Content Based Image Synthesis

Nicholas Diakopoulos, Irfan Essa, Ramesh Jain

GVU Center / Georgia Institute of Technology
{nad|irfan|rjain}@cc.gatech.edu

Abstract. *A new method allowing for semantically guided image editing and synthesis is introduced. The editing process is made considerably easier and more powerful with our content-aware tool. We construct a database of image regions annotated with a carefully chosen vocabulary and utilize recent advances in texture synthesis algorithms to generate new and unique image regions from this database of material. These new regions are then seamlessly composited into a user's existing photograph. The goal is to empower the end user with the ability to edit existing photographs and synthesize new ones on a high semantic level. Plausible results are generated using a small prototype database and showcase some of the editing possibilities that such a system affords.*

1 Introduction and Related Work

The ongoing digital media revolution has resulted in an untold amount of new digital content in the form of images and videos. Much of this digital content is generated by capturing real scenarios using cameras. We feel that digital media production would become much simpler, more effective, and cheaper if intuitive tools were available for combining *existing* photos to generate *new* images. Merging images and image segments together to form new images has been around for a long time in the form of image and video compositing. The composition process is however tedious and best carried out by the practiced eye of an artist. For such synthesis an artist combines various image regions from different sources to achieve a predefined content and context in the final image. We seek to simplify this process by providing a content-aware tool for generating new images.

To facilitate this form of semantic image synthesis, we first create a *database* of imagery which has regions annotated with semantic labels and image characteristics. Then, based on the content that the user desires, the user can *query* this database for imagery that will suit the region that will be composited. The user then chooses an image region from the query results, which acts as a source for texture *synthesis* algorithms [1–3]. The synthesized region is composited into the image being edited. The system pipeline is shown in Fig. 1. In this way a user can edit a photo by synthesizing any number of user defined regions from an existing database of imagery. We have termed this semantically guided media recombination process Content Based Image Synthesis (CBIS).

At the heart of the CBIS method is a reliance on semantic annotations of regions (e.g. sky, mountain, trees etc.) so that if the user wants to synthesis a new mountain range into his photo he can search the database for other mountain regions. The idea

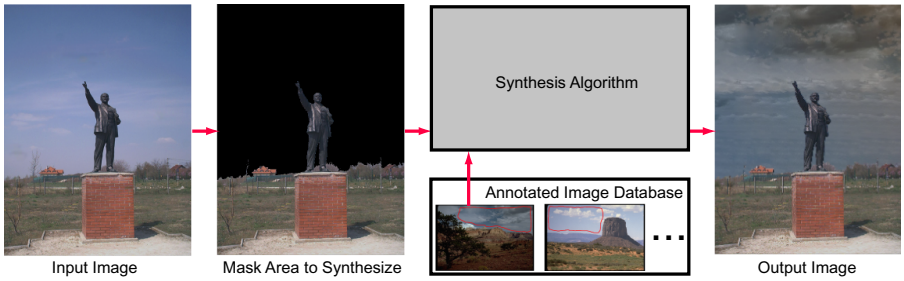

Fig. 1. The flow of the CBIS application. An input image is loaded and the area to replace masked. A query against an annotated image region database is then made by a user in order to find suitable content with which to fill the area. This source region is then used by a texture synthesis algorithm to produce a new region which is finally composited into the input image.

of using such high level annotations for doing search and synthesis has been applied successfully by Arikan [4] in the domain of motion data for animation. Their vocabulary, consisting of such verbs as *walk*, *run*, and *jump* is used to annotated a database of motion data. When the vocabulary is applied to a timeline, the system synthesizes plausible motion which corresponds to the annotated timeline.

The analogue in the image domain is the segmentation and annotation of distinct image regions with high level semantic tags. Zalesny [5] segments textures into subtextures based on clique partitioning. Each subtexture is represented by a label in a spatial label map. The label map guides their texture synthesis procedure such that the correct sub texture appears in the correct spatial position. Hertzmann [6] also defines some notion of a label map in his texture-by-numbers synthesis scenario. Our method borrows from these ideas for a label map but defines a much higher level mapping, less based on image statistics and more based on the semantic annotation of the region. For example, the user may desire one region to be synthesized as *sky* and another region as *mountain*. These region annotations ultimately map to the regions from the database that have been selected by the user as synthesis sources.

In the next section we further motivate the semantic power of such a CBIS application using semiotics. In section 3 we detail our methods for segmentation and annotation of images. Section 4 lays out our query method and details the hybrid texture synthesis approach used. Finally, we show some results that have been generated with the system in 5 and conclude with a number of directions for future work.

