Content Adaptive Image Matching by Color-Entropy Segmentation and Inpainting

Yuanchang Sun and Jack Xin

Math Department, Univ of California Irvine, Irvine, CA 92697, USA
yuanchas@uci.edu, jxin@math.uci.edu

Abstract. Image matching is a fundamental problem in computer vision. One of the well-known techniques is SIFT (scale-invariant feature transform). SIFT searches for and extracts robust features in hierarchical image scale spaces for object identification. However it often lacks efficiency as it identifies many insignificant features such as tree leaves and grass tips in a natural building image. We introduce a content adaptive image matching approach by preprocessing the image with a color-entropy based segmentation and harmonic inpainting. Natural structures such as tree leaves have both high entropy and distinguished color, and so such a combined measure can be both discriminative and robust. The harmonic inpainting smoothly interpolates the image functions over the tree regions and so blurrs and reduces the features and their unnecessary matching there. Numerical experiments on building images show around 10% improvement of the SIFT matching rate with 20% to 30% saving of the total CPU time.

Keywords: Content Adaptivity, Color-Entropy Segmentation, Inpainting, Image Matching.

1 Introduction

Image matching is a fundamental aspect of numerous tasks in computer vision, such as object and scene recognition, 3D structure reconstruction from multiple planar images, stereo correspondence, and motion tracking. Two key ingredients are critical to matching accuracy and efficiency. One is the selection of image features that are distinctive between different image classes yet invariant to unimportant attributes of the images (e.g. scaling and rotation), and partially invariant to change in illumination and camera viewpoint. The other is the design of a reliable matching criterion that quantifies feature similarity. A successful technique is Lowe's scale-invariant feature transform (SIFT, [5]). SIFT extracts from an image a collection of feature vectors (feature point descriptors), each of which is invariant to image translation, scaling, and rotation, partially invariant to illumination changes and robust to local geometric distortion. Such invariant features have proven useful in the context of image registration and object recognition. SIFT assigns location, scale, and orientation to each feature point. The location is defined as extrema of Difference of Gaussian functions

A. Berciano et al. (Eds.): CAIP 2011, LNCS 6855, pp. 471–478, 2011.

applied in scale-space to a series of smoothed and re-sampled images. Due to the large numbers of extrema, there could be up to thousands of detected feature points in an image, see Fig. 1. Yet many of the SIFT feature points are often found to be insignificant as they may well fall on uninteresting part of an image. As in Fig. 1, a building is typically surrounded by a natural landscape consisting of trees, lawns or bushes which can attract a large number of SIFT feature points. In the image of the left panel of Fig. 4, there are 4692 SIFT feature points. Almost half of them are on trees and lawns. Clearly, the feature points on the building are more important and reliable (less sensitive to weather and season) for building identification.

In this paper, we propose a content adaptive measure to exclude the uninteresting feature points, so subsequent image matching focuses more on the man-made corner-like features on the buildings. The adaptive measure combines color and entropy information of the image pixels. We shall locate the tree like natural regions by a combined measure of color and entropy, and apply image inpainting technique to blur and smooth out features in the tree regions. SIFT then acts on the inpainted image where regions of no interest have been weakened. Matching of feature points follows that of [5]. In brief, a descriptor (a multi-dimensional vector) is assigned to each feature point. A match p' in image A for a feature point p in image B is found by identifying the nearest neighbor of p from the feature points in image A. The nearest neighbor is defined as the feature point with minimum Euclidean distance in terms of the associated descriptor vectors. In [6], Lowe proposed a more robust matching criterion which we shall discuss later.

The paper is organized as follows. In section 2, we discuss the identification of tree like natural structures by a combined color and entropy measure. In section 3, we present an image inpainting method based on solving the Laplace equation on irregular domains to blur out the trees regions. In section 4, we discuss feature points matching criterion, and present experimental matching results on building images. The conclusion and remarks on future work are in section 5.

2 Color and Entropy Based Segmentation

Let us consider to match a building image to a similar image of the same scene. The images may contain background of trees, bushes, lawns, and sky. The goal is to locate the trees (or other vegetation) in an automatic fashion. A straightforward way is to use the colors. It is known that L*a*b color model is a better choice for images of natural scenes than RGB color space. The L*a*b* color space is derived from the CIE XYZ tristimulus values [4]. The L*a*b* space consists of a luminosity layer 'L*', chromaticity-layer 'a*' indicating where color falls along the red-green axis, and chromaticity-layer 'b*' indicating where the color falls along the blue-yellow axis. All of the color information is in the 'a*' and 'b*' layers. L*a*b color is designed to approximate human vision. It aspires to perceptual uniformity, and its L* component closely matches human perception of lightness. Therefore we propose to segment colors in the L*a*b* color space with K-means clustering method. We first convert image from RGB color

