Semantic colorization with internet images

Conference Paper *in* ACM Transactions on Graphics · December 2011

DOI: 10.1145/2070752.2024190

Shaojie Zhuo
Qualcomm, Canada
19 PUBLICATIONS **603** CITATIONS
SEE PROFILE

Raj Kumar Gupta
Nanyang Technological University
19 PUBLICATIONS **272** CITATIONS
SEE PROFILE

Yu-Wing Tai
91 PUBLICATIONS **2,957** CITATIONS
SEE PROFILE

David Cho
University of Nottingham, Ningbo Campus
71 PUBLICATIONS **1,068** CITATIONS
SEE PROFILE

Semantic Colorization with Internet Images

Alex Yong-Sang Chia[1] Shaojie Zhuo[2] Raj Kumar Gupta[3] Yu-Wing Tai[4]
Siu-Yeung Cho[3] Ping Tan[2] Stephen Lin[5]
[1]Institute for Infocomm Research [2]National University of Singapore [3]Nanyang Technological University
[4]Korea Advanced Institute of Science and Technology [5]Microsoft Research Asia

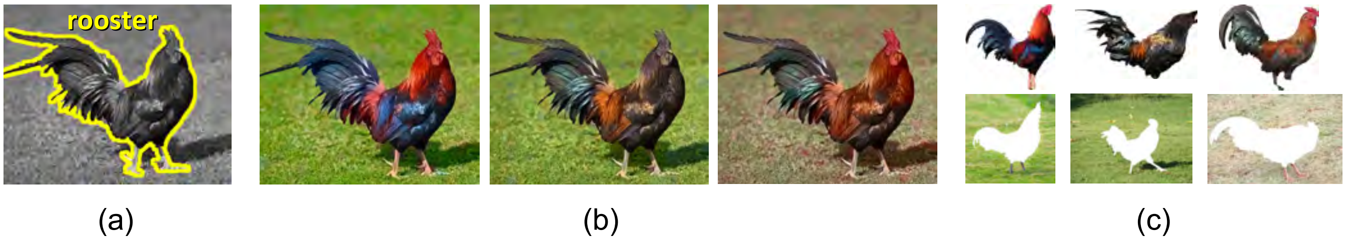


Figure 1: *Our system takes a grayscale photo with labeled and segmented foreground objects (a) as input and generates a set of colorization results (b) using reference image regions automatically searched from the internet and filtered to obtain the most suitable examples. Some of these examples are shown in (c).*

Abstract

Colorization of a grayscale photograph often requires considerable effort from the user, either by placing numerous color scribbles over the image to initialize a color propagation algorithm, or by looking for a suitable reference image from which color information can be transferred. Even with this user supplied data, colorized images may appear unnatural as a result of limited user skill or inaccurate transfer of colors. To address these problems, we propose a colorization system that leverages the rich image content on the internet. As input, the user needs only to provide a semantic text label and segmentation cues for major foreground objects in the scene. With this information, images are downloaded from photo sharing websites and filtered to obtain suitable reference images that are reliable for color transfer to the given grayscale photo. Different image colorizations are generated from the various reference images, and a graphical user interface is provided to easily select the desired result. Our experiments and user study demonstrate the greater effectiveness of this system in comparison to previous techniques.

Links: DL PDF

1 Introduction

Image colorization can bring a grayscale photo to life, but often demands extensive user interaction. In techniques such as [Levin et al. 2004; Huang et al. 2005], a user typically needs to specify many color scribbles on the image to achieve a desirable result. Moreover, it can be difficult for a novice user to provide these color scribbles in a consistent and perceptually coherent manner. Other methods take a different approach by using a color image of a similar scene as a reference, and transferring its colors to the grayscale input image [Reinhard et al. 2001; Welsh et al. 2002]. This requires less skill from the user, but a suitable reference image may take much effort to find. In addition, inaccuracies in color transfer can lead to results that appear unnatural.

To colorize grayscale photos with less manual labor, we present a system that takes advantage of the tremendous amount of image data available on the internet. The internet is almost certain to contain images suitable for colorizing a given grayscale input, but finding those images in a sea of photos is a challenging task, especially since search engines often return images with incompatible content. We address this problem with a novel image filtering method that analyzes spatial distributions of local and regional image features to identify candidate reference regions most compatible with the grayscale target. The user needs only to provide semantic labels and segmentation cues for major foreground objects in the grayscale image, which is more intuitive than previous scribble based user interaction. For each foreground object, a multitude of images is downloaded from the internet using the semantic label as a search term, and our system filters them down to a small number of best matches. To minimize the amount of user input, our system does not require the user to label and segment background regions. Rather, it exploits correlations between the foregrounds and backgrounds of scenes by re-using the images downloaded for the foreground objects, which likely contain some backgrounds that can serve as a reference for background colorization.

From the filtered reference images, the system transfers colors to the corresponding foreground objects and background with a graph-based optimization based on local properties at a super-pixel resolution. Since the filtering method seeks reference objects with spatial distributions of features most consistent with the target object, color transfer becomes more reliable, as corresponding locations between the reference and target can be identified more accurately. Various colorization results are generated from the set of reference images, and the user is provided an intuitive interface to rapidly explore the results and select the most preferred colorization.

2 Related Work

Colorization methods can be roughly divided into those based on user drawn scribbles and those that utilize example images. Scribble based methods propagate the colors from an initial set of user drawn strokes to the whole image. For example, Levin et al. [2004] derived an optimization framework for this propagation to ensure that similar neighboring pixels are assigned a similar color. This method was improved in [Huang et al. 2005] to reduce color blending at image edges. Yatziv et al. [2006] combined the colors of multiple scribbles to colorize a pixel, where the combination weights depend on a distance measure between the pixel and the scribble. Qu et al. [2006] and Luan et al. [2007] both propagated colors according to texture similarity among local areas within an image to reduce the number of scribbles needed.

