Distinguishing Local and Global Edits for Their Simultaneous Propagation in a Uniform Framework

Wencheng Wang, *Member, IEEE*, Panpan Xu, Ying Song, Miao Hua, Minying Zhang, and Xiaohui Bie

***Abstract*— In propagating edits for image editing, some edits are intended to affect limited local regions, while others act globally over the entire image. However, the ambiguity problem in propagating edits is not adequately addressed in existing methods. Thus, tedious user input requirements remain since the user must densely or repeatedly input control samples to suppress ambiguity. In this paper, we address this challenge to propagate edits suitably by marking edits for local or global propagation and determining their reasonable propagation scopes automatically. Thus, our approach avoids propagation conflicts, effectively resolving the ambiguity problem. With the reduction of ambiguity, our method allows fewer and less-precise control samples than existing methods. Furthermore, we provide a uniform framework to propagate local and global edits simultaneously, helping the user to quickly obtain the intended results with reduced labor. With our unified framework, the potentially ambiguous interaction between local and global edits (evident in existing methods that propagate these two edit types in series) is resolved. We experimentally demonstrate the effectiveness of our method compared with existing methods.**

***Index Terms*— Edit propagation, image editing, unambiguity.**

I. Introduction

EDIT propagation is a foremost topic in image editing. It entails setting control samples in the image to define the edits then propagating these edits locally or globally in the image by measuring the affinity between pixels. In local propagation methods [1]–[3], the effects of the control samples are propagated locally in limited regions. Although this is sufficient when modifying individual regions, many samples need to be set when modifying many disconnected regions, thereby degrading editing efficiency. In global propagation methods [4]–[6], the edits are propagated without constraint such that similar regions are adjusted using common samples, leading to high editing efficiency. However, unsuitable color blending may be introduced by incompatible influences from different edits. Thus, repeated or dense user-input strokes are required for suitable edit propagation. Considering this, Xu et al. [7] investigated user input density and distribution in different regions, and proposed a method to reduce the ambiguity for pixels distant from or in between the samples. They used a sparse control model, which relates each pixel to only one of the control samples. This differs from global methods where each pixel is influenced by several samples (the cause of color blending). This method contributed significant progress on the topic, but may still cause ambiguity because the pixels of a meaningful region (e.g., an object) may be influenced by different samples, as shown in Fig. 1. This is because the probability computation to obtain the most confident sample for each pixel is highly dependent on the distances from the pixel to the samples. As a result, to obtain a suitable editing result, the user must either set many samples or carefully decide where to set the samples. This is especially crucial for limiting the propagation scope of some control samples, as illustrated in Fig. 7, where it is shown an example of the editing task that is difficult to complete properly even with many strokes provided. In sum, all of these obstacles are the result of edit propagation ambiguity. As a result, the user has to input control samples tediously.

As discussed in the above paragraph, the lack of well-determined edit propagation scope leads to ambiguity. In this paper, we address this challenge by effectively determining the reasonable propagation scopes of the edits. For this, we mark edits for local or global propagation to treat them separately. Thus, the edits are propagated in their own regions, avoiding conflicts. In particular, the global edits are not propagated over the entire image as in existing methods, but rather in the regions outside of the scopes of local edits. With this scheme, the user's desired edits are achieved more effectively because there are no conflicts between different edits. Moreover, this method increases efficiency by only requiring the user to set control samples inside the regions of interest. This differs from existing methods whereby users must determine the ideal locations for control samples or input many control samples for adequate propagation. Because we separate the propagation scopes for edits, we are able to propose a uniform framework allowing the user to deploy local edits and global edits simultaneously. This is superior

Manuscript received August 26, 2014; revised January 9, 2015 and March 12, 2015; accepted April 2, 2015. Date of publication April 10, 2015; date of current version May 5, 2015. This work was supported in part by the National Natural Science Foundation of China under Grant 61379087, in part by the Knowledge Innovation Program through the Chinese Academy of Sciences, and in part by the European Unions Seventh Framework Programme (FP7/2007-2013) under Grant 612627. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Richard J. Radke. *(Corresponding author: W. Wang.)*

The authors are with the State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing 100190, China (e-mail: whn@ios.ac.cn; xupp@ios.ac.cn; songy@ios.ac.cn; huam@ios.ac.cn; zhangmy@ios.ac.cn; xiaohui@ios.ac.cn).

The material includes the results of the user study on editing efficiency by the authors of the paper and the method described in [7]. The total size of the file is 551 kB. Contact whn@ios.ac.cn for further questions or comments.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2015.2421442

