Saliency Driven Image Manipulation

Roey Mechrez
Technion
roey@tx.technion.ac.il

Eli Shechtman
Adobe Research
elishe@adobe.com

Lihi Zelnik-Manor
Technion
lihi@ee.technion.ac.il

Abstract

Have you ever taken a picture only to find out that an unimportant background object ended up being overly salient? Or one of those team sports photos where your favorite player blends with the rest? Wouldn't it be nice if you could tweak these pictures just a little bit so that the distractor would be attenuated and your favorite player will stand-out among her peers? Manipulating images in order to control the saliency of objects is the goal of this paper. We propose an approach that considers the internal color and saliency properties of the image. It changes the saliency map via an optimization framework that relies on patch-based manipulation using only patches from within the same image to maintain its appearance characteristics. Comparing our method to previous ones shows significant improvement, both in the achieved saliency manipulation and in the realistic appearance of the resulting images.

1. Introduction

Saliency detection, the task of identifying the salient and non-salient regions of an image, has drawn considerable amount of research in recent years, e.g., [14, 18, 22, 33, 35]. Our interest is in manipulating an image in order to modify its corresponding saliency map. This task has been named before as *attention retargeting* [24] or *re-attentionizing* [27] and has not been explored much, even though it could be useful for various applications such as object enhancement [25, 27], directing viewers attention in mixed reality [26] or in computer games [3], distractor removal [13], background de-emphasis [30] and improving image aesthetics [15, 31, 34]. Imagine being able to highlight your child who stands in the chorus line, or making it easier for a person with a visual impairment to find an object by making it more salient. Such manipulations are the aim of this paper.

Image editors use complex manipulations to enhance a particular object in a photo. They combine effects such as increasing the object's exposure, decreasing the background exposure, changing hue, increasing saturation, or blurring the background. More importantly, they adapt the manipu-

(a) Input image   (b) Input saliency map

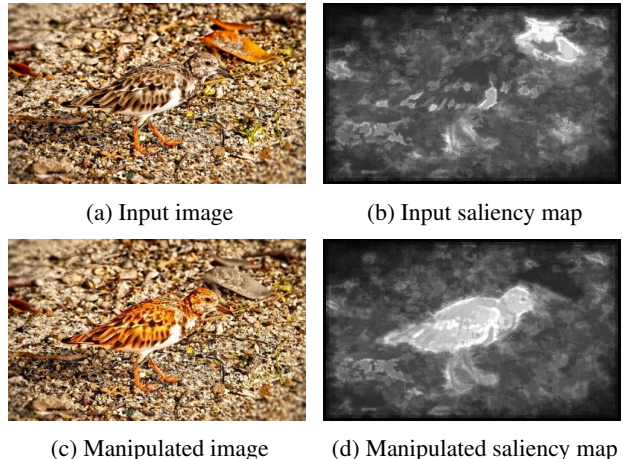(c) Manipulated image   (d) Manipulated saliency map

Figure 1: Our saliency driven image manipulation algorithm can increase or decrease the saliency of a region. In this example the manipulation highlighted the bird while obscuring the leaf. This can be assessed both by viewing the image before (a) and after (c) manipulation, and by the corresponding saliency maps (b),(d) (computed using [22]).

lation to each photo – if the object is too dark they increase its exposure, if its colors are too flat they increase its saturation etc. Such complex manipulations are difficult for novice users that often do not know what to change and how. Instead, we provide the non-experts an intuitive way to highlight objects. All they need to do is mark the target region and tune a single parameter, that is directly linked to the desired saliency contrast between the target region and the rest of the image. An example manipulation is presented in Figure 1.

The approach we propose makes four key contributions over previous solutions. First, our approach handles multiple image regions and can either increase or decrease the saliency of each region. This is essential in many cases to achieve the desired enhancement effect. Second, we produce realistic and natural looking results by manipulating the image in a way that is consistent with its internal characteristics. This is different from many previous methods

that enhance a region by recoloring it with a preeminent color that is often very non-realistic (e.g., turning leaves to cyan and goats to purple). Third, our approach provides the user with an intuitive way for controlling the level of enhancement. This important feature is completely missing from all previous methods. Last, but not least, we present the first benchmark for object enhancement that consists of over 650 images. This is at least an order of magnitude larger than the test-sets of previous works, that were satisfied with testing on a very small number of cherry-picked images.

The algorithm we propose aims at globally optimizing an overall objective that considers the image saliency map. A key component to our solution is replacing properties of image patches in the target regions with other patches from the same image. This concept is a key ingredient in many patch-bases synthesis and analysis methods, such as texture synthesis [11], image completion [1], highlighting irregularities [5], image summarization [29], image compositing and harmonization [9] and recently highlighting non-local variations [10]. Our method follows this line of work as we replace patches in the target regions with similar ones from other image regions. Differently from those methods, our patch-to-patch similarity considers the saliency of the patches with respect to the rest of the image. This is necessary to optimize the saliency-based objective we propose. A key observation we make is that these patch replacements do not merely copy the saliency of the source patch to the target location as saliency is a complex global phenomena (similar idea was suggested in [7] for saliency detection). Instead, we interleave saliency estimation within the patch synthesis process. In addition, we do not limit the editing to the target region but rather change (if necessary) the entire image to obtain the desired global saliency goal.

we propose a new quantitative criteria to assess performance of saliency editing algorithms by comparing two properties to previous methods: (i) The ability to manipulate an image such that the saliency map of the result matches the user goal. (ii) The realism of the manipulated image. These properties are evaluated via qualitative means, quantitative measures and user studies. Our experiments show a significant improvement over previous methods. We further show that our general framework is applicable to two other applications: distractor attenuation and background decluttering.