Instead of relying on user scribbles for color information, example based methods transfer colors automatically from a color reference image of a similar scene. This approach was proposed in [Reinhard et al. 2001; Welsh et al. 2002] and was also demonstrated as an application in [Hertzmann et al. 2001]. Irony et al. [2005] transferred colors only to points of high confidence as a first step, then treated them as user scribble input to [Levin et al. 2004] for colorization. Charpiat et al. [2008] proposed a global optimization method for automatic color assignment. The success of these methods depends heavily on finding a suitable reference image, which can be a time-consuming task. Moreover, correct color assignments can be difficult to infer from these images due to correspondence ambiguities. These issues are avoided in the method of Liu et al. [2008], which colorizes photos of famous landmarks using internet images. For famous landmarks, suitable reference images are easily found on the web, and color assignments for these fixed, rigid objects can be determined by image registration. Our work also takes advantage of internet image search to find appropriate reference images, but addresses the much more challenging problem of colorizing general objects and scenes for which exact matches typically cannot be found.

3 Overview

A block diagram of our system framework is shown in Fig. 2. To colorize a grayscale image, the user first segments the foreground object(s) with the powerful and intuitive Lazy Snapping technique [Li et al. 2004], and provides a semantic text label for each object. These are the only inputs required from the user. We use the semantic labels to download a large set of photos from image sharing websites such as Flickr, Google Image Search and Picasa. To expand the diversity and quantity of downloaded images, we employ Google Translation to translate the text labels into German, Chinese, French, Spanish, Italian and Portuguese, then search with the translated terms as well. Among the downloaded images, we find the most suitable reference photos (for both foreground objects and background) by filtering the search results with respect to similarity to the grayscale input. Each scene object and the background is then colorized using the reference images to obtain a diverse set of natural colorization results. The user can efficiently select from among these results with a provided user interface.

4 Reference Image Selection

Internet image search is far from perfect, and many downloaded images do not contain the desired object. Furthermore, certain images allow for more reliable color assignment because their high similarity to the grayscale target facilitates accurate correspondence. We wish to identify such images and use them as color references. Here, we build on the recent work of [Chen et al. 2009] to find the most appropriate reference images from among the downloaded photos for each foreground object and background.

4.1 Foreground Object Filtering

In selecting reference images for a foreground object, we aim to identify photos in which the foreground object provides a close match to grayscale target object in terms of several appearance properties, since such objects have a high likelihood of being both correct and reliable for color transfer. We first download a set of internet images (around 30K) with the user supplied text label. We next select so-called ‘algorithm-friendly’ internet images (about 10K) using saliency filtering as done in [Chen et al. 2009]. Salient foreground objects are segmented automatically from these images by applying the saliency detector in [Liu et al. 2007] and the Grabcut algorithm [Rother et al. 2004]. We apply contour consistency filtering [Chen et al. 2009] to these segmented objects to find objects with shape context descriptors [Belongie et al. 2002] similar to the input grayscale object. To allow for some shape deformation, we apply an affine transformation to register the internet object with the input object, where the affine matrix is computed from the correspondences of matching shape context descriptors. We sum up the shape context matching cost with that of the affine registration to give an overall shape matching score. This overall score is used to rank the segmented internet objects, and the top 250 objects are retained for further filtering. In Fig. 3(a), we show some internet objects found in this manner for the ‘rooster’ example of Fig. 1(a), sorted according to their shape consistency score.

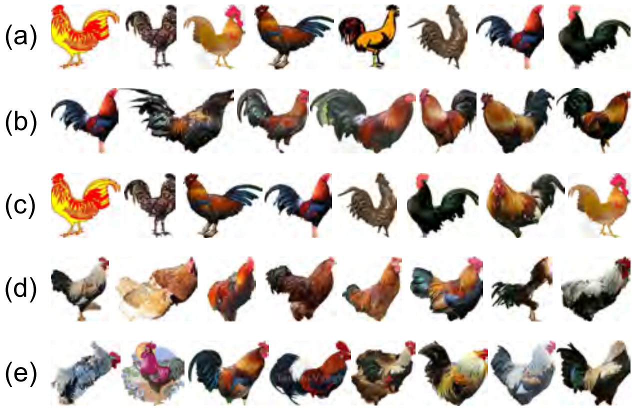


Figure 3: *Reference image selection for foreground objects. Top internet images according to: (a) shape context scores, (b) our method with lowest combined score from Eq. (1), (c) [Chen et al. 2009], (d) [Hays and Efros], and (e) [Zhu et al. 2011].*

While shape information can be a vital cue to filter away irrelevant internet objects, some inappropriate objects are likely to remain, e.g. the top ranked object in Fig. 3(a). In the following, we present a novel filter to improve image selection based on local information (e.g. intensity, texture and SIFT features), which leads to reference images more suitable than those obtained by [Chen et al. 2009].

Intensity We describe each internet object by a 3D intensity histogram spanned by the dimensions of row, column and intensity. The row and column dimensions of this representation encode the spatial distribution of the intensities. All internet objects are registered to the input grayscale image by an affine transformation before building their histograms. To avoid boundary artifacts, each contribution into a histogram bin is weighted by $1 - d$ for each dimension, where d is the distance of the pixel to the bin center measured in terms of bin spacing. We compute the distance between two histograms A, B by the χ^2 distance. Our implementation employs ten bins for the row and column dimensions and 64 bins for the intensity dimension.