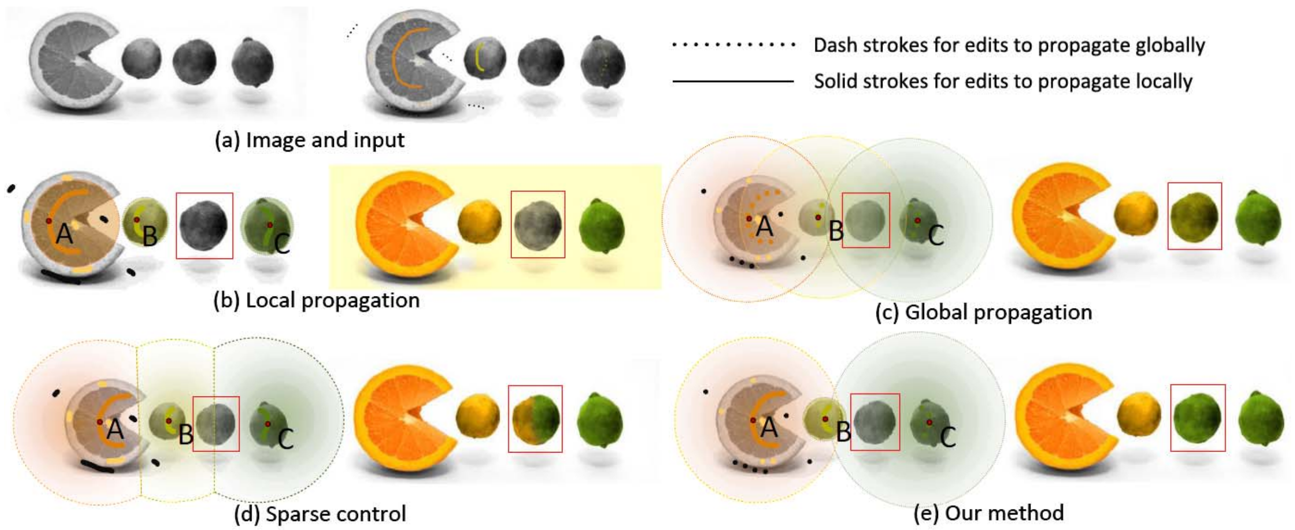


Fig. 1. Comparison between our method and existing methods: local propagation [1], global propagation [4], and the sparse control model [7]. The provided drawn strokes directing the edits are the same for all these methods but have different meanings (i.e., there are no global edits in local propagation and no local edits in global propagation). Their propagation scopes are shown with their corresponding edited results. (a) Original image with the strokes showing the input for the editing schemes. (b) Result of local propagation [1], where the orange color is propagated outside the first fruit to the background, while the third fruit does not change its color according to the edits on the fourth fruit. (c) Result of global propagation [4], where the third fruit from the left does not take on the intended color because of color blending from multiple edit strokes. (d) Result of the sparse control model [7], where the third fruit from the left has different colors on the left and right because of the distance-dependent probability computation. (e) Result of our proposed method, where local edits and global edits are propagated suitably in their own regions, and the third fruit from the left takes on the color of the fourth one because a global edit was specified. Noteworthy comparisons are indicated by red boxes.

to existing methods that perform local edits and global edits in sequence, where the propagation scopes for local edits and global edits may overlap to cause ambiguity, and the user must resolve it by determining the suitable order of local edits versus global edits. Furthermore, existing methods may display undesired modifications caused by unsuitable convergence of the energy function at a local optimization point for global edit propagation. This may hinder realization of specific effects in the region of interest. In Section V, these comparisons are further discussed in detail. In implementing our method, we first use the edge-aware image filter [8] to determine the propagation scopes of the edits. Then, we propagate local and global edits simultaneously in our uniform framework. Our method is more effective and efficient than existing methods in performing edit propagation to produce results corresponding to the user's intentions. Its effectiveness is supported by the experimental results shown in Section V.

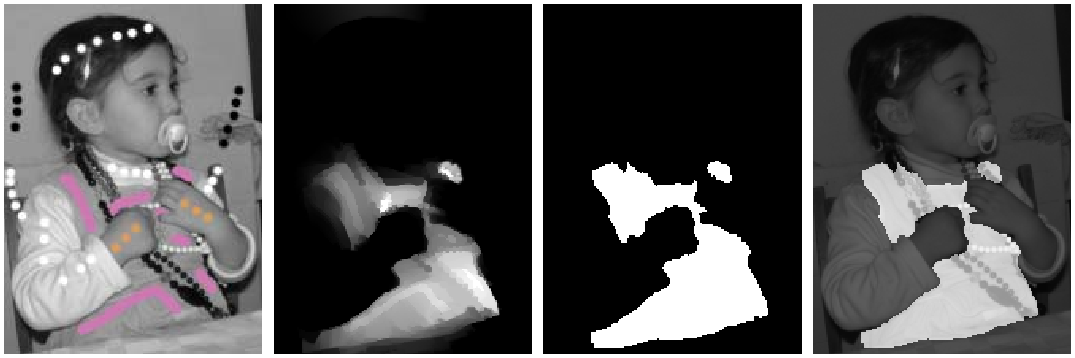
II. Related Work

Edit propagation is a popular approach for editing image appearance. It is intuitive, and allows interactive specification of granular adjustments by applying specified edits to spatially-proximate regions of similar appearance [4]. According to the propagation scopes of edits, existing methods can be classified into two categories: local propagation methods [1]–[3] for propagating edits in a limited region, and global propagation methods [4]–[6] for propagating edits over the entire image.

The edit propagation approach was first introduced by Levin et al. [1] for colorization, and later extended by Lischinski et al. [2] for image tonal manipulations. It constructs a sparse affinity matrix for neighboring pixel pairs and propagates edits according to the affinity between neighboring pixels. This involves solving a sparse linear system for optimization based on the observation that the affinity between two distant pixels is very low and even zero in many cases. This strategy leads to edits generally propagating in their respectively limited scopes. As a result, these methods require many user-input strokes when there are many regions to edit. This is very tedious, even though Li et al. [3] alleviate this problem via pixel clustering.

To efficiently propagate edits over disconnected regions, An and Pellacini [4] generalized the optimization process. They proposed a new energy function accounting for the interactions between all pairs of pixels in the image and represent this in an affinity matrix. Thus, it requires solving a linear system defined by a dense matrix. To simplify the computation, they proposed a low rank approximation of the matrix to solve the system using the inherent structure of the matrix. However, local adjustments lead to a high rank affinity matrix, which cannot be approximated with a low rank one, severely limiting the effectiveness of this approach.