2. Related Work

Attention retargeting methods have a mutual goal – to enhance a selected region. They differ, however, in the way the image is manipulated [15, 25, 27, 30, 31]. We next briefly describe the key ideas behind these methods. A more thorough review and comparison is provided in [24].

Some approaches are based solely on color manipula-

tion [25, 27]. This usually suffices to enhance the object of interest, but often results in non-realistic manipulations, such as purple snakes or blue flamingos. Approaches that integrate also other saliency cues, such as saturation, illumination and sharpness have also been proposed [15, 26, 30, 31]. While attempting to produce realistic and aesthetic results, they do not always succeed, as we show empirically later on.

Recently Yan et al. [34] suggested a deep convolutional network to learn transformations that adjust image aesthetics. One of the effects they study is Foreground Pop-Out, which is similar in spirit to object saliency enhancement. Their method produces aesthetic results, however, it requires intensive manual labeling by professional artists in the training phase and it is limited to the labeled effect used by the professional.

3. Problem Formulation

Our *Object Enhancement* formulation takes as input an image I, a target region mask R and the desired saliency contrast ΔS between the target region and the rest of the image. It generates a manipulated image J whose corresponding saliency map is denoted by S_J.

We pose this task as a patch-based optimization problem over the image J. The objective we define distinguishes between salient and non-salient patches and pushes for manipulation that matches the saliency contrast ΔS. To do this we extract from the input image I two databases of patches of size $w \times w$: $\mathcal{D}^+ = \{p; S_I(p) \geq \tau^+\}$ of patches p with *high* saliency and $\mathcal{D}^- = \{p; S_I(p) \leq \tau^-\}$ of patches p with *low* saliency. The thresholds τ^+ and τ^- are found via our optimization (explained below).

To increase the saliency of patches $\in R$ and decrease the saliency of patches $\notin R$ we define the following energy function:

$$
\begin{aligned}
E(J, \mathcal{D}^+, \mathcal{D}^-) &= E^+ + E^- + \lambda \cdot E^\nabla & (1) \\
E^+(J, \mathcal{D}^+) &= \sum_{q \in R} \min_{p \in \mathcal{D}^+} D(q, p) \\
E^-(J, \mathcal{D}^-) &= \sum_{q \notin R} \min_{p \in \mathcal{D}^-} D(q, p) \\
E^\nabla(J, I) &= \|\nabla J - \nabla I\|_2
\end{aligned}
$$

where $D(q, p)$ is the sum of squared distances (SSD) over $\{L, a, b\}$ color channels between patches q and p. The role of the third term, E^∇, is to preserve the gradients of the original image I. The balance between the color channels and the gradient channels is controlled by λ.

Recall, that our goal in minimizing (1) is to generate an image J with saliency map S_J, such that the contrast in saliency between R and the rest of the image is ΔS. The key to this lies in the construction of the patch sets $\mathcal{D}^+$ and

Algorithm 1 Saliency Manipulation

1: **Input**: Image I; object mask R; saliency contrast ΔS.
2: **Output**: Manipulated image J.

3: Initialize τ^+, τ^- and $J = I$.
4: **while** $\|\psi(S_J, R) - \Delta S\| > \epsilon$ * **do**
5:     1. **Database Update**
6:         $\rightarrow$ Increase τ^+ and decrease τ^-.
7:     2. **Image Update**
8:         $\rightarrow$ Minimize (1) w.r.t. J, holding $\mathcal{D}^+, \mathcal{D}^-$ fixed.
9: **end while**
10: **Fine-scale Refinement**
    * the iterations also stopped when the τ^+ and τ^- stop changing between subsequent iterations.

$\mathcal{D}^-$. The higher the threshold τ^+ the more salient will be the patches in $\mathcal{D}^+$ and in return those in R. Similarly, the lower the threshold τ^- the less salient will be the patches in $\mathcal{D}^-$ and in return those outside of R. Our algorithm performs an approximate greedy search over the thresholds to determine their values.

To formulate mathematically the affect of the user control parameter ΔS we further define a function $\psi(S_J, R)$ that computes the saliency difference between pixels in the target region R and those outside it:

$$\psi(S_J, R) = \operatorname*{mean}_{\beta_{top}}\{S_J \in R\} - \operatorname*{mean}_{\beta_{top}}\{S_J \notin R\} \quad (2)$$

and seek to minimize the saliency-based energy term:

$$E^{sal} = \|\psi(S_J, R) - \Delta S\| \quad (3)$$

For robustness to outliers we only consider the β_{top} (= 20%) most salient pixels in R and outside R in the mean calculation.

4. Algorithm Overview

The optimization problem in (1) is non-convex with respect to the databases $\mathcal{D}^+$, $\mathcal{D}^-$. To solve it, we perform an approximate greedy search over the thresholds τ^+, τ^- to determine their values. Given a choice of threshold values, we construct the corresponding databases and then minimize the objective in (1) w.r.t. J, while keeping the databases fixed. Pseudo-code is provided in Algorithm 1.