Texture In addition to intensity, we also exploit texture features to evaluate object similarity. For each of the retained internet objects, we apply Gabor filters with eight orientations varying in increments of $\pi/8$ from 0 to $7\pi/8$, and five exponential scales $\exp(i \times \pi), i = 0, 1, 2, 3, 4$ to compute a 40-dimensional texture

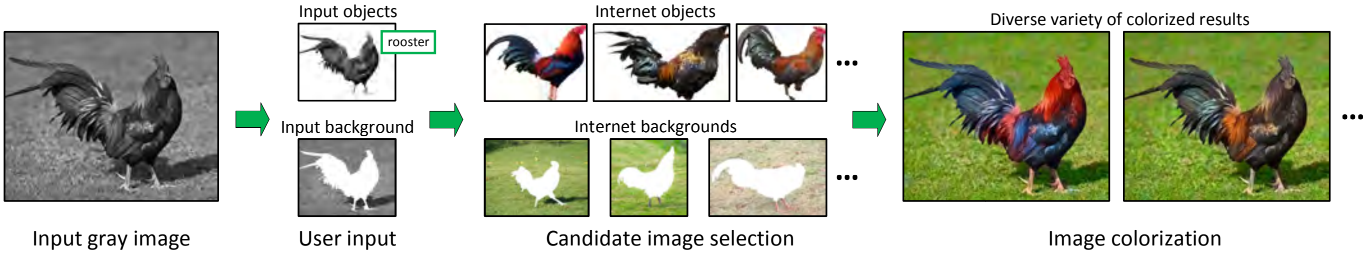


Figure 2: *System pipeline. Foreground objects are labeled and segmented by the user. Our system searches the internet to find relevant reference images for colorization, and utilizes these images to colorize the grayscale input image.*

feature at each pixel. These texture features are then grouped by k-means clustering, with k set to 64 in our work. The cluster centers are taken as texton codewords, and each pixel is associated with the codeword with the smallest Euclidean distance to its texture feature. Based on the image position and texton codeword index of each pixel, we build a 3D texture histogram for each object and use the χ^2 distance to measure the distance between two histograms.

Dense SIFT We adopt dense SIFT features [Lowe 1999] in a manner similar to that for texture. Specifically, we extract SIFT features at the finest scale, and cluster the SIFT features from different pixels to a set of (64) codewords, which are used together with pixel position to construct a 3D histogram. The χ^2 distance is used to measure the distance between histograms.

Combined similarity metric The intensity, texture and dense SIFT matching scores are then linearly combined as

$$D(A,B) = \omega_i D_i(A,B) + \omega_t D_t(A,B) + \omega_s D_s(A,B) \qquad (1)$$

where A, B denote an input object and an internet object respectively. We denote $D_i(\cdot,\cdot), D_t(\cdot,\cdot), D_s(\cdot,\cdot)$ as the χ^2 distance between their intensity, texture and SIFT histograms. ω_i, ω_t and ω_s are set to 0.10, 0.45 and 0.45 in all our experiments. We show the top ranked internet images based on this distance in Fig. 3(b).

4.2 Background Image Filtering

To limit the amount of user interaction, our system does not require labeling and segmentation of background regions, but instead capitalizes on the strong correlation that exists between the foreground and background components of a scene [Oliva and Torralba 2007]. Reference images for the background regions in the input grayscale photo can generally be found among the images that were downloaded for the foreground objects, so we filter these images for suitable reference backgrounds as well.

In comparison to labeled foreground objects, close matches for backgrounds are difficult to find, since they tend to exhibit much more diversity. Rather than employ the filtering method used for foreground objects, we utilize the compact GIST descriptor [Oliva and Torralba 2006], which has been shown to be effective at finding semantically similar scenes such as forests and beaches. We exclude the foreground from the background filtering process by specifically adopting the weighted GIST descriptor [Hays and Efros], where the segmentation mask is used to weight each spatial bin of the GIST descriptor in proportion to the number of valid pixels in it. Filtering is thus performed using the sum of squared differences between the GIST descriptor of the input image and that of the internet images, weighted by the segmentation masks of both the internet and input images. Note that the background masks for the internet images are simply the inverse of those computed for foregrounds during foreground image filtering. The top row of Fig. 4 shows some background images found to be most relevant to the input background according to this method.

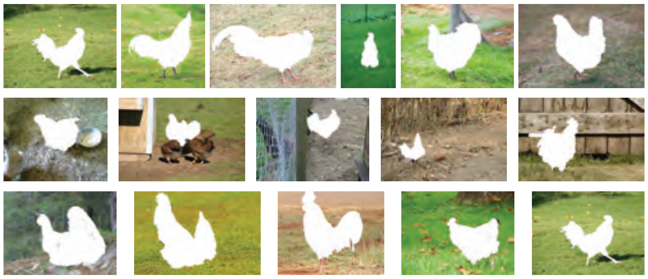


Figure 4: *Selection of background images for Fig. 1(a). Top internet images selected by [Hays and Efros] (top row), [Chen et al. 2009] (middle row), and [Zhu et al. 2011] (bottom row).*

5 Image Colorization

We colorize each scene object and the background according to their highest scoring internet images. First, colors are transferred at the resolution of super-pixels using an energy minimization framework. These results are then smoothed using guided image filtering [He et al. 2010] to suppress color noise while preserving color structures.

5.1 Colorization by optimization

Super-pixel representation All internet objects are first normalized to have the same diagonal bounding box length as the input object. We then apply over-segmentation [Comanicu and Meer 2002] with stringent thresholds (of spatial bandwidth 1 and range bandwidth 3) to break these images into super-pixels. Typically, there are between 1000 to 5000 super-pixels per image. We describe each super-pixel by the average intensity, texture, SIFT feature values and spatial coordinates of its pixels. We also extract the median CbCr color values from each super-pixel of a reference object.