Subsequently, many techniques were proposed to improve the method. Bie et al. [5], Xu et al. [9], and Xiao et al. [10] tried to optimize over pairs of clusters in the feature space, rather than between pairs of pixels, for acceleration. For higher computation and memory efficiency, Li et al. [6] formulated the edit propagation as a Radial Basis Function (RBF) interpolation problem and Chen et al. [11] learned sparse representative samples for edit propagation using the learned samples instead of pixels. Farbman et al. [12] used diffusion distances to accurately measure the affinity between pixels in a feature space. Ma and Xu [13], Yang et al. [14], and Chen et al. [15] enhanced edited images by preserving anti-



(a) (b) (c) (d)

Fig. 2. Propagation scopes of local edits. (a) Input strokes (solid and dashed strokes refer to local and global edits, respectively). (b) Local edit propagation results from applying the image filter in [8]. (c) Consequently obtained segmentation determining the propagation scope of local edits. (d) Result displayed on the image.

aliased edges. Though they improve edit quality and efficiency, such methods are not entirely suitable for local editing because they propagate edits globally (i.e., editing one object may inadvertently change another object of similar appearance).

Because the propagation scope heavily influences the editing result, Baek et al. [16] suggested defining the propagation scopes interactively. This is not suitable for editing large or complicated images because it is tedious for the user. In [7], Xu et al. discussed the problem of propagation scope, and proposed a sparse control model to determine the propagation scope of each control sample according to a probability computation. This can partially solve the ambiguity problem for pixels distant from or in between samples. However, unexpected results will occur if the control samples are not suitably located. For local editing, their method still requires significant user interaction to confine the propagation scope, as discussed in Section I.

As discussed in the above, the ambiguity in propagating edits by existing methods stems from propagation scopes not well defined. Thus, we propose to effectively determine the propagation scopes of edits and propagate them in their own scopes to avoid conflicts. To effectively determine the propagation scopes, many segmentation methods can be used, e.g. the methods based on a learning process [17], [18] or the method using local and nonlocal priors to produce edge-aware mattes [19]. We observe that when the user inputs control samples to edit a region, this region generally has a homogeneous appearance and very distinct from its adjacent regions. Thus, we can employ a simple segmentation method to determine the propagation scopes. Our adoption for this is the edge-aware image filter [8].

III. Suitable Propagation of Edits

When editing an image, the user knows where to perform local edits and which edits should be propagated globally. Such semantic information is essential for suitable image editing but absent from existing methods. In our method, we include such semantic information to determine the propagation scopes of edits. The edits are then propagated within their own scopes thereby avoiding conflicts. This propagation scheme ensures that the resulting edits correspond to the user's original intentions. The key points are effectively determining the propagation scopes for local edits, and designing a new energy function to properly optimize the propagation of local and global edits. These are discussed in the following subsections.

A. Scopes for Local Edits

In [8], it was proposed an energy function to minimize for image processing. It works by enhancing edge information in image processing such that the boundaries of meaningful regions can be highlighted for easy segmentation. Actually, it is by local edit propagation to detect meaningful regions with good boundaries, as illustrated in Fig. 2. Thus, the obtained meaningful regions for local edits are their propagation scopes. Our measure for applying the method in [8] is by firstly assigning 1.0 or -1.0 to the pixels covered by input strokes for local or global editing and then determining the propagation scopes of local edits by the energy function from [8]:

$$E(o) = \sum_{p \in o} E_d(p) + \beta E_g(p) + \gamma\, E_e(p) \tag{1}$$

where p is a pixel of the target image o, E_d is the data cost function, E_g is the gradient cost function, E_e is our edge-aware cost function. The relative contributions of the latter two cost functions are controlled by weights β (typically between 0.5 and 1.5) and γ (typically between 1.0 and 2.0).

These three cost functions are computed as follows.

- $$E_d(p) = [o(p) - \widetilde{l(p)}]^2,$$
  where $\widetilde{l(p)}$ is 1.0 or -1.0 for pixel p covered by an input stroke.
- $E_g(p) = w_x(p)[o_x(p) - t_x(p)]^2 + w_y(p)[o_y(p) - t_y(p)]^2$, where o_x and o_y denote the x and y derivative of o, respectively; t_x and t_y specify the x and y derivative of the input image t, respectively; and w_x and w_y are per-pixel constraint weights computed as follows [20]. $w_x = (\left|\frac{\partial h}{\partial x}(p)\right|^{\alpha} + \varepsilon)^{-1}$ $w_y = (\left|\frac{\partial h}{\partial y}(p)\right|^{\alpha} + \varepsilon)^{-1}$, where h is the log-luminance channel of the input image t, the

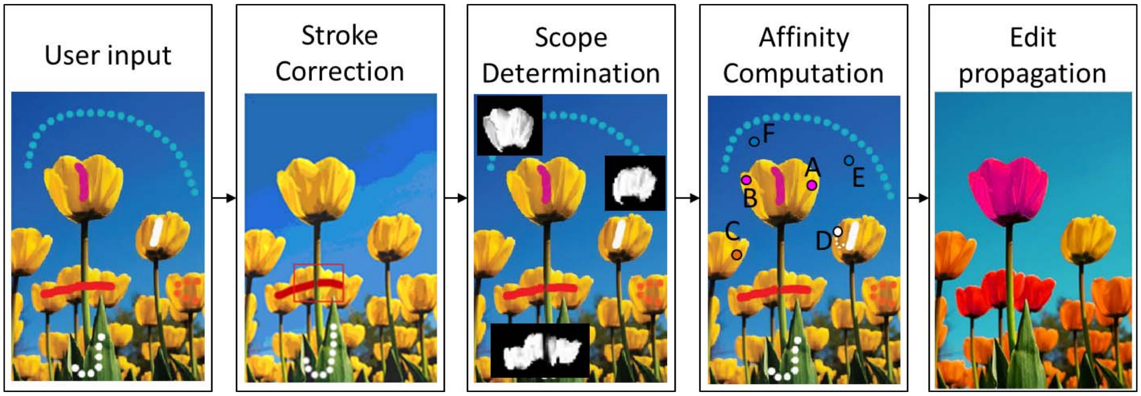


Fig. 3. Our propagation framework for local and global edits (solid and dashed strokes, respectively). In the Stroke Correction step, the red stroke is modified to exclude some pixels for guaranteeing the meaning of editing the yellow tulips, not the stem. In the Scope Determination step, the propagation scopes of edits are determined. In the Affinity Computation step, example pixels A, B, C, D, E, and F show affinity relationships in terms of colors (i.e., similar colors represent closer affinity). The last step (Edit Propagation) shows the final result.

exponent α (typically between 1.2 and 2.0) determines the sensitivity to the gradients of t, and ε is a small constant that prevents division by zero in areas where t is constant.