Fig. 1. There are 2280 SIFT feature points in the image, many of them fall on tree, lawn or bush areas

space to L*a*b* color space. Then we classify the colors in ‘a*b*’ space using K-means clustering. For example, three colors are used to segment the image into 3 images consisting of trees, building and sky. The segmented tree/lawn region is shown in the right panel of Fig. 2. Colors alone suffice for the segmentation in this example. In general, identification of tree like objects solely based on color is not enough. Some man-made structures could have nearly the same color as trees. Additional information is necessary. We observe that tree leaves and branches appear as rough or fractal, and so the entropy of image patches is helpful for their separation from uniform, and smoothly varying object surfaces. To compute the entropy of the gray-level images, let $V_{\max}$ be the maximum value in an image patch, the patch entropy is defined as

$$E = -\sum_{i=0}^{V_{\max}} h_i \log(h_i) \;, \tag{2.1}$$

where $h_i = n_i/N$ is the $i-$th histogram count n_i divided by the total number of pixels (N) in the image patch. Image patches in the tree regions in general have larger entropy than those in the uniform or regularly shaped objects such as buildings and sky. Measure of entropy proves to be rotation-invariant, and can serve as an indication of tree regions. However, other objects such as walls of building in the image may also have high entropy. Fig. 3 shows the white high entropy regions in an image, it can be seen that some parts of the building have large entropy too. This example suggests that a combination of color and entropy produces a better segmentation criterion. The idea is to first segment the image via colors, then compute the entropy of each segmented region. The part with high entropy will be recognized as belonging to trees. Now with the tree regions detected, we smooth and blur the image function in the tree regions by the Laplacian inpainting technique.

Fig. 2. The left panel is the original image. The right panel is the color segmented image consisting of trees and lawn.

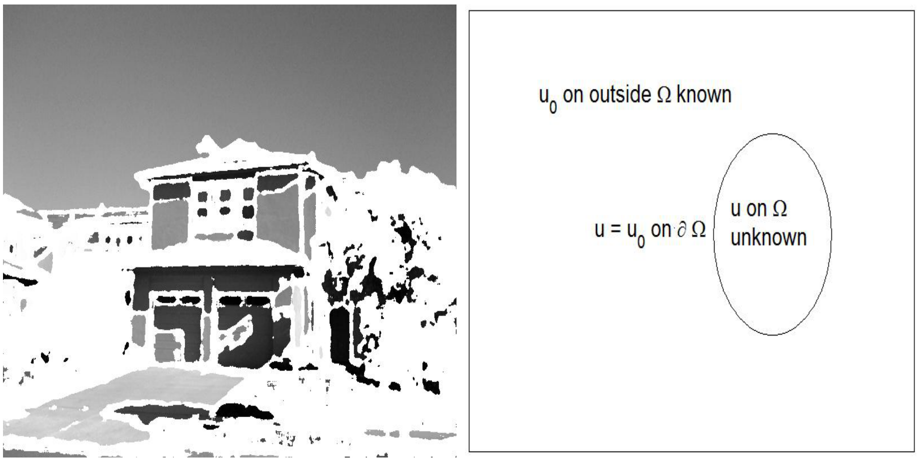

Fig. 3. Left panel: the white regions have high entropy. Right panel: the inpainting set-up. The domain Ω may have highly irregular boundaries in real-world images.

3 Harmonic In-Painting

Inpainting aims to restore a damaged or corrupted image. The goal of inpainting algorithm varies from making the restored parts of the image consistent with the rest of the image, to making it as close as possible to the original image. We consider the former, and hope to interpolate the values of the image function in the tree regions smoothly by outside values. This way, the image functions are blurred in the tree regions and so feature points will be sharply reduced there.

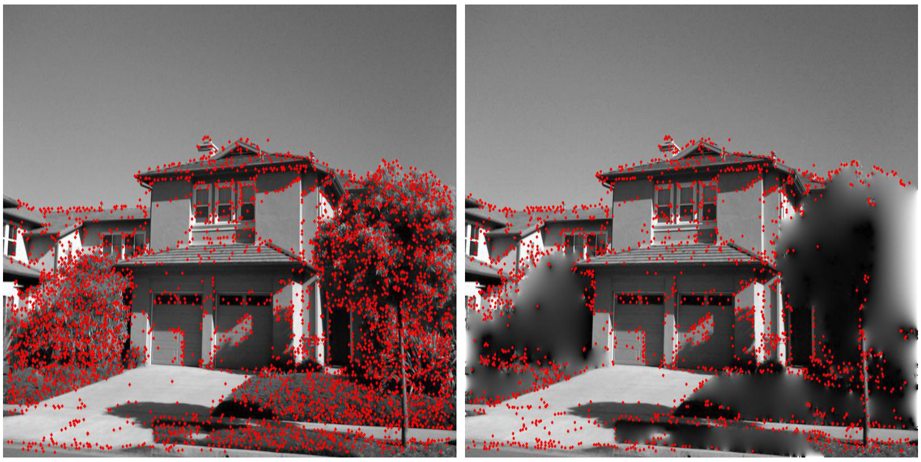

Fig. 4. SIFT features detected from the original image (left) and the preprocessed image (right). There are 4692 SIFT feature points detected from the original image. In contrast, there are 2097 SIFT feature points in the pre-processed image.