Image Update: Manipulate J to enhance the region R. Patches $\in R$ are replaced with similar ones from $\mathcal{D}^+$, while, patches $\notin R$ are replaced with similar ones from $\mathcal{D}^-$.

Database Update: Reassign the patches from the input image I into two databases, $\mathcal{D}^+$ and $\mathcal{D}^-$, of salient and non-salient patches, respectively. The databases are updated at every iteration by shifting the thresholds τ^+, τ^-, in order to find values that yield the desired foreground enhancement and background demotion effects (according to ΔS).

Fine-scale Refinement: We observed that updating both the image J and the databases $\mathcal{D}^+, \mathcal{D}^-$, at all scales, does not contribute much to the results, as most changes happen already at coarse scales. Similar behavior was observed by [29] in retargeting and by [1] in reshuffling. Hence, the iterations of updating the image and databases are performed only at coarse resolution. After convergence, we continue and apply the Image Update step at finer scales, while the databases are held fixed. Between scales, we down-sample the input image I to be of the same size as J, and then reassign the patches from the scaled I into $\mathcal{D}^+$ and $\mathcal{D}^-$ using the current thresholds.

In our implementation we use a Gaussian pyramid with 0.5 scale gaps, and apply 5-20 iterations, more at coarse scales and less at fine scales. The coarsest scale is set to be 150 pixels width.

5. Detailed Description of the Algorithm

Saliency Model Throughout the algorithm when a saliency map is computed for either I or J we use a modification of [22]. Because we want the saliency map to be as sharp as possible, we use a small patch size of 5×5. In addition, we omit the center prior which assumes higher saliency for patches at the center of the image. We found it to ambiguate the differences in saliency between patches, which might be good when comparing prediction results to smoothed ground-truth maps, but not for our purposes. We selected the saliency estimation of [22] since its core is to find what makes a patch distinct. It assigns a score$\in [0, 1]$ to each patch based on the inner statistics of the patches in the image, which is a beneficial property to our method.

Image Update In this step we minimize (1) with respect to J, while holding the databases fixed. This resembles the optimization proposed by [9] for image synthesis. It differs, however, in two important ways. First, [9] consider only luminance gradients, while we consider gradients of all three $\{L, a, b\}$ color channels. This improves the smoothness of the color manipulation, preventing generation of spurious color edges, like those evident in Figure 2c. It guides the optimization to abide to the color gradients of the original image and often leads to improved results (Figure 2d).

As was shown in [9], the energy terms in (1) can be optimized by combining a patch *search-and-vote* scheme and a discrete Screened Poisson equation that was originally suggested by [4] for gradient domain problems. At each scale, every iteration starts with a *search-and-vote* scheme that replaces patches of color with similar ones from the appropriate patch database. For each patch $q \in J$ we search for the Nearest Neighbor patch p. Note, that we perform two separate searches, for the target region in $\mathcal{D}^+$ and for the background in $\mathcal{D}^-$. This is the second difference from [9] where a single search is performed over one source region.

(a) Input image I (b) Mask R

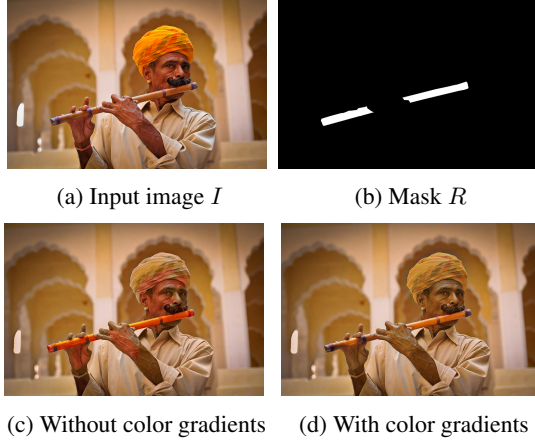(c) Without color gradients   (d) With color gradients

Figure 2: **Chromatic gradients.** A demonstration of the importance of chromatic gradients. (c) When *not* using color gradients - artifacts appear: orange regions on the flutist' hat, hands and face. (d) By solving the screened Poisson equation on all three channels we improve the smoothness of the color manipulation, stopping it from generating spurious color edges, and the color of the flute is more natural looking.

To reduce computation time the databases are represented as two images: $I_{\mathcal{D}^+} = I \cap (S_I \geq \tau^+)$ and $I_{\mathcal{D}^-} = I \cap (S_I \leq \tau^-)$. The search is performed using PatchMatch [1] with patch size 7×7 and translation transformation only (we found that rotation and scale were not beneficial). In the *vote* step, every target pixel is assigned the mean color of all the patches that overlap with it. The voted color image is then combined with the original gradients of image I using a Screened Poisson solver to obtain the final colors of that iteration. We fixed $\lambda = 5$ as the gradients weight.

Having constructed a new image J, we compute its saliency map S_J to be used in the database update step explained next.

Database Update The purpose of the database update step is to search for the appropriate thresholds that split the patches of I into salient $\mathcal{D}^+$ and non-salient $\mathcal{D}^-$ databases. Our underlying assumption is that there exist threshold values that result in minimizing the objective E^{sal} of (3).