2 Semiotics Basics

Denotation and Connotation: Semiotics provides us with a rich set of abstractions and quasi-theoretical foundations supporting our recombinant media application. A user may want to composite a newly synthesized region into an existing photograph for practical purposes, aesthetic reasons or, from a semiotic perspective, to change the *meaning* of the photograph. Barthes [7] identifies several ways in which one can alter the connotation of an image (e.g composition, pose, object substitution/insertion, photogenia

etc.). Our system thus uses image composition to enable the user to alter both the denotation (pixels) and the connotation. For example, given the photo of the stalin statue with the happy sky seen in Fig. 1 (input), we might want to substitute a cloudy or stormy sky (output). The resultant change in meaning is left to the subjective interpretation of the reader.

Structural Analysis: Semiotic systems can be defined structurally along two primary axes: *syntagm* and *paradigm*. The syntagm defines the spatial positioning of elements in an image, whereas the paradigm is defined by the class of substitutions that can be made for a given element [7–9]. A linguistic analogy is usually helpful for understanding. The syntagm of a sentence corresponds to its grammatical structure (e.g. noun-verb-prepositional phrase); the paradigm corresponds to the set of valid substitutions for each word in the sentence. For instance, a verb should be substituted with a verb in order that the sentence still make sense.

Syntagm and paradigm are the structural axes along which a user may vary an image in the course of editing. We can consider three combinations of variation along these axes: (1) vary the paradigm and fix the syntagm; (2) fix the paradigm and vary the syntagm; or (3) vary both the paradigm and syntagm. (1) is analogous to playing with a Mr. Potatoe Head: the structure of the face remains the same, but various noses, mouths, etc. can be substituted into that structure. (2) roughly corresponds to the method of texture-by-numbers in [6]. In that work, a new syntagm (a label map) is drawn by a user and filled in with the corresponding labelled regions from a source image. (3) defines a more difficult operation since changing the syntagm of an image can change the valid paradigmatic substitutions as well. We consider variation (1) in this work. The layout of the input photo remains the same (i.e. mountains stay mountains of the same shape), but the database provides a set of paradigmatic variations (i.e. rocky mountains can become snowy mountains).

3 Database Generation

The database creation process consists primarily of segmenting meaningful and useful regions in images and annotating them with the appropriate words from our annotation vocabulary. Each of these segmented, annotated regions is then stored in an XML structure which can later be queried by the end user. In our prototype system we have annotated slightly more than 100 image regions which serve as the database.

Region Segmentation: The first step in generating the database of images is the segmentation of meaningful regions in these images. While there has been some recent progress in the automatic segmentation of semantically meaningful regions of images [10–12], or even semi-automatic segmentation, currently we opt for a fully manual procedure as this allows for a more directed user input of higher semantic import.

Region Annotation: The annotation vocabulary is chosen to fit the domain of natural landscape imagery. This domain makes sense for us since textures are prevalent and because a relatively small vocabulary can describe the typical regions in such a scene.

The choice of words used to annotate quantities like hue, saturation, and lightness is informed by [13, 14]. The vocabulary follows:

Region: {Sky | Mountains | Trees | Vegetation | Water | Earth | Hue | Lightness | Saturation | Distance}
Sky: {*Clear* | *Partly Cloudy* | *Mostly Cloudy* | *Cloudy* | *Sunset* | *Sunrise* | *Stormy* }
Mountains: {*Snowy* | *Desert* | *Rocky* | *Forested*}
Trees: {*Deciduous* | *Coniferous* | *Bare*}
Vegetation: {*Grass* | *Brush* | *Flowering*}
Water: {*Reflective* | *Calm* | *Rough* | *Ocean* | *Lake* | *Stream* | *River*}
Earth: {*Rocky* | *Sandy* | *Dirt*}
Hue: {*Red* | *Orange* | *Yellow* | *Green* | *Blue* | *Purple* | *Pink* | *Brown* | *Beige* | *Olive* | *Black* | *White* | *Gray*}
Lightness: {*Blackish* | *Very Dark* | *Dark* | *Medium* | *Light* | *Very Light* | *Whitish*}
Saturation: {*Grayish* | *Moderate* | *Strong* | *Vivid*}
Distance: {*Very Close* | *Close* | *Medium* | *Distant* | *Very Distant* | *Changing*}

Region annotations are made manually using a GUI. Relying on a fully manual process allows us to work with higher level semantic categories. Of course, as automatic annotation methods get better, they can be integrated to supplement the manual procedure. Given the categories chosen above, there should be little problem with consensus on the appropriate annotation(s) for a given region, though in general subjectivity can be a problem for manual annotation.