Graph-based optimization In assigning colors from reference images to the grayscale photo, we aim to optimize the quality and likelihood of correspondences while favoring smoothness/consistency between neighboring super-pixels. We compute a color c_a for each super-pixel P_a of the grayscale input image by minimizing the following energy function:

$$E = \sum_{P_a} E_d(c_a, P_a) + \lambda \sum_{P_b \in N(P_a)} E_s(c_a, c_b, P_a, P_b) \qquad (2)$$

where $N(P_a)$ denotes the set of neighboring super-pixels of P_a, and λ is a weight parameter $\lambda \in \{0.1, 0.5\}$.

The term $E_d(\cdot,\cdot)$ defines the data cost of assigning color c_a to P_a. To evaluate this cost, we first find the internet object super-pixel of color $c = c_a$ with the most similar intensity to P_a. We then evaluate the intensity based cost as

$$E_d^{intensity}(c, P_a) = \alpha_i 1/Pr_{intensity}(c, i) + \beta_i ||i_a - i|| \qquad (3)$$

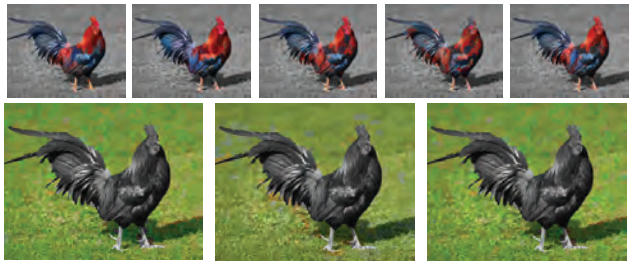


Figure 5: *Colorization results with different weight settings.*

where i_a and i are the intensity of P_a and this internet object super-pixel respectively. $Pr_{intensity}(c, i) = \sum_{P_k \in \Upsilon} |P_k|/A$ is the probability that an internet object pixel has the intensity i and color c. Hence, the cost will be smaller if many pixels of the internet object have color c. The second term $||i_a - i||$ is the difference between their intensities, which favors internet object super-pixels with an intensity similar to P_a. $|\cdot|$ denotes the number of pixels within a super-pixel, and Υ denotes the set of internet object super-pixels whose intensity and color values are i and c respectively. The normalization factor A is the total number of pixels in the internet object. α_i, β_i are normalization coefficients to ensure $1/Pr_{intensity}(c, i)$ and $||i_a - i||$ vary from 0 to 1.

Similarly, we also define

$$\begin{aligned} E_d^{texture}(c, P_a) &= \alpha_t 1/Pr_{texture}(c, t) + \beta_t ||t_a - t|| \\ E_d^{SIFT}(c, P_a) &= \alpha_s 1/Pr_{SIFT}(c, s) + \beta_s ||s_a - s|| \\ E_d^{spatial}(c, P_a) &= ||x_a - x||. \end{aligned}$$

Here, $Pr_{texture}(c, t)$ (or $Pr_{SIFT}(c, s)$) is the probability of a pixel having color c and texture t (or SIFT value s). The normalization coefficients $\alpha_t, \beta_t, \alpha_s, \beta_s$ are determined in the same way as α_i, β_i. We denote t_a, s_a, x_a as the average texture, SIFT value and spatial position of the super-pixel P_a respectively. Similarly, t, s, x are respectively the texture, SIFT value and position of the internet object super-pixel of color c with the minimum Euclidean distance to t_a, s_a, x_a.

The overall data cost is defined as

$$\begin{aligned} E_d(c, P_a) &= w_1 E_d^{intensity}(c, P_a) + w_2 E_d^{texture}(c, P_a) \\ &+ w_3 E_d^{SIFT}(c, P_a) + w_4 E_d^{spatial}(c, P_a) \qquad (4) \end{aligned}$$

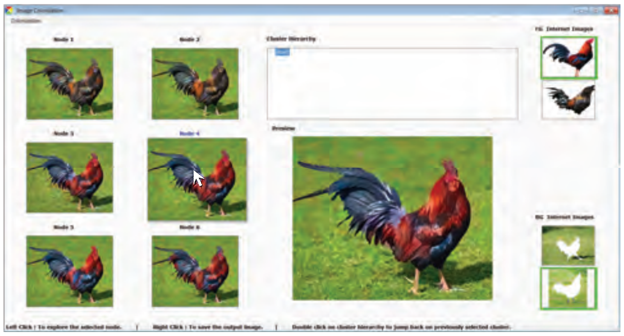
where $w_1, \ldots w_4$ are the combination weights, $w_i \in \{0, 10, 20, 30\}$.

The term $E_s(\cdot,\cdot,\cdot,\cdot)$ measures smoothness between two super-pixels and is defined as

$$E_s\left(c_a, c_b, P_a, P_b\right) = \mathcal{F}(a, b)||c_a - c_b|| \qquad (5)$$

where c_a and c_b are the colors assigned to P_a and P_b respectively, and $\mathcal{F}(a, b) = \exp^{-(w_1||i_a - i_b|| + w_2||t_a - t_b|| + w_3||s_a - s_b||)}$ controls the relative strength of the smoothing term. We use the belief propagation framework [Komodakis and Tziritas 2007] to minimize the overall objective function and obtain the color assignments.

