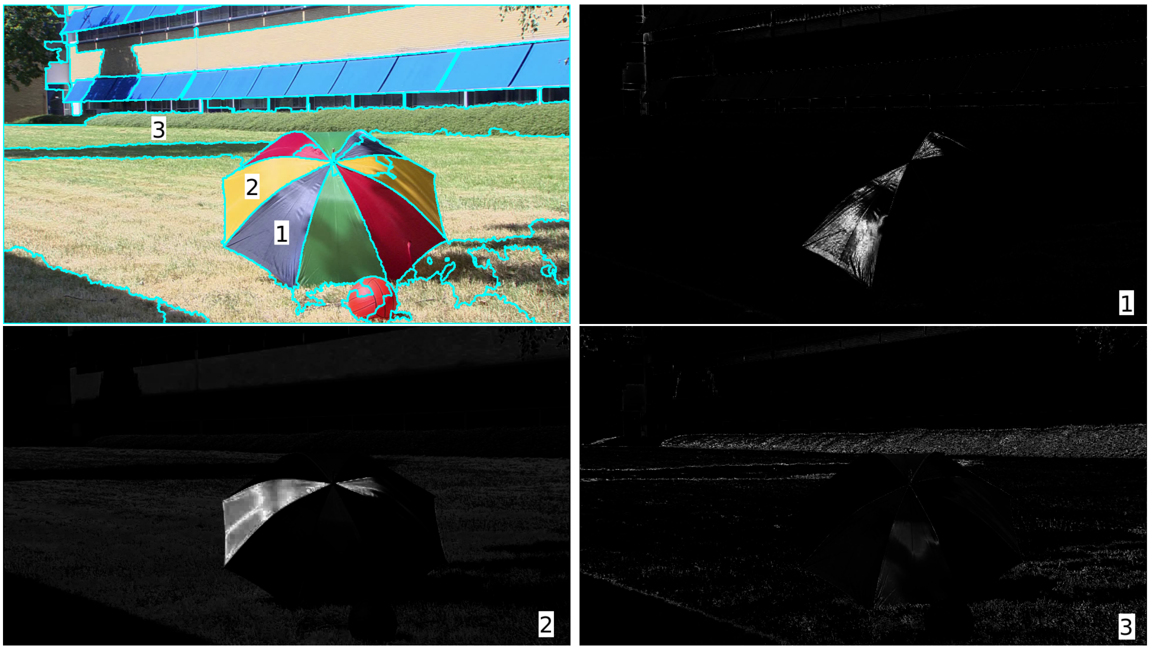

Figure 7: Three example regions of the input image that have been matched with the reference image and their corresponding influence masks. The closer the color of a pixel is to the mean color of the particular region, the higher its value in the influence mask even though it is not within the region.

Figure 8: Color correction. Row 1: reference image (left) and input image (right). Row 2: output of global color correction (left) and region-based color correction (right)