- $E_e(p) = \sum_{q \in N_4(p)} w^e(p,q)[(o(p)-t(p))-(o(q)-t(q))]^2$, where $N_4(p)$ is the set of 4-connected neighbors of pixel p, and $w^e(p,q) = \exp(-\|t(p)-t(q)\|^2/\sigma^2)$ are weights defined as Gaussians of the distance between adjacent pixels in the Lab color space.

B. The New Energy Function

We design our energy function by the one for global edit propagation as follows [4]:

$$\sum_i \sum_j w_j z_{ij}(o_i - l_j)^2 + \lambda \sum_i \sum_j z_{ij}(o_i - o_j)^2 \qquad (2)$$

where z_{ij} is the affinity between pixel i and pixel j, computed as a product of two Gaussians based on the distance between the pixels' spatial locations x_i and x_j and their appearance vectors f_i and f_j:

$$z_{ij} = \exp(-\|f_i - f_j\|^2/\sigma_a)\exp(-\|x_i - x_j\|^2/\sigma_s) \qquad (3)$$

The energy function in Equation (2) is a sum of two terms whose relative contributions are controlled by λ, where λ is set as $\lambda = \sum_i w_i/n$, and n is the number of pixels in the input image, as done in [4]. The first term is for satisfying the user-specified edits l in producing the output o, weighted by a strength w. The second term accounts for smoothing the variation between neighbor pixels when propagating edits.

To ensure that local edits are propagated in their own scopes and global edits are propagated outside the scopes of local edits, we modify z_{ij} to allow propagation over a pair of pixels only when they are within the same propagation scope. Thus, our energy function is designed as follows:

$$\sum_i \sum_s A_{is}(o_i - l_s)^2 + \lambda \sum_i \sum_j B_{ij}(o_i - o_j)^2 \qquad (4)$$

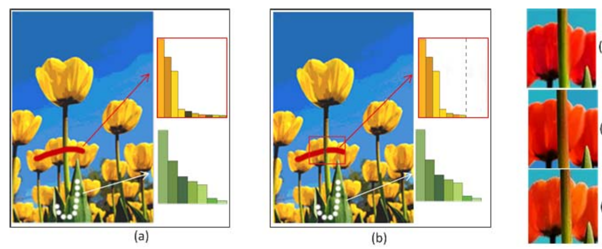


Fig. 4. Correcting strokes. (a) Quantized image with strokes and their corresponding color histograms. (b) Stroke correction result with the updated color histograms, where the pixels of the stem are excluded from the red stroke because their corresponding colors account for a minority group in this stroke and a majority group in the dashed white stroke. (c), (d), and (e) are the propagated results using our method, the method in [1], and the method in [4], respectively. Our method surpasses the methods from [1] and [4] in retaining the stem.

where:

$$\begin{aligned} A_{is} &= z_{is} * T(i,s), \quad \text{and} \\ T(i,s) &= \begin{cases} 1 & if\ pixel\ i\ is\ inside\ the\ propagation \\ & scope\ of\ edit\ s \\ 0 & otherwise, \end{cases} \\ B_{ij} &= z_{ij} * M(i,j), \quad \text{and} \\ M(i,j) &= \begin{cases} 1 & if\ pixels\ i\ and\ j\ are\ within\ the \\ & same\ local\ edit\ scope \\ 1 & if\ pixels\ i\ and\ j\ are\ influenced \\ & by\ global\ edits \\ 0 & others. \end{cases} \end{aligned}$$

l_s refers to the expected edited result at pixel s within a particular edit according to the set of input samples, S. The difference between Equations (2) and (4) is that we remove the weight w because we can always obtain good results by setting $w = 1$ for the pixels covered by inputted control samples and $w = 0$ otherwise.

IV. Unified Framework

Our aim is to provide an efficient image editing approach that allows the user to focus on creativity and productivity without the burden of meticulous stroke placement for

Fig. 5. Comparison of results from our method and the local edit method in [1] and the global edit method in [4]. The undesired changes are indicated by red boxes. Here, dashed and solid lines refer to global and local edits, respectively, when using our method. The two types of strokes and edits are equivalent with the methods [1] and [4]. A white stroke signifies the region to remain unchanged, a black stroke signifies reducing exposure, saturation, or sharpness, and the other colors signify the intended color changes.

inputting precise control samples. To achieve this, we are motivated by the techniques in [21] to extract precise control samples from imprecise inputs. Also, we use clustering techniques to efficiently solve our energy function, as done in [5]. In sum, our framework is composed of the following steps (Fig. 3):

1) The user inputs local and global edits freely by drawing two different types of strokes: solid strokes to represent local edits and dashed strokes to represent global edits.
2) The input strokes are analyzed to obtain precise control samples.
3) The propagation scopes of the edits are determined using the technique discussed in Section III-A.
4) The affinity between pixels and corresponding parameters are computed.
5) The energy function (Equation (4)) is optimized to propagate local and global edits simultaneously.

A. Obtaining Precise Control Samples

In existing stroke-based methods for edit propagation, the inputted strokes are presumed perfectly accurate (i.e., they are assumed to cover only pixels of interest). When this assumption is not valid, the edited results degrade in quality (Fig. 4), even when the constraint is softened as in [4]. Subr et al. [21] and Sener et al. [22] investigated this problem and presented techniques to effectively extract precise control samples from imprecise input. Both techniques are based on [23] and use Markov random fields to estimate the probability of a pixel covered by a stroke being part of the region of interest. However, their computation is complex and time-consuming.