Weight sampling The performance of our method depends heavily on weights $w_1 \ldots w_4$, which control the relative importance of different features, and on λ which controls the extent of smoothing. In practice, we find that fixed weights cannot generate good results for all data. To address this problem, we colorize the object using many different weight settings, cluster the results, and let the user choose one via an intuitive interface. We sample each weight value w_i as 0, 10, 20 or 30 and λ as 0.1 or 0.5, which yields 512 different results for each scene object. Prior to clustering, we evaluate the colorfulness of these results [Hasler and Strunk 2003], and discard images whose color quality is below the recommended threshold in [Hasler and Strunk 2003]. The remaining images are grouped by k-means clustering (k is set to 5), where image distance is computed by the χ^2 distance of color histograms. From each cluster, we retain the two images that are closest to the internet color reference image. The top row of Fig. 5 displays examples of retained foreground object colorizations.



Left Click : To explore the selected node. | Right Click : To save the output image. | Double click on cluster hierarchy to jump back on previously selected cluster.

Figure 6: *Our graphical user interface, based on a hierarchical cluster tree, helps the user to quickly select the preferred image.*

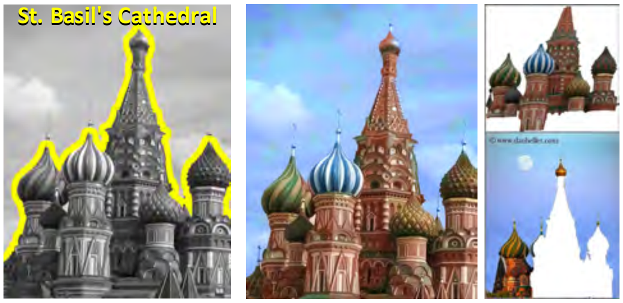


Figure 7: *Comparison to intrinsic colorization [Liu et al. 2008].*

We apply the same method for background colorization, except with $w_4 = 0$ because the geometric relationship between backgrounds is weak. The bottom row of Fig. 5 shows some colorized backgrounds.

5.2 Interface for result selection

To compose a diverse set of colorized results, we typically use four to six reference internet images for each input grayscale image. Colorized foreground objects and backgrounds are then combined to form a set of results. We have designed a user interface (see Fig. 6) to help the user to quickly select a desirable result. We apply hierarchical k-means clustering (k=6) to the set of colorization results to generate a hierarchical tree of images, where images in the same cluster have similar color. We display six cluster centers to the user at a time. Clicking on a thumbnail brings the user one level down the hierarchical tree, and the six cluster centers at that level will then be shown. With this interface, a user can select a colorized photo from among the $O(6^n)$ results in n clicks.

6 Experiments

We first evaluate our filtering methods and compare them with existing works. Fig. 3(b)-(e) show filtering results on the ‘rooster’ example by our method, [Chen et al. 2009], [Hays and Efros] and [Zhu et al. 2011], respectively. The foreground objects returned by our method are perceptually more similar to the input. Back-

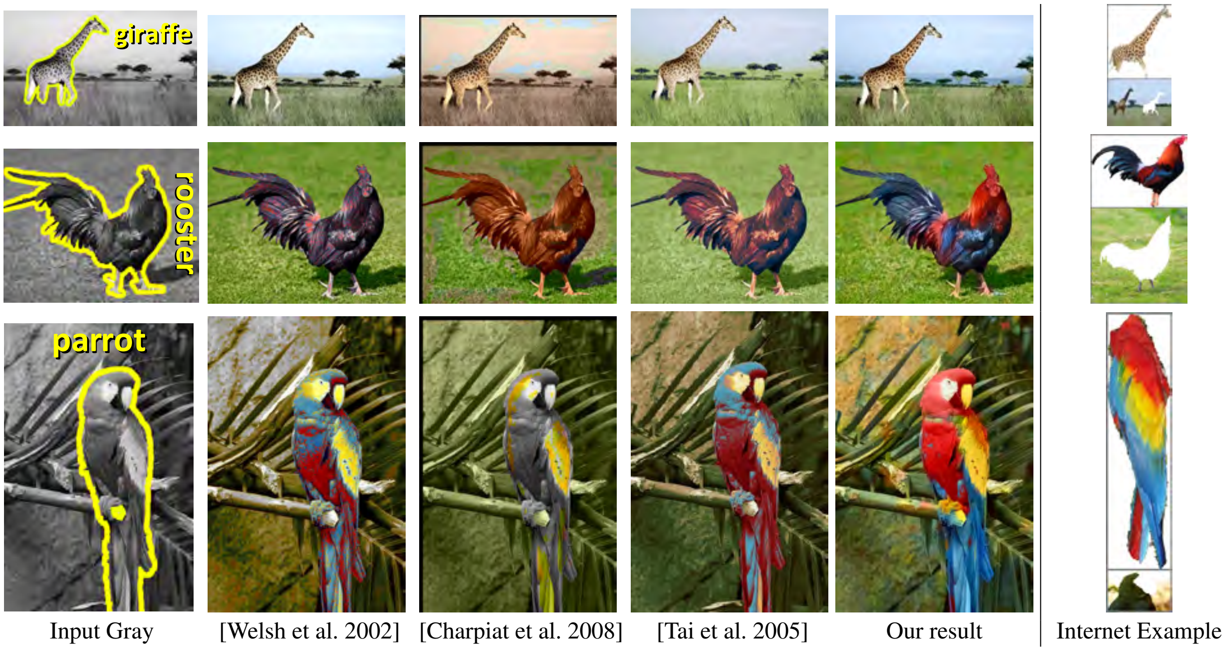


Figure 8: *Comparison of colorization methods. The last column shows the foreground and background reference examples selected using our filtering technique. These reference images were used for all algorithms in this comparison.*

ground filtering results by [Hays and Efros], [Chen et al. 2009] and [Zhu et al. 2011] are shown in the top, middle and bottom rows of Fig. 4, respectively. Here, it can be seen that [Hays and Efros], which utilizes GIST descriptors, finds background images perceptually similar to the input background. More filtering results are provided in the supplementary materials.