Recall that the databases are constructed using two thresholds on the saliency map S_I such that $\mathcal{D}^+ = \{p; S_I(p) \geq \tau^+\}$ and $\mathcal{D}^- = \{p; S_I(p) \leq \tau^-\}$. An exhaustive search over all possible threshold values is non-tractable. Instead, we perform an approximate search that starts from a low value for τ^+ and a high value for τ^- and then gradually increases the first and reduces the second until satisfactory values are found. Note, that $\mathcal{D}^+$ and $\mathcal{D}^-$ could be overlapping if $\tau^+ < \tau^-$.

The naive thresholds $\tau^+ \approx 1, \tau^- \approx 0$, would leave only

the most salient patches in $\mathcal{D}^+$ and the most non-salient in $\mathcal{D}^-$. This, however, could lead to non-realistic results and might not match the user's input for a specific saliency contrast ΔS. To find a solution which considers realism and the user's input we seek the maximal τ^- and minimal τ^+ that minimize the saliency term E^{sal}.

At each iteration we continue the search over the thresholds by gradually updating them:

$$\tau_{n+1}^+ = \tau_n^+ + \eta \cdot \|\psi(S_J, R) - \Delta S\| \qquad (4)$$
$$\tau_{n+1}^- = \tau_n^- - \eta \cdot \|\psi(S_J, \overline{R}) - \Delta S\| \qquad (5)$$

where $\overline{R}$ is the inverse of the target region R. Since the values of the thresholds are not bounded, we trim them to be in the range of $[0, 1]$. Convergence is declared when $E^{sal} = \|\psi - \Delta S\| < \epsilon$, i.e., when the desired contrast is reached. If convergence fails the iterations are stopped when the thresholds stop changing between subsequent iterations. In our implementation $\eta = 0.1$ and $\epsilon = 0.05$.

An important property of our method is that if $\tau^- = 1$ (or very high) and $\tau^+ = 0$ (or very low) the image would be left unchanged as the solution where all patches are replaced by themselves will lead to a zero error of our objective energy function (1).

Robustness to parameters The only parameter we request the user to provide is ΔS which determines the enhancement level. We argue that this parameter is easy and intuitive to tune as it directly relates to the desired saliency contrast between the target region and the background. We used a default value of $\Delta S = 0.6$, for which convergence was achieved for 95% of the images. In only a few cases the result was not aesthetically pleasing and we used other values in the range $[0.4, 0.8]$. Throughout the paper, if not mentioned otherwise, $\Delta S = 0.6$.

An additional parameter is λ, which was fixed to $\lambda = 5$ in our implementation. In practice, we found that for any value $\lambda > 1$ we got approximately the same results, while for $\lambda < 1$ the manipulated images tend to be blurry (mathematical analysis can be found in [4], since our λ is equivalent to that of the screened Poisson).

Convergence and speed Our algorithm is not guaranteed to reach a global minima. However we found that typically the manipulated image is visually plausible, and pertains a good match to the desired saliency.

It takes around 2 minutes to run our algorithm on a 1000×1000 image – the most time demanding step of our method is solving the screened Poisson equation at each iteration. Since our main focus was on quality we did not optimize the implementation for speed. Significant speed-up could be achieved by adopting the method of [12]. As was shown by [9] replacing these fast pyramidal convolutions with our current solver, will reduce run-time from minutes to several seconds.

6. Empirical Evaluation

To evaluate object enhancement one must consider two properties of the manipulated image: (i) the similarity of its saliency map to the user-provided target, and, (ii) whether it looks realistic. Through these two properties we compare our algorithm to HAG [15], OHA [25], and WSR [31], that were identified as top performers in [24]. [1].

We start by providing a qualitative sense of what our algorithm can achieve in Figure 9. Many more results are provided in the supplementary, and we encourage the reader to view them. Comparing to OHA, it is evident that our results are more realistic. OHA changes the hue of the selected object such that its new color is unique with respect to the color histogram of the rest of the image. This often results in unrealistic colors. The results of WSR and HAG, on the other hand, are typically realistic since their manipulation is restricted not to deviate too much from the original image in order to achieve realistic outcomes. This, however, comes at the expense of often failing to achieve the desired object enhancement altogether.

The ability of our approach to simultaneously reduce and increase saliency of different regions is essential in some cases, e.g. Figure 9, rows 1 and 4. In addition, it is important to note that our manipulation latches onto the internal statistics of the image and emphasizes the objects via a combination of different saliency cues, such as color, saturation and illumination. Examples of these complex effects are presented in Figure 9, rows 2, 6 and 7, respectively.

A new benchmark: To perform quantitative evaluation we built a corpus of 667 images gathered from previous papers on object enhancement and saliency [2, 8, 13, 16, 20, 25] as well as images from MS COCO [19]. Our dataset is the largest ever built and tested for this task and sets a new benchmark in this area. Our dataset, code and results are publicly available [2].