We develop a new measure to extract precise control samples without constructing Markov random fields. We use a simple metric based on our observation that if a color has a small count in one stroke and a large count in another stroke, it is likely that this color is not of interest in the former stroke.

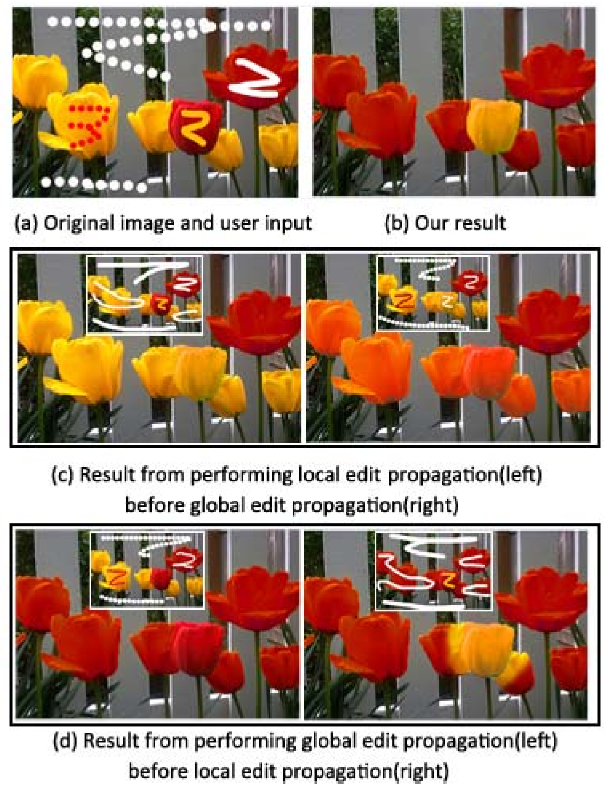
Thus, its related pixels can be removed from the former stroke to accurately represent the user's intent. Our measure thus builds color histograms of strokes and uses them to compute the likelihood of a particular color within a stroke.

Histograms for computing the likelihood for an RGB image generally contain 256^3 colors. Fortunately, motivated by the work in [24], we can quantize the image into 7^3 colors for histogram construction and obtain reliable results.

Our likelihood computation for excluding a color c from a stroke R is as follows:

$$P_{error}(c|\mathcal{R}) = \frac{max\, P(c|\mathcal{W})}{max\, P(c|\mathcal{W}) + P(c|\mathcal{R})} \tag{5}$$

where $P(c|\mathcal{R})$ is the probability of color c in the stroke $\mathcal{R}$ and $P(c|\mathcal{W})$ is the probability of color c in another stroke $\mathcal{W}$ containing many pixels in color c. Empirically, if $P_{error}(c|\mathcal{R})$ is higher than a threshold of 0.7, the pixels in color c should be excluded from stroke $\mathcal{R}$. Determining the stroke that contains the most pixels in color c is theoretically time-consuming. However, because strokes are always very few in number, this does not affect the overall efficiency of the approach. Moreover, we only consider the colors that have a small proportion in a stroke, so the computation is further simplified. For example, correcting the strokes in Fig. 4 takes less than 0.5 seconds whereas the method in [21] takes 2 seconds. As a benefit of this, we can reduce the number of strokes needed in regions with a complicated background by drawing an opposite stroke in the background.



(a) Original image and user input (b) Our result

(c) Result from performing local edit propagation(left) before global edit propagation(right)

(d) Result from performing global edit propagation(left) before local edit propagation(right)

Fig. 6. Our method avoids the ambiguities caused by propagating local and global edits in series. The solid and dashed strokes indicate local and global propagation, respectively. White strokes signify regions that should remain unchanged and other colors indicate the expected color changes. The input strokes are shown in the inlayed white boxes.

B. Cluster-Based Solver

As discussed in [5], pixel clustering can lead to significant efficiency gains in edit propagation. Therefore, we adopt their scheme to develop a cluster-based solver. The scheme consists of the following steps:

1) Pixels are classified into clusters by appearance.
2) The affinity between clusters and the affinity between a cluster and a stroke are computed.
3) Edits are first propagated through clusters (Equation (6)) and then to each cluster's respective member pixels (Equation (7)).

In classifying pixels, we use the k-means clustering technique and accelerate it by ANN searching, just as used in [5]. According to Equation (4), the cluster-based energy function is:

$$\sum_{u}\sum_{s} w_u A_{us}(m_u - l_s)^2 + \lambda \sum_{u}\sum_{v} B_{uv} w_u w_v (m_u - m_v)^2, \tag{6}$$

where m_u represents the u^{th} cluster, w_u denotes the number of pixels in cluster u and acts as a weight, A_{us} is the affinity between cluster u and stroke s, and B_{uv} is the affinity between clusters u and v. The affinity computation is the same as introduced in Section III if every cluster is regarded as a super-pixel, where the location is the center of the cluster, and the color is the mean color of the cluster.

After obtaining the edited results for the clusters, the pixel edits are computed by minimizing the following energy function:

$$\sum_{u\in U}\sum_{s} w_u A_{us}(m_u - l_s)^2 + \lambda \sum_{u\in U}\sum_{j} B_{uj} w_u (m_u - o_j)^2, \tag{7}$$

where B_{uj} is the affinity between pixel j and cluster u. As suggested in [5], it is sufficient to achieve high-quality results with U consisting of three nearest clusters.

V. Results and Discussion

Our method determines the suitable propagation scopes of edits for simultaneous propagation of global and local edits while avoiding conflicts. It allows a potentially flawed user input. Thus, an offhand stroke is sufficient to signal each intended edit type. As a result, we are able to achieve the user's intended edits with more accuracy and less tedious user interaction than existing methods (as illustrated in Fig. 1 and Fig. 7). User experiments demonstrate the effectiveness of our method, and showing that our method is efficient and able to handle medium sized images at interactive rates with low storage requirements. In the following, we provide a detailed comparison of our approach with existing methods.