We compare our colorization to [Liu et al. 2008] in Fig. 7. The reference internet examples used for our colorization are shown on the right. Our system produces a result similar in quality to that shown in Fig. 1(c) of [Liu et al. 2008], but is capable of colorizing a substantially broader range of imagery. We note that [Liu et al. 2008] requires the reference internet object to be identical to the target object for precise per-pixel registration between reference and target objects. Hence, it is limited to colorizing objects like landmark buildings. In Fig. 8, we compare our method with [Welsh et al. 2002], [Charpiat et al. 2008] and [Tai et al. 2005]. For all the methods, the foreground objects and background image were colorized separately using the same reference internet examples obtained by our filtering. The threshold settings for [Welsh et al. 2002; Charpiat et al. 2008; Tai et al. 2005] were varied to obtain the best results for these methods. Our method is seen to work well for images with different amounts of texture, as well as for very colorful objects such as the rooster and parrot. From our observations, the amount of time a user needs to spend on semantic labeling and providing segmentation cues for foreground objects, and selecting a final result from the user interface, is typically about 1-1.5 minutes in total for the images used in this work.

Fig. 9 exhibits several more colorization results from our method. For colorizing input images with multiple objects, we use a semantic label to download internet images for each object, and utilize these images to colorize the object. The background is colorized using the collective sets of downloaded internet images. Fig. 10 displays different colorizations of an input image using different internet examples. We show colorization results using different keywords (given at the top of each result) in Fig. 11. Colorizations obtained with excessively broad keywords are shown on the left, and those obtained with more specific keywords are on the right. It is seen that colorizations are poor with excessively broad keywords. This is due to the lack of relevant internet images that are downloaded. Taking the input parrot image in the top row of Fig. 11 as an example, visual inspection yields no useful parrot images from the first 1000 internet images downloaded by Flickr with the 'animal' keyword. Colorizations improved markedly when slightly more compatible keywords are used (*e.g.* using 'bird' as the keyword for the parrot image). To zoom into the images, please view the pdf file. Additional colorization results are provided in the supplementary materials.

6.1 User study

We performed a user study to quantitatively evaluate our method. We chose 120 images from the internet and randomly converted 30 of them to grayscale (using Matlab's *rgb2gray* function). The 30 images were then colorized by our method and the methods in [Welsh et al. 2002; Charpiat et al. 2008; Tai et al. 2005]. The other 90 images were used for comparison.

We show each subject a set of four different images at a time (such as in Fig. 12) for a total of 30 sets, and asked the subject to identify all artificially colorized photos in each set. Beforehand, the subject is told that at most two of the four images in each set are artificially colorized. In each set, the colorized images (if present) are obtained using the same method, to avoid bias against a comparatively weaker method. We perform two tests. In the first test, the subject is given five seconds to view the four displayed images. In the second test, the subject has unlimited time. In total, thirty subjects took part in this study.

In the first test, the subjects classified results of our system as real 66.59% of the time. This compares favorably to the 48.90% obtained by [Welsh et al. 2002], 32.30% by [Charpiat et al. 2008] and 40.79% by [Tai et al. 2005]. Interestingly, they identified original color images as real only 90.57% of the time. We believe this is due to subjects scrutinizing these images to such an extent to believe they were artificially colored. In the second test, the subjects identified our results as real 64.52% of the time, while that by [Welsh et al. 2002], [Charpiat et al. 2008] and [Tai et al. 2005] drop to 45.23%, 27.36% and 39.55% respectively. The original color images were identified as such 90.33% of the time. These re-