Enhancement evaluation: To measure how successful a manipulated image is, we do the following. We take the user provided mask as the ground-truth saliency map. We then compute the saliency map of the manipulated image and compare it to the ground-truth. To provide a reliable assessment we use five different salient object detection methods: MBS [35], HSL [33], DSR [18], PCA [22] and MDP[17], each based on different principles (patch based, CNN, geodesic distance etc.). The computed saliency maps are compared to the ground-truth using two commonly-used metrics for saliency evaluation: (i) Pearsons-Correlation-Coefficient (CC) which was recommended by [6] as the best option for assessing saliency maps, and, (ii) Weighted F-
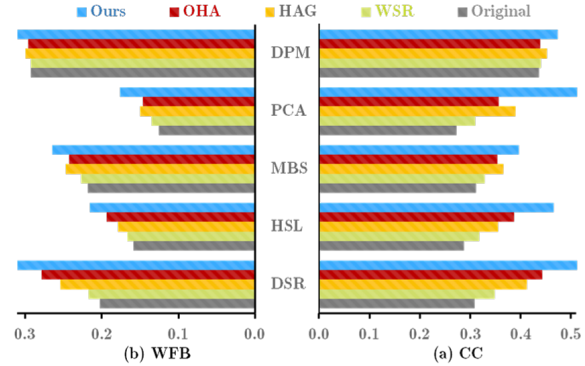
Figure 3: **Enhancement evaluation:** The bars represent the (right) Correlation-Coefficient (CC) and (left) the Weighted F-beta (WFB) [23] scores obtained when comparing the ground-truth masks with saliency maps computed using five different saliency estimation algorithms (see text). The longer the bar, the more similar the saliency maps are to the ground-truth. It can be seen that the saliency maps of our manipulated images are consistently more similar to the ground-truth.
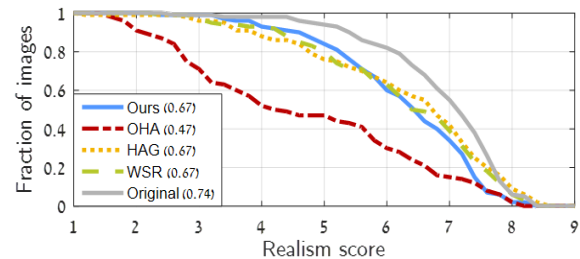

Figure 4: **Realism evaluation.** Realism scores obtained via a user survey (see text for details). The curves show the fraction of images with average score greater than *Realism score*. The Area-Under-Curve (AUC) values are presented in the legend. Our manipulated images are ranked as more realistic than those of OHA and similar to those of WSR and HAG. this is while our enhancement effects are more robust, as shown in Figure 9.

beta (WFB) [23] which was shown to be a preferred choice for evaluation of foreground maps.

The bar plots in Figure 3 show that the saliency maps of our manipulated images are more similar to the ground-truth than those of OHA, WSR and HAG. This is true for both saliency measures and for all five methods for saliency estimation.

Realism: As mentioned earlier, being able to enhance a region does not suffice. We must also verify that the manipulated images look plausible and realistic. We measure this via a user survey. Each image was presented to human participants who were asked a simple question: "Does the image look realistic?" The scores were given on a scale of [1-9], where 9 is 'definitely realistic' and 1 is 'definitely

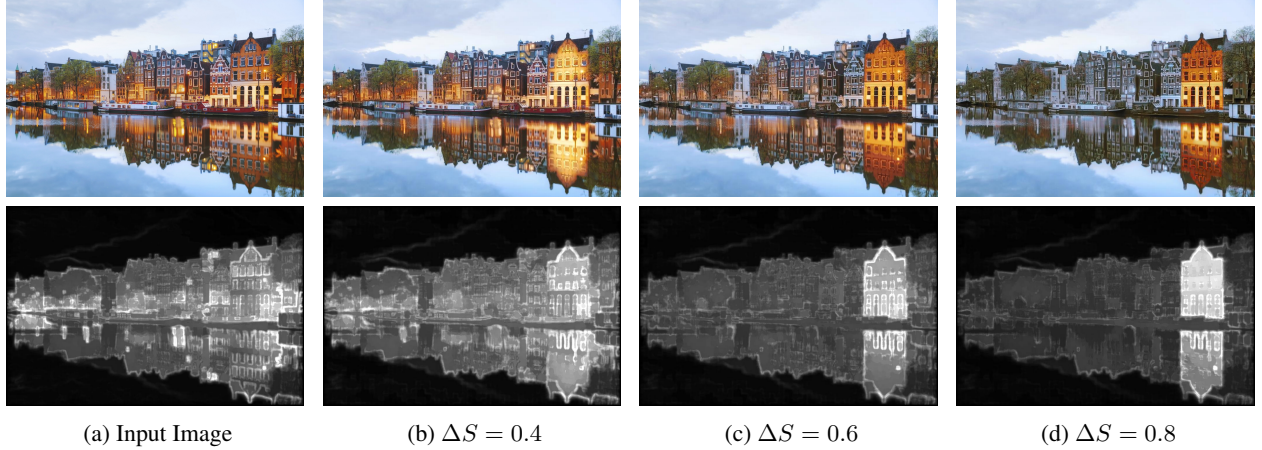|			
(a) Input Image	(b) $\Delta S = 0.4$	(c) $\Delta S = 0.6$	(d) $\Delta S = 0.8$

Figure 5: **Controlling the level of enhancement.** (Top) (a) Input image. (b,c,d) The manipulated image J with $\Delta S = 0.4, 0.6, 0.8$, respectively. (Bottom) the corresponding saliency maps. As ΔS is increased, so does the saliency contrast between the foreground and the background. As mask, the user marked the rightmost house and its reflection on the water.

unrealistic'. We used Amazon Mechanical Turk to collect 20 annotations per image, where each worker viewed only one version of each image out of five. The survey was performed on a random subset of 20% of the data-set.

Figure 4 shows for each enhancement method the fraction of images with average score larger than a realism score $\in [1, 9]$ and the overall AUC values. OHA results are often non-realistic, which is not surprising given their approach uses colors far from those in the original image. Our manipulated images are mostly realistic and similar to WSR and HAG in the level of realism. Recall, that this is achieved while our success in obtaining the desired enhancement effect is much better.