A. Comparison With the Methods for Local or Global Edits

In Fig. 5, we provide results comparing our method and two existing methods: one for local edits [1] and the other for global edits [4]. Our method is the only one capable

Fig. 7. Our method can use fewer input strokes than required by the method in [7] to obtain the desired results. Using the strokes in (a), we obtain significantly better results through our method (b) than with the method in [7] (c) (where errors are marked by red boxes). When more strokes are added in (d), the results of the method in [7] can be improved (e) but these results are still not ideal.

of obtaining all of the desired results. The inaccurate edits stemming from the methods in [1] and [4] are indicated by red boxes and discussed in the following points.

- For the color editing test, the local method changes the color of the stem of the biggest flower. It also fails to change the colors of some smaller flowers. The global method changes the color of the flowers on the left to that of the middle flower marked to be edited locally, rather than to the color of the right flower marked to be edited globally.
- For the exposure edit test, the intended effect is to decrease the exposure of the background while retaining that of the foreground. The local method cannot influence the regions far from the marked region. The global method changes the foreground and the background simultaneously, and some parts of the background do not show the expected decrease in exposure, because of unsuitable global optimization over the entire image (i.e., unsuitable convergence at a local optimization point).
- For the saturation and blurring edit test, our method effectively decreases the saturation of the background to emphasize the girl, and blurs the background bricks while retaining the sharp focus of the boy. However, these edits cannot be achieved with the local or global methods [1], [4], respectively, for the same reasons as discussed for the exposure example in the second row.

These results show the advantages of our method for edit propagation by distinguishing edits for local or global propagation and constraining their propagation scopes.

B. Comparison With Deploying Local and Global Edits in Succession

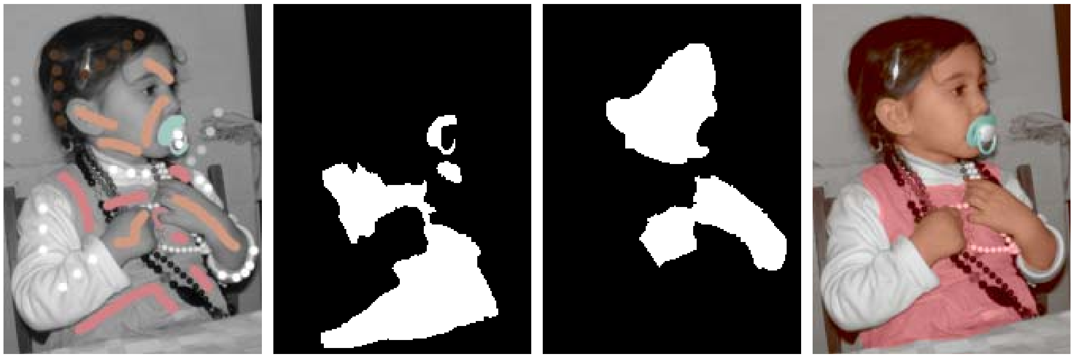
It has been suggested that local and global edits can be applied sequentially to realize the user's intentions. Aside from the inconvenience of waiting for the result of one layer of edits before performing the next edit, this scheme may lead to inaccuracies if the order of local and global edits is incorrect. This stems from the fact that the propagation scopes of global and local edits may overlap and cause ambiguity. As illustrated in Fig. 6, the editing instructions in (a) can be easily realized with our method in (b). With regard to deploying local and global edits in succession, if local edits are performed before global edits, some flowers are erroneously changed to orange (c), because the global method changes the result of the local method. Otherwise, color bleeding results from performing global edits before local edits (d). This example demonstrates the necessity of constraining propagation scopes.

C. Comparison With the Sparse Control Model

The drawbacks of performing local and global edits in series are addressed by Xu et al. [7]. However, their approach for propagating local and global edits harmoniously has some limitations as discussed in Section I, and requires many control samples to achieve the desired results. With our method, this requirement is relaxed, only requiring strokes within the regions of interest. As illustrated in Fig. 7, we can use few strokes to obtain edited results, superior to the method in [7]. In some cases, even if the method in [7] uses many input strokes, the desired results still cannot be obtained (e.g., the tomato example shown in Fig. 7). This indicates that the propagation ambiguity of edits cannot be solved entirely by simply adjusting the placement of the control samples. Moreover, precise placement of control samples is itself a tedious task.

D. Efficiency

Our method provides significant savings with respect to tedious user interaction for stroke placement. We achieve efficient performance by integrating some acceleration techniques (see Section IV). We test the efficiency through a user study as described in the Appendix. The resulting efficiency gains over



(a) (b) (c) (d)

Fig. 8. Scopes for different local edits are detected by applying the image filter twice, once for the shirt and pacifier (b), and again for the face and hands (c). The edited result according to the strokes in (a) is shown in (d).

the sparse control method in [7] are a 80.04% reduction in strokes and a 278.5% increase in editing speed. In particular, we can reduce the times of interactions and the time cost for drawing a stroke, meaning the user can use our method to realize his editing desires more easily.

E. Storage

Though our method propagates local and global edits simultaneously, the storage requirement is not high compared with existing methods. This is because our energy function retains the form used for global edit propagation, except that we perform the affinity computation in a different way. For example, in treating the images in Fig. 5, we need 73 MB storage on average, while the method in [5] requires 68 MB storage on average. This is significant because the method in [5] has a relatively low storage requirement among existing methods and was used as the base to develop our clustering techniques.