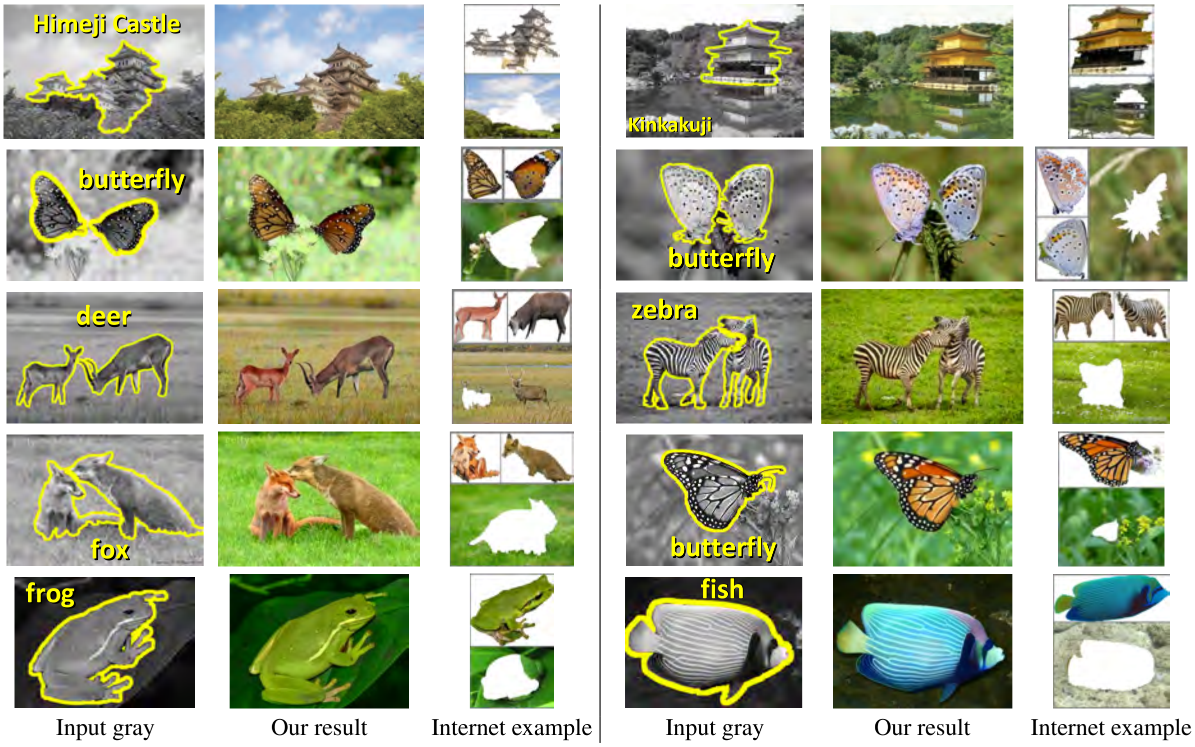


Figure 9: *Colorization results obtained using the proposed method.*

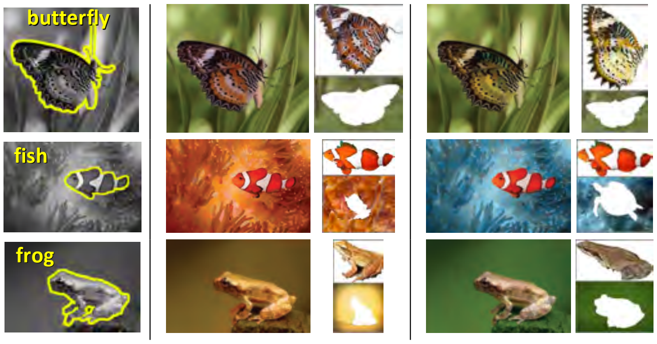


Figure 10: *Colorization results using different internet examples.*

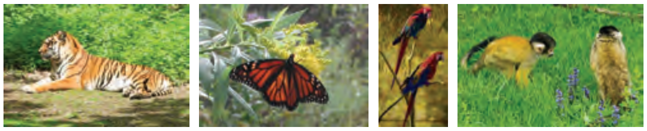


Figure 12: *Examples of images displayed for quantitative evaluation. The third image is artificially colored with our colorization method, while the others are original photos.*

sults demonstrate the better performance of our method, where even with unlimited viewing time, over half of the colorized images are of sufficient quality to appear real. A t-test shows the comparison results to be statistically significant, ($p < 10^{-6}$).

7 Conclusion

We proposed a novel colorization method that utilizes internet photos and image filtering to minimize user effort and facilitate accurate color transfer. Both image filtering and colorization results were shown to outperform related methods.

There are limitations of our system that we plan to investigate in future work. One is that foreground segmentation by Lazy Snapping can be coarse for boundaries with fine-scale structure, such as butterfly feelers. Methods for alpha matting may be more appropriate in such cases. Another is that color transfer for background regions is generally less accurate than for foregrounds, since spatial constraints cannot be as effectively leveraged for background matching. We plan to examine more discriminative region based properties to compensate for this lack of spatial information. Thirdly, scenes with many foreground objects may be time-consuming for users to label and segment. To address this issue, we intend to reduce user interaction by taking advantage of foreground object co-occurrences within a scene, in a manner similar to backgrounds in the current implementation. Additionally, for complex scenes with serious occlusions between objects, our image filtering may fail as shape context matching favors examples with a similar contour, which may be difficult to find. Incorporating object recognition into the image filtering framework may help to identify relevant internet examples in such cases. We also plan to implement our internet-based system as a web-based application where users can benefit from the massively parallel search resources of a web server.

Acknowledgements: We thank the anonymous reviewers for their valuable input, and Yianling Cheong for her help in preparing the supplementary video. This work is supported by Institute for Infocomm Research and by Microsoft Research Asia. Shaojie Zhuo and Ping Tan are supported by Singapore Grant R-263-000-555-112. Yu-Wing Tai is supported by the NRF-2011-0013349 of Korea.

References

BELONGIE, S., MALIK, J., AND PUZHICA, J. 2002. Shape matching and object recognition using shape contexts. *IEEE Trans. PAMI 24*, 4, 509–532.

CHARPIAT, G., HOFMANN, M., AND SCHÖLKOPF, B. 2008. Automatic image colorization via multimodal predictions. In *Proc. ECCV*, 126–139.