Controlling the Level of Enhancement: One of the advantages of our approach over previous ones is the control we provide the user over the degree of the manipulation effect. Our algorithm accepts a single parameter from the user, ΔS, which determines the level of enhancement. The higher ΔS is, the more salient will the region of interest become, since our algorithm minimizes E^{sal}, i.e., it aims to achieve $\psi(S_J, R) = \Delta S$. While we chose $\Delta S = 0.6$ for most images, another user could prefer other values to get more or less prominent effects. Figure 5 illustrates the influence ΔS on the manipulation results.

The user-provided mask: In our dataset, the mask was marked by users to define a salient object in the scene. In order to use our method on a new image the user is required to mark the region that input region. Note that similarly to other imaging tasks, such as, image completion, compositing, recoloring and warping, the definition of the target region is up to the user to determine and is not part of the method. To facilitate the selection the user can utilize interactive methods such as [21, 28, 32] to easily generate region-of-interest masks.

6.1. Other Applications

Since our framework allows both increasing and decreasing of saliency it enables two additional applications: (i) *Distractor Attenuation*, where the target's saliency is decreased, and (ii) *Background Decluttering*, where the target is unchanged while salient pixels in the background are demoted. A nice property of our approach is that all that is required for these is using a different mask setup, as illustrated in Figure 6.

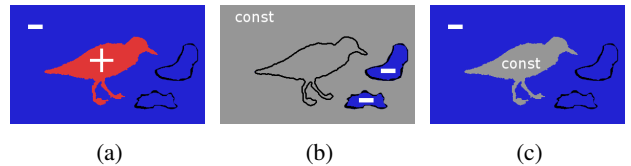|		
(a)	(b)	(c)

Figure 6: **Mask setups.** Illustration of the setups used for: (a) object enhancement, (b) distractor attenuation and (c) decluttering. We increase the saliency in red, decrease it in blue and apply no change in gray.

Distractor Attenuation: The task of getting rid of distractors was recently defined by Fried et al. [13]. Distractors are small localized regions that turned out salient against the photographer's intentions. In [13] distractors were removed entirely from the image and the holes were filled by inpainting. This approach has two main limitations. First, it completely removes objects from the image thus changing the scene in an obtrusive manner that might not be desired by the user. Second, hole-filling methods hallucinate data and sometimes produce weird effects.

Instead, we propose to keep the distractors in the image while reducing their saliency. Figure 7 presents some of our results and comparisons to those obtained by inpainting. We succeed to attenuate the saliency of the distractors, without having to remove them from the image.

(a) Example 1	(b) Example 2	(c) Example 3	(d) Example 4	(e) Inpainting 1	(f) Inpainting 2

Figure 7: **Distractor Attenuation.** (a)-(d) Top: Input images. The distractors were the balloon, the red flag, the shiny lamp and the red roof. Bottom: our manipulated images after reducing the saliency of the distractors. (e)-(f) Top: Zoom in on our result. Bottom: Zoom in on the inpainting result by Adobe Photoshop showing typical artifacts of inpainting methods.

(a) Input image	(b) Manipulated image	(c) Input image	(d) Manipulated image	(e) Masks

Figure 8: **Background DeCluttering.** Often in cluttered scenes one would like to reduce the saliency of background regions to get a less noisy image. In such cases it suffices to loosely mark the foreground region as shown in (e), since the entire background is manipulated. In (a,b) saliency was reduced for the boxes on the left and red sari on the right. In (c,d) the signs in the background were demoted thus drawing attention to the bride and groom.

Background Decluttering: Reducing saliency is also useful for images of cluttered scenes where one's gaze dynamically shifts across the image to spurious salient locations in the background. Some examples of this phenomena and how we attenuate it are presented in Figure 8. This scenario resembles that of removing distractors, with one main difference. Distractors are usually small localized objects, therefore, one could potentially use inpainting to remove them. Differently, when the background is cluttered, marking all the distractors could be tedious and removing them would result in a completely different image.

Our approach easily deals with cluttered background. The user is requested to loosely mark the foreground region. We then leave the foreground unchanged and manipulate only the background, using $\mathcal{D}^-$ to automatically decrease the saliency of clutter pixels. The optimization modifies only background pixels with high saliency, since those with low saliency are represented in $\mathcal{D}^-$ and therefore are matched to themselves.

7. Conclusions and Limitations

We propose a general visual saliency retargeting framework that manipulates an image to achieve a saliency change, while providing the user control over the level of change. Our results outperform the state of the art in object enhancement, while maintaining realistic appearance. Our framework is also applicable to other image editing tasks such as distractors attenuation and background decluttering. Moreover, We establish a benchmark for measuring the effectiveness of algorithms for saliency manipulation.