F. Limitation

Our method is based on segmentation to determine the propagation scopes of edits. When the segmented results are poor, our method cannot determine accurate scopes and thus produces artifacts. In our current implementation, we use an edge-aware image filter for segmentation, which can get the local propagation regions in a pass when they do not neighbor each other. Otherwise, we need treat neighboring local propagation regions gradually in a sequence rather than simultaneously. Of course, once the propagation scopes are determined, we can obtain high-quality edited results as illustrated in Fig. 8.

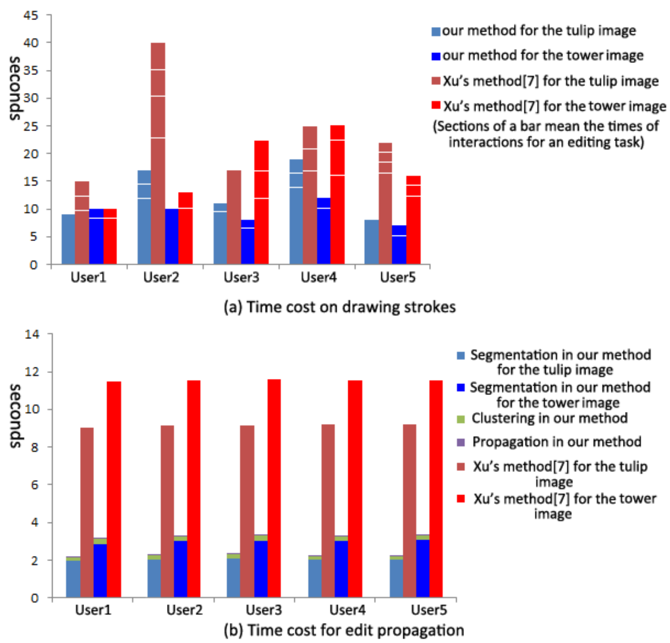


Fig. 9. Performance comparison between our method and the method in [7] to get similar results in the user study. Here, the sections of a bar in (a) mean the times of interactions for an editing task, and they are arranged from bottom to up by the series of interactions to add strokes gradually. Clearly, with our method, the user can save the times of interactions and the time cost for drawing a stroke. As for the efficiency on edit propagation, ours is much faster than the method in [7] by several times (b).

VI. Conclusions

In this paper, we investigate the ambiguity problem for edit propagation. We show that existing methods have shortcomings to obtain accurate, artifact-free results because they do not effectively constrain the propagation scopes of the set control samples. We address this challenge by determining the reasonable propagation scopes for the control samples. Thus, edits are propagated within their respective scopes to avoid conflicts from propagation ambiguity. The advantages of our approach include deploying local and global edits simultaneously in a uniform framework and achieving the user's intended edits with significantly reduced requirements for tedious input adjustments. We also develop and integrate many advanced techniques for acceleration, running our method much faster than existing methods; even allowing interactive image editing. Overall, our method is robust and easy to use. We require less effort from the user in inputting edits, and so increasing the user's focus on creative and productive tasks.

In theory, our method could be extended to apply to videos. However, this is not straightforward because the contents in video frames may change drastically and the appearance of the

(a) (b) (c) (d)

Fig. 10. The editing process and the edit results for a participant in our user study. To get similar results (b) (d), he can input fewer strokes with our method than with the method in [7]. (a) Inputted strokes with our method. (b) Our results. (c) Inputted strokes with the method in [7]. (d) Results from the method in [7].

same content may also change abruptly, making it difficult to track the contents to edit. This is a fruitful subject for future study.

Appendix A
The User Study on Comparing Editing Efficiency

We invited 5 graduate students to edit two images using our method or the method with a sparse control model in [7], both of which are implemented by ourselves. The two images are the tulip image and the tower image in $640 * 480$ and $640 * 640$ pixels respectively.

In Fig. 9, it is listed the statistics on drawing strokes and propagating edits for getting similar results by these two methods, where the used personnel computer is installed with an Intel i7 CPU and 4GB RAM. By the statistics, it is known that we can significantly save the strokes to input and the time cost on drawing strokes, and speed up edit propagation. The reduction ratio on drawing strokes for every participant to perform a editing task is computed in $(time2 - time1)/time1$, where $time1$ and $time2$ are the time cost for the participant to take on drawing strokes for a same editing task in using our method and the method in [7] respectively, and then these ratios are averaged as our obtainment over the method in [7] on drawing strokes. As for the acceleration ratios on propagating edits, they are treated in the same way. The results are given in Section V-D.

In Fig. 10, the images for one participant to draw strokes and his edited results are displayed as an example to show the editing process and the edited results.

Acknowledgment

Thanks are given to the anonymous reviewers for their valuable suggestions, and the participants in the user study.