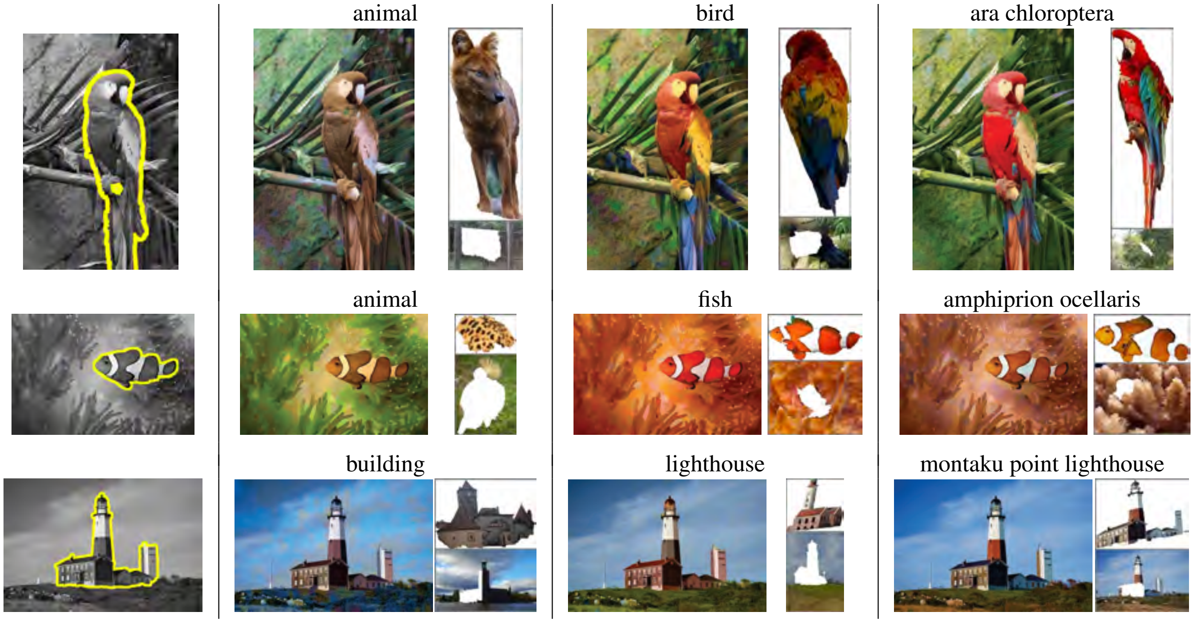


Figure 11: *Colorization results using different keywords (shown at the top of each result). Colorizations obtained with excessively broad keywords are shown on the left, and those obtained with more specific keywords on the right. Results obtained with another keyword for the top input image can be found in Fig. 8.*

CHEN, T., CHENG, M.-M., TAN, P., SHAMIR, A., AND HU, S.-M. 2009. Sketch2photo: internet image montage. In *ACM Trans. Graph.*, vol. 28, 1–10.

COMANICU, D., AND MEER, P. 2002. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. PAMI 24*, 603–619.

HASLER, D., AND STRUNK, S. 2003. Measuring colourfulness in natural images. In *Human Vision and Electronic Imaging*.

HAYS, J., AND EFROS, A. Scene completion using millions of photographs. *ACM Trans. Graph. 26*, 87–94.

HE, K., SUN, J., AND TANG, X. 2010. Guided image filtering. In *Proc. ECCV*, 1–14.

HERTZMANN, A., JACOBS, C. E., OLIVER, N., CURLESS, B., AND SALESIN, D. H. 2001. Image analogies. In *SIGGRAPH*, 327–340.

HUANG, Y.-C., TUNG, Y.-S., CHEN, J.-C., WANG, S.-W., AND WU, J.-L. 2005. An adaptive edge detection based colorization algorithm and its applications. In *Proc. ACM Multimedia*.

IRONY, R., COHEN-OR, D., AND LISCHINSKI, D. 2005. Colorization by example. In *Proc. EGSR*, 201–210.

KOMODAKIS, N., AND TZIRITAS, G. 2007. Approximate labeling via graph-cuts based on linear programming. *IEEE Trans. PAMI 29*, 8, 1436–1453.

LEVIN, A., LISCHINSKI, D., AND WEISS, Y. 2004. Colorization using optimization. *ACM Trans. Graph. 23*, 689–694.

LI, Y., SUN, J., TANG, C.-K., AND SHUM, H.-Y. 2004. Lazy snapping. In *ACM Trans. Graph.*, 303–308.

LIU, T., SUN, J., ZHENG, N.-N., TANG, X., AND SHUM, H.-Y. 2007. Learning to detect a salient object. In *Proc. CVPR*, 1–8.

LIU, X., WAN, L., QU, Y., WONG, T.-T., LIN, S., LEUNG, C.-S., AND HENG, P.-A. 2008. Intrinsic colorization. *ACM Trans. Graph. 27*, 1–9.

LOWE, D. G. 1999. Object recognition from local scale-invariant features. In *Proc. ICCV*, 1150–1157.

LUAN, Q., WEN, F., COHEN-OR, D., LIANG, L., XU, Y.-Q., AND SHUM, H.-Y. 2007. Natural image colorization. In *Proc. EGSR*, 309–320.

OLIVA, A., AND TORRALBA, A. 2006. Building the gist of a scene: The role of global image features in recognition. In *Visual Perception, Progress in Brain Research*, vol. 155.

OLIVA, A., AND TORRALBA, A. 2007. The role of context in object recognition. *Trends in Cognitive Sciences 11*, 12.

QU, Y., WONG, T.-T., AND HENG, P.-A. 2006. Manga colorization. *ACM Trans. Graph. 25*, 1214–1220.

REINHARD, E., ASHIKHMIN, M., GOOCH, B., AND SHIRLEY, P. 2001. Color transfer between images. *IEEE Comput. Graph. Appl. 21*, 34–41.

ROTHER, C., KOLMOGOROV, V., AND BLAKE, A. 2004. Grabcut: Interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph. 23*, 309–314.

TAI, Y.-W., JIA, J., AND TANG, C.-K. 2005. Local color transfer via probabilistic segmentation by expectation-maximization. In *Proc. CVPR*, 747–754.

WELSH, T., ASHIKHMIN, M., AND MUELLER, K. 2002. Transferring color to greyscale images. *ACM Trans. Graph. 21*, 277–280.

YATZIV, L., AND SAPIRO, G. 2006. Fast image and video colorization using chrominance blending. *IEEE Trans. PAMI 15*, 1120–1129.

ZHU, J., HOI, S., LYU, M., AND YAN, S. 2011. Near-duplicate keyframe retrieval by semi-supervised learning and nonrigid image matching. *ACM Trans. Multimedia Comput. Commun. Appl.*.