The hue, saturation, and lightness user annotations are augmented by fuzzy histograms of HSL pixel values. A 13 bin hue histogram, a 4 bin saturation histogram, and a 7 bin lightness histogram are generated based on the HSL pixel values in a given region. Bins are fuzzy insofar as a given pixel can contribute (bi-linearly) to adjacent bins. Each bin also corresponds to one of the vocabulary words for that category; for saturation the 4 bins correspond to <Grayish, Moderate, Strong, Vivid>. A saturation histogram of <.8, .2, 0, 0> therefore indicates a very grayish region.

This dual representation of lower level features should also mitigate the somewhat subjective user annotations. Currently, we are also studying other automatic or semi-automatic methods for annotating entities such as the lighting direction, or camera perspective. These would serve to make database query results even more pertinent to the image into which they will be synthesized.

4 Image Recombination

Query: The image recombination procedure begins with the user defining a region that will be replaced in his input image (e.g. a mountain range is selected). This region is then annotated with keywords from the vocabulary using a drag and drop GUI. This annotation is used to query the XML database and return the N most pertinent images from which the user selects a source image. Subtle decisions in lighting, perspective, and color are thus not made automatically and can be evaluated by the user. This maximizes the user's potential to affect connotation in the image since he has good suggestions from the database, but ultimately has the final choice in the paradigmatic substitution.

Matching of annotated regions proceeds as suggested by Santini [15]. This approach allows for binary feature sets (i.e. presence/absence of keywords) to be compared with fuzzy predicates. Thus we can compare keyword annotations of a region's lower-level

features (e.g. saturation) with the histograms of those features as calculated directly from pixel values.

Let $\hat{f}(R_i) =< f_1(R_i), f_2(R_i), ..., f_p(R_i) >$ represent a p-dimensional feature vector for a region R_i. $f_k(R_i) = 1$ if region R_i has the keyword annotation associated with feature $k \in \{1...p\}$. For fuzzy features such as saturation we maintain two feature vectors with $f_k(R_i) \in (0, 1)$. One vector is based directly on the histogram of pixel values. The other vector is based on the keyword annotation, but is also made fuzzy. As an example let's consider the two feature vectors describing saturation for R_1. The feature vector calculated from pixel saturation values might be: $\hat{f}(R_1) =< 0, .1, .7, .2 >$. If R_1 also has the *strong* keyword annotation the second feature vector is $\hat{f}(R_1) =< 0, .25, 1, .25 >$. This fuzzyness allows for more meaningful retrieval results since it allows for a smoother range of similarity scores. In addition to hue, saturation, and lightness, we make the distance vector fuzzy since it also benefits from a gradient score. In general, any attribute which can naturally be described using an intensity scale benefits from a fuzzy representation.

Based on [15] a symmetric similarity function σ is defined between two feature vectors in the following way,

$$\sigma(\hat{f}(R_a), \hat{f}(R_b)) = \sum_{i=1}^{p} min(f_i(R_a), f_i(R_b)) \quad (1)$$

The dissimilarity δ can be written as, $\delta(\hat{f}(R_a), \hat{f}(R_b)) = p - \sigma(\hat{f}(R_a), \hat{f}(R_b))$. We maintain a feature vector for each semantic category of each region, though we could concatenate these vectors and arrive at the same result. Comparing two image regions then consists of computing dissimilarity scores for each semantic category and summing them to arrive at an aggregate dissimilarity score between those two regions. Where necessary the dissimilarity of two regions also takes into account the dual feature vector representation by equally weighting the histogram vector and fuzzy keyword vector in the final score.

Synthesis: In general there are many methods for doing image based synthesis. Here we focus specifically on using patch-based texture synthesis algorithms such as those of [1–3]. Patch-based approaches work by copying small regions of pixels from a source texture such that when these regions are stitched together they give the same impression as the original texture. The output texture need not be the same shape or size as the source.