Our method is not without limitations. First, since we rely on internal patch statistics, and do not augment the patch database with external images, the color transformations are limited to the color set of the image. Second, since our method is not provided with semantic information, in some cases the manipulated image may be non-realistic. For example, in Figure 7, the balloon is colored in gray, which is an unlikely color in that context. Despite its limitations, our technique often produces visually appealing results that adhere to the user's wish.
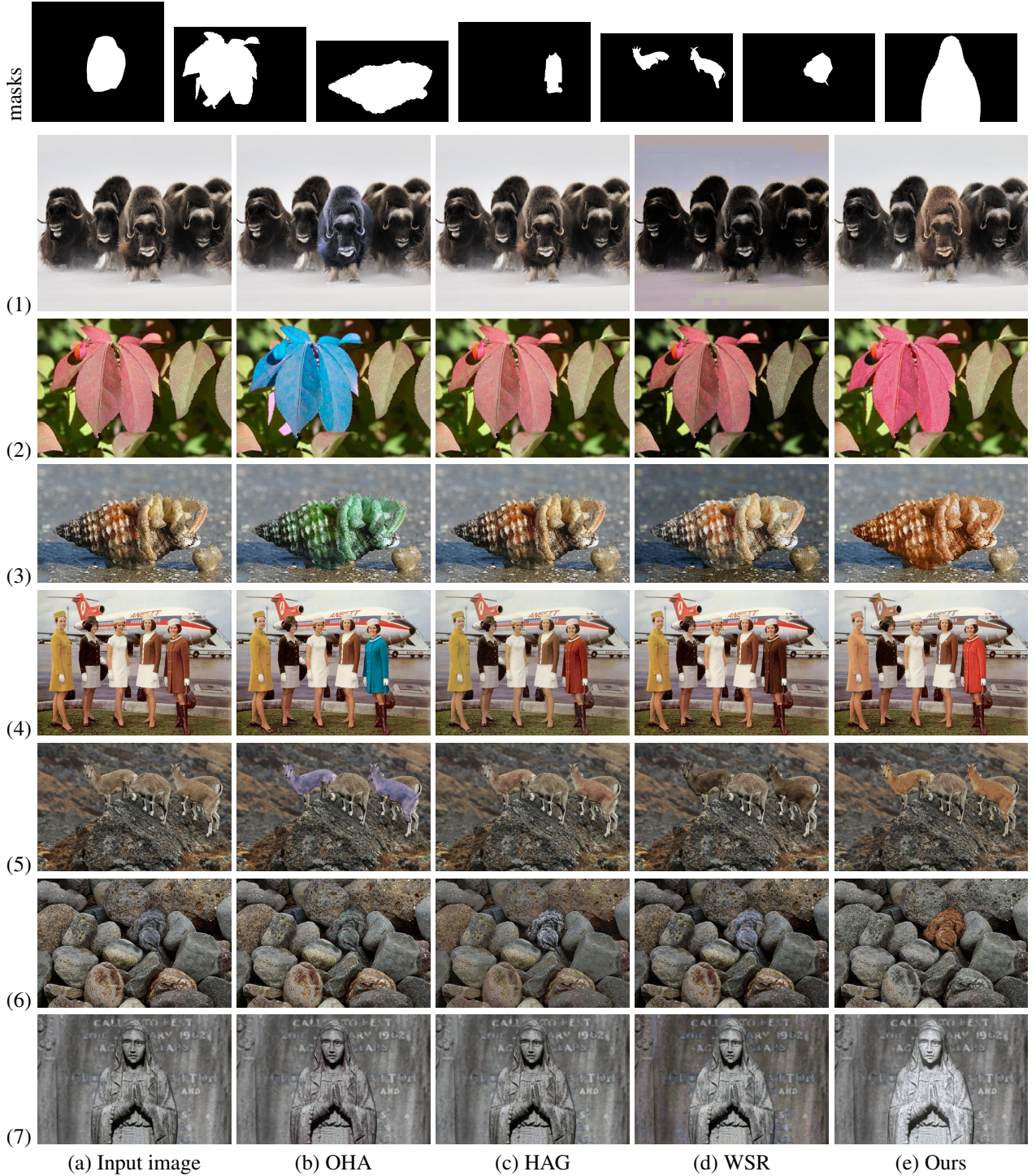
Figure 9: **Object Enhancement** In these examples the user selected a target region to be enhanced (top row). To qualitatively assess the enhancement effect one should compare the input images in (a) to the manipulated images in (b,c,d,e), while considering the input mask (top). The results of OHA in (b) are often non realistic as they use arbitrary colors for enhancement. HAG (c) and WSR (d) produce realistic results, but sometimes (e.g., rows 1,2,6 and 7) they completely fail at enhancing the object and leave the image almost unchanged. Our manipulation, on the other hand, consistently succeeds in enhancement while maintaining realism. Our enhancement combines multiple saliency effects: emphasis by illumination (rows 1 and 7), emphasis by saturation (rows 2, 3 and 4) and emphasis by color (rows 1, 4-7).

References

[1] C. Barnes, E. Shechtman, A. Finkelstein, and D. Goldman. PatchMatch: A randomized correspondence algorithm for structural image editing. *ACM TOG*, 28(3):24, 2009. 2, 3, 4

[2] S. Bell, K. Bala, and N. Snavely. Intrinsic images in the wild. *ACM TOG*, 33(4):159, 2014. 5

[3] M. Bernhard, L. Zhang, and M. Wimmer. Manipulating attention in computer games. In *Ivmsp workshop, 2011 ieee 10th*, pages 153–158, 2011. 1

[4] P. Bhat, B. Curless, M. Cohen, and C. L. Zitnick. Fourier analysis of the 2D screened poisson equation for gradient domain problems. In *ECCV*, pages 114–128. 2008. 3, 4

[5] O. Boiman and M. Irani. Detecting irregularities in images and in video. *IJCV*, 74(1):17–31, 2007. 2