References

[1] A. Levin, D. Lischinski, and Y. Weiss, "Colorization using optimization," in *Proc. ACM SIGGRAPH*, 2004, pp. 689–694.
[2] D. Lischinski, Z. Farbman, M. Uyttendaele, and R. Szeliski, "Interactive local adjustment of tonal values," in *Proc. ACM SIGGRAPH*, 2006, pp. 646–653.
[3] Y. Li, E. Adelson, and A. Agarwala, "ScribbleBoost: Adding classification to edge-aware interpolation of local image and video adjustments," in *Proc. 19th EGSR*, 2008, pp. 1255–1264.
[4] X. An and F. Pellacini, "AppProp: All-pairs appearance-space edit propagation," in *Proc. ACM SIGGRAPH*, 2008, Art. ID 40.
[5] X. Bie, H. Huang, and W. Wang, "Real time edit propagation by efficient sampling," *Comput. Graph. Forum*, vol. 30, no. 7, pp. 2041–2048, 2011.
[6] Y. Li, T. Ju, and S.-M. Hu, "Instant propagation of sparse edits on images and videos," *Comput. Graph. Forum*, vol. 29, no. 7, pp. 2049–2054, 2010.
[7] L. Xu, Q. Yan, and J. Jia, "A sparse control model for image and video editing," *ACM Trans. Graph.*, vol. 32, no. 6, 2013, Art. ID 197.
[8] M. Hua, X. Bie, M. Zhang, and W. Wang, "Edge-aware gradient domain optimization framework for image filtering by local propagation," in *Proc. IEEE CVPR*, Jun. 2014, pp. 2838–2845.
[9] K. Xu, Y. Li, T. Ju, S.-M. Hu, and T.-Q. Liu, "Efficient affinity-based edit propagation using K-D tree," *ACM Trans. Graph.*, vol. 28, no. 5, 2009, Art. ID 118.
[10] C. Xiao, Y. Nie, and F. Tang, "Efficient edit propagation using hierarchical data structure," *IEEE Trans. Vis. Comput. Graphics*, vol. 17, no. 8, pp. 1135–1147, Aug. 2011.
[11] X. Chen, D. Zou, J. Li, X. Cao, Q. Zhao, and H. Zhang, "Sparse dictionary learning for edit propagation of high-resolution images," in *Proc. IEEE CVPR*, Jun. 2014, pp. 2854–2861.
[12] Z. Farbman, R. Fattal, and D. Lischinski, "Diffusion maps for edge-aware image editing," in *Proc. ACM SIGGRAPH*, 2010, Art. ID 145.
[13] L.-Q. Ma and K. Xu, "Antialiasing recovery for edit propagation," in *Proc. ACM VRCAI*, 2011, pp. 125–130.
[14] L. Yang, P. V. Sander, J. Lawrence, and H. Hoppe, "Antialiasing recovery," *ACM Trans. Graph.*, vol. 30, no. 3, 2011, Art. ID 22.
[15] X. Chen, D. Zou, Q. Zhao, and P. Tan, "Manifold preserving edit propagation," *ACM Trans. Graph.*, vol. 31, no. 6, 2012, Art. ID 132.
[16] J. Baek, D. Pająk, K. Kim, K. Pulli, and M. Levoy, "WYSIWYG computational photography via viewfinder editing," *ACM Trans. Graph.*, vol. 32, no. 6, 2013, Art. ID 198.

[17] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 898–916, May 2011.

[18] J. Carreira and C. Sminchisescu, "Constrained parametric min-cuts for automatic object segmentation," in *Proc. IEEE Conf. CVPR*, Jun. 2010, pp. 3241–3248.

[19] X. Chen, D. Zou, S. Z. Zhou, Q. Zhao, and P. Tan, "Image matting with local and nonlocal smooth priors," in *Proc. IEEE Conf. CVPR*, Jun. 2013, pp. 1902–1907.

[20] Z. Farbman, R. Fattal, D. Lischinski, and R. Szeliski, "Edge-preserving decompositions for multi-scale tone and detail manipulation," in *Proc. ACM SIGGRAPH*, 2008, Art. ID 67.

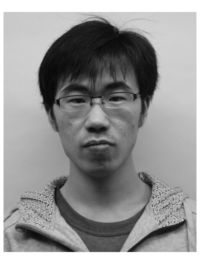
[21] K. Subr, S. Paris, C. Soler, and J. Kautz, "Accurate binary image selection from inaccurate user input," *Comput. Graph. Forum*, vol. 32, no. 2, pp. 41–50, 2013.

[22] O. Sener, K. Ugur, and A. A. Alatan, "Error-tolerant interactive image segmentation using dynamic and iterated graph-cuts," in *Proc. 2nd ACM IMMPD*, 2012, pp. 9–16.

[23] Y. Boykov, O. Veksler, and R. Zabih, "Markov random fields with efficient approximations," in *Proc. 11th IEEE Comput. Soc. Conf. CVPR*, Jun. 1998, pp. 648–655.

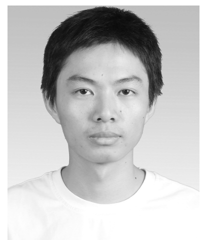
[24] M.-M. Cheng, G.-X. Zhang, N. J. Mitra, X. Huang, and S.-M. Hu, "Global contrast based salient region detection," in *Proc. 24th IEEE Conf. CVPR*, Jun. 2011, pp. 409–416.

Wencheng Wang (M'14) received the Ph.D. degree in software from the Institute of Software, Chinese Academy of Sciences, in 1998. He is currently a Professor with the State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, where he leads a Research Group on Computer Graphics and Image Processing. His research interests include visualization, image editing and virtual reality. He has been a member of the ACM since 2008 and a member of the IEEE since 2014.



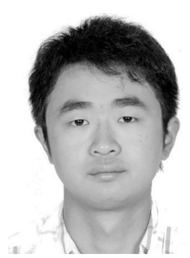
Panpan Xu received the B.E. degree from the Qingdao University of Science and Technology, in 2012. He is currently pursuing the Ph.D. degree with the State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences. His research interests include image editing and computer graphics.

Ying Song received the Ph.D. degree from Zhejiang University, in 2009. She conducted post-doctoral studies with the State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, from 2011 to 2014, where she has served as an Assistant Researcher. Her research has focused on computer graphics and computer vision, especially in the areas of realistic appearance modeling, real time rendering, texture and image analysis, object recognition, and image editing.



Miao Hua received the B.S. degree from the University of Science and Technology of China, in 2009. He is currently pursuing the Ph.D. degree with the State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences. His research interests include image and video processing and computer graphics.

Minying Zhang received the B.E. degree from Jilin University, China, in 2010, and the master's degree from the Institute of Software, Chinese Academy of Sciences, in 2014. His research interests include image and video processing, and computer graphics.



Xiaohui Bie received the bachelor's degree in engineering mechanics from the Huazhong University of Science and Technology, and the Ph.D. degree from the Institute of Software, Chinese Academy of Sciences, in 2013. His main research interests are computer vision and interactive image/video editing.