In particular we have implemented the method of texture synthesis and transfer detailed by Efros in [1]. Our pixel blocks are rectangular and tiled over the synthesis plane with overlapping regions through which seams are optimally cut using a dynamic programming algorithm (see Fig. 2). Each successive block of pixels is chosen by scanning the source texture for areas that will minimize a euclidean error metric as measured against imbricated adjacent areas. As many textures in our application are non-stationary and change due to perspective effects, we also use vertical correspondence maps when calculating the error metric. This ensures that parts of a source texture far away (i.e. at the top of a region) are used as samples when synthesizing the parts of the destination texture region that are also far away. Though this seems to work well

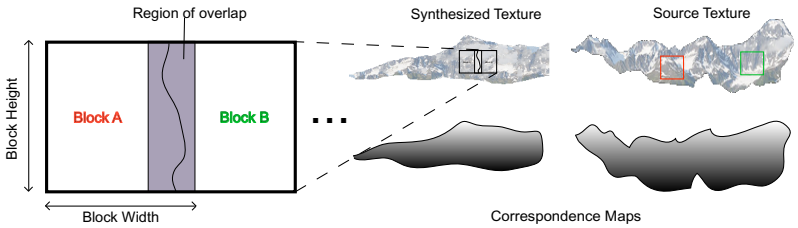

Fig. 2. A texture is synthesized. The blow up shows two overlapping blocks taken from the source and the seam between them. Vertical correspondence maps guide sampling from the source.

in practice, the amount of perspective can vary considerable between source and destination. Vertical correspondence maps can fail to generate convincing results in such cases.

Though the dynamic programming algorithm for seam generation detailed in [1] does a decent job for textures with some high frequency content, it falters for more slowly changing textures such as a sky gradient. In these cases, we smooth these seams with a technique in which a Poisson equation is solved across the boundary of the seam [16]. This has the effect of creating smooth transitions between blocks without sacrificing the saliency of the underlying texture.

There are a few algorithm parameters that are worth mentioning briefly here. The block size must be chosen large enough to capture the largest feature or texton size of the source texture. Choosing a value too small can lead to synthesized results which lack the structure of the original. Additionally, we allow the algorithm to iterate across the whole synthesized texture a variable number of times such that the initial pass is done with larger blocks (thus "laying out" the texture) and subsequent passes done with a smaller block size such that finer details are preserved.

5 Results

Queries were performed against a small proof of concept database of about 50 images, representing just over 100 distinct image regions. Synthesis timings vary according to parameters and region size in both the source and destination images, but were in the range of about 2 to 30 minutes for a Java implementation on a 2.4 GHz processor.

Some example images generated using our CBIS system are shown in Fig. 3 and Fig. 4. Results can also be viewed online at (http://cpl.cc.gatech.edu/projects/CBIS/); we encourage viewing of results digitally. In Fig. 3 (a) a city skyline is inserted; (b) a field of flowers is replaced with a field of rocks. Fig. 4 contains additional results: (a) a gentle sunset replaces a cloudy sky over Florence; (b) a distant island is inserted; (c) the night sky over Atlanta is replaced using the Starry Night painting by van Gogh leading to a unique blend of photograph and impressionist painting; (d) a rocky mountain is replaced with a snowy mountain.

The parameters for each of the images in Fig. 3 and Fig. 4 were chosen carefully. In particular the block size was chosen to be slightly larger than any repeatable features in

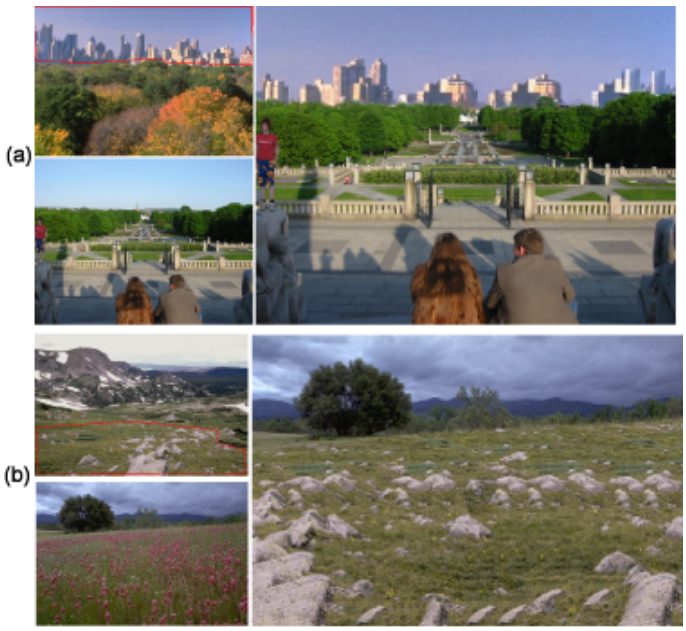

Fig. 3. Results generated using the CBIS application. Each block (a,b) consists of a source texture outlined in red (upper left); an input image (lower left); and the output (right).

the source. All sky replacements utilized the Poisson smoothing functionality. Vertical correspondence maps were used for all source and destination region pairs with the exception of Fig. 4 (c). Also, two synthesis passes were made for each output image.