[6] Z. Bylinskii, T. Judd, A. Oliva, A. Torralba, and F. Durand. What do different evaluation metrics tell us about saliency models? *arXiv preprint arXiv:1604.03605*, 2016. 5

[7] M.-M. Cheng, N. J. Mitra, X. Huang, P. H. Torr, and S.-M. Hu. Global contrast based salient region detection. *TPAMI*, 37(3):569–582, 2015. 2

[8] M.-M. Cheng, J. Warrell, W.-Y. Lin, S. Zheng, V. Vineet, and N. Crook. Efficient salient region detection with soft image abstraction. In *ICCV*, pages 1529–1536, 2013. 5

[9] S. Darabi, E. Shechtman, C. Barnes, D. B. Goldman, and P. Sen. Image Melding: Combining inconsistent images using patch-based synthesis. *ACM TOG*, 31(4):82:1–82:10, 2012. 2, 3, 4

[10] T. Dekel, T. Michaeli, M. Irani, and W. T. Freeman. Revealing and modifying non-local variations in a single image. *ACM TOG*, 34(6):227, 2015. 2

[11] A. A. Efros and T. K. Leung. Texture synthesis by non-parametric sampling. In *ICCV*, volume 2, pages 1033–1038, 1999. 2

[12] Z. Farbman, R. Fattal, and D. Lischinski. Convolution pyramids. *ACM TOG*, 30(6):175, 2011. 4

[13] O. Fried, E. Shechtman, D. B. Goldman, and A. Finkelstein. Finding distractors in images. In *CVPR*, pages 1703–1712, 2015. 1, 5, 6

[14] S. Goferman, L. Zelnik-Manor, and A. Tal. Context-aware saliency detection. *TPAMI*, 34(10):1915–1926, 2012. 1

[15] A. Hagiwara, A. Sugimoto, and K. Kawamoto. Saliency-based image editing for guiding visual attention. In *Proceedings of international workshop on pervasive eye tracking & mobile eye-based interaction*, pages 43–48. ACM, 2011. 1, 2, 5

[16] T. Judd, K. Ehinger, F. Durand, and A. Torralba. Learning to predict where humans look. In *ICCV*, 2009. 5

[17] G. Li and Y. Yu. Visual saliency based on multiscale deep features. In *CVPR*, June 2015. 5

[18] X. Li, H. Lu, L. Zhang, X. Ruan, and M.-H. Yang. Saliency detection via dense and sparse reconstruction. In *ICCV*, pages 2976–2983, 2013. 1, 5

[19] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755. 2014. 5

[20] H. Liu and I. Heynderickx. Tud image quality database: Eye-tracking release 1, 2010. 5

[21] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, pages 3431–3440, 2015. 6

[22] R. Margolin, A. Tal, and L. Zelnik-Manor. What makes a patch distinct? In *CVPR*, pages 1139–1146, 2013. 1, 3, 5

[23] R. Margolin, L. Zelnik-Manor, and A. Tal. How to evaluate foreground maps? In *CVPR*, pages 248–255, 2014. 5

[24] V. A. Mateescu and I. Bajić. Visual attention retargeting. *IEEE MultiMedia*, 23(1):82–91, 2016. 1, 2, 5

[25] V. A. Mateescu and I. V. Bajić. Attention retargeting by color manipulation in images. In *Proceedings of the 1st International Workshop on Perception Inspired Video Processing*, pages 15–20. ACM, 2014. 1, 2, 5

[26] E. Mendez, S. Feiner, and D. Schmalstieg. Focus and context in mixed reality by modulating first order salient features. In *International Symposium on Smart Graphics*, pages 232–243, 2010. 1, 2

[27] T. V. Nguyen, B. Ni, H. Liu, W. Xia, J. Luo, M. Kankanhalli, and S. Yan. Image Re-attentionizing. *IEEE Transactions on Multimedia*, 15(8):1910–1919, 2013. 1, 2

[28] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. In *ACM TOG*, volume 23, pages 309–314, 2004. 6

[29] D. Simakov, Y. Caspi, E. Shechtman, and M. Irani. Summarizing visual data using bidirectional similarity. In *CVPR*, pages 1–8, 2008. 2, 3

[30] S. L. Su, F. Durand, and M. Agrawala. De-emphasis of distracting image regions using texture power maps. In *Proceedings of IEEE International Workshop on Texture Analysis and Synthesis*, pages 119–124. ACM, October 2005. 1, 2

[31] L.-K. Wong and K.-L. Low. Saliency retargeting: An approach to enhance image aesthetics. In *WACV*, pages 73–80, 2011. 1, 2, 5

[32] N. Xu, B. Price, S. Cohen, J. Yang, and T. S. Huang. Deep interactive object selection. In *CVPR*, pages 373–381, 2016. 6

[33] Q. Yan, L. Xu, J. Shi, and J. Jia. Hierarchical saliency detection. In *CVPR*, pages 1155–1162, 2013. 1, 5

[34] Z. Yan, H. Zhang, B. Wang, S. Paris, and Y. Yu. Automatic photo adjustment using deep neural networks. *ACM TOG*, 2015. 1, 2

[35] J. Zhang, S. Sclaroff, Z. Lin, X. Shen, B. Price, and R. Mech. Minimum barrier salient object detection at 80 FPS. In *ICCV*, pages 1404–1412, 2015. 1, 5