6 Conclusions and Future Work

We have introduced a method for the content-based generation of new images. A user defines a region in his image to replace and then queries a database of images annotated with region semantics and image characteristics to find a suitable source image. The chosen source image is then used in a texture synthesis step to produce the final output image. Results, such as those seen in Fig. 3 and Fig. 4, are visually convincing.

Invoking semiotic theory allows us to view such a CBIS tool as a powerful way of changing meaning in photographic content though careful substitution and insertion of image elements. The extent of this editing power is dictated only by the size and breadth of the underlying database, and of the vocabulary used to annotated it. Thus, there are several obvious areas of future work such as expanding the database, annotating image regions with additional visual information such as perspective or lighting characteristics, and increasing the effective size of the query vocabulary by tying into a system such as WordNet. To expand the database we would specifically like to explore adding segmented objects to the database. A substitute object could then have the area around it filled in using a hole filling algorithm such as [17]. Improvements also need to be made in the synthesis phase, such as accounting for lighting effects (e.g. shadow in Fig. 1) or interactions between adjacent textures in the final image.

Another important direction for future work is in applying a content-based synthesis framework to other types of media such as video or audio. To be successfully applied in each of these domains several things must first be defined: (1) appropriate segmentation methods and units, (2) annotation vocabulary and low-level features, and (3) synthesis algorithms that combine the segmented units so that the output is believable. In short, workable segmentation and integration algorithms must be found for these other types of media.

7 Acknowledgements

Thanks to Derik Pack for his help implementing elements of the XML storage system. We appreciate the helpful comments of Stephanie Brubaker in preparing this paper. We also acknowledge the copyright holders of image segments used, many of which were downloaded from http://elib.cs.berkeley.edu/photos or elsewhere on the internet.

References

1. Efros, A.A., Freeman, W.T.: Image quilting for texture synthesis and transfer. In: Proceedings of ACM SIGGRAPH. (2001) 341–346
2. Ashikhmin, M.: Synthesizing natural textures. In: Symposium on Interactive 3D Graphics. (2001) 217–226
3. Kwatra, V., Schödl, A., Essa, I., Turk, G., Bobick, A.: Graphcut textures: Image and video synthesis using graph cuts. Proceedings of ACM SIGGRAPH (2003) 277–286
4. Arikan, O., Forsyth, D.A., O'Brien, J.F.: Motion synthesis from anotations. In: Proceedings of ACM SIGGRAPH. (2003) 402–408
5. Zalesny, A., Ferrari, V., Caenen, G., VanGool, L.: Parallel composite texture synthesis. In: ECCV Texture Workshop. (2002) 151–155
6. Hertzmann, A., Jacobs, C.E., Oliver, N., Curless, B., Salesin, D.H.: Image analogies. In: Proceedings of ACM SIGGRAPH. (2001) 327–340
7. Barthes, R.: Image, Music, Text. The Noonday Press, New York (1977)
8. Chandler, D.: Semiotics: The Basics. Routledge, New York (2002)
9. Manovich, L.: The Language of New Media. MIT Press, Cambridge, MA (2001)
10. Duygulu, P., Barnard, K., de Freitas, J.F.G., Forsyth, D.A.: Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In: Proceedings ECCV, Springer-Verlag (2002) IV: 97–112
11. Malik, J., Belongie, S., Shi, J., Leung, T.K.: Textons, contours and regions: Cue integration in image segmentation. In: Proceedings of ICCV. (1999) II: 918–925
12. Singhal, A., Luo, J., Zhu, W.: Probabilistic spatial context models for scene content understanding. In: Proceedings of CVPR. (2003) 235–241
13. Mojsilovic, A.: A method for color naming and description of color composition in images. In: Proc. Int. Conf. Image Processing (ICIP). (2002) 789–792
14. Corridoni, J.M., Del Bimbo, A., Pala, P.: Image retrieval by color semantics. Multimedia Syst. **7** (1999) 175–183
15. Santini, S., Jain, R.: Similarity measures. IEEE Transactions on Pattern Analysis and Machine Intelligence **21** (1999) 871–883
16. Perez, P., Gangnet, M., Blake, A.: Poisson image editing. In: Proceedings of ACM SIGGRAPH. (2003) 313–318
17. Criminisi, A., Perez, P., Toyama, K.: Object removal by exemplar-based inpainting. In: Proceedings of CVPR. (2003) II: 721–728

Fig. 4. Results generated using the CBIS application. Each block (a,b,c,d) consists of a source texture outlined in red (upper left); an input image (lower left); and the output (right).