As a result, feature points are more focused on buildings or other man-made objects in the inpainted image. Since the original work of Bertalmio et al. [1], automatic inpainting algorithms have appeared in recent years. Inpainting is an interpolation problem: Given a rectangular image u_0 known outside a hole Ω (see Fig. 3), we want to find an image u, an inpainting of u_0, that matches u_0 outside the hole and has meaningful content inside the hole Ω. Broadly speaking, probabilistic [3] and variational [1,2] approaches are available for inpainting. We shall use the latter which has two representative prototypes: the harmonic and total variation (TV) models [2]. They amount to minimizing the square and absolute value of the gradient of the image, or solving the variational problems:

$$\text{Harmonic}: \min_{u \in \mathcal{A}_H} \int_{\Omega} |\nabla u|^2 dx \tag{3.1}$$

and TV : $\min_{u\in\mathcal{A}_{TV}} \int_{\Omega} |\nabla u| dx$. The admissibility sets are $\mathcal{A}_H = \{u \in W^{1,2}(\Omega) : u = u_0 \text{ on } \partial\Omega\}$ and $\mathcal{A}_{TV} = \{u \in BV(\Omega; R) : u = u_0 \text{ on } \Omega\}$. Here $W^{1,2}$ is Sobolev space of square integrable functions and square integrable gradients, BV is the function space of bounded variations. The minimizers can be found by solving the Euler-Lagrange partial differential equation: $\Delta u = 0$ on Ω, $u = u_0$ on $\partial\Omega$, for (3.1) and $\mathrm{div}\left(\frac{\nabla u}{|\nabla u|}\right) = 0$ on Ω, $u = u_0$ on $\partial\Omega$ for the TV case. Harmonic inpainting gives a desired smoother inpainted image than TV inpainting which preserves the intensity jumps or edges better. Harmonic inpainting is also much faster to compute. A five point stencil finite differencing discretizes the Laplace equation to a linear algebra problem $A\,u = b$, where b stores the known values of the image function at the boundary of Ω, A is a banded matrix with irregular off-diagonal entries reflecting the irregular $\partial\Omega$. The harmonic inpainting blurs the background regions (trees and grass). The right panel of Fig. 4 shows the effects of our proposed content adaptive pre-processing, after which more than half of the un-interesting SIFT feature points are removed for more efficient SIFT building matching. We remark that though the simple strategy of turning the tree regions blank also removes the unimportant features, it can introduce new feature points and cause mis-matches at the boundaries of tree regions which tend to be irregular. Our numerical tests suggest that inpainting is a better solution.

4 Feature Matching and Test Results

4.1 Matching

Matching of feature points is carried out as in SIFT which is an Euclidean-distance nearest neighbor approach in the descriptor space. A pair of feature points is considered matched if the distance ratio between the nearest neighbor distance and a second nearest neighbor distance is below τ,

$$\frac{d(p, p_{1st})}{d(p, p_{2nd})} < \tau \ ,$$

where $p \in \mathbb{R}^n$ is the descriptor to be matched and d_{1st} and d_{2nd} are the nearest and second nearest descriptors respectively, with d denoting the Euclidean distance. The $\tau = 0.8$ is suggested in [6]. This measure performs well because correct matches need to have the nearest neighbor significantly closer than the nearest incorrect match to achieve reliable matching. For false matches, there will likely be a number of other false matches within similar distances due to the high dimensionality of the descriptor space. We can think of the second nearest match as providing an estimate of the density of false matches within this portion of the feature space and at the same time identifying specific instances of feature ambiguity.

4.2 Examples of Image Matching

Here we show two experiments with images containing buildings and trees. Each of the tested images contains 800×600 pixels. In the first experiment, we match two near views of the same scene in Fig. 4. In Fig. 5, the left plot shows the matching of the SIFT feature points of the two original images, the right plot is the SIFT matching of reduced features in the pre-processed images. Computation is performed on a Dell laptop with 6G RAM and 1.6 GHz i7 CPU. The cpu time for the left plot is 47.16 seconds, and it is 39.09 seconds for the right plot. The the cpu time reported here and below included preprocessing. The matching rate for the left plot is 53.80 % (333 correct matches, 286 false matches), it is 67.08 % for the right plot (218 correct matches, 107 false matches). In the second experiment, we compare SIFT feature matching on two images with and without preprocessing, where one image is a near view of a farm house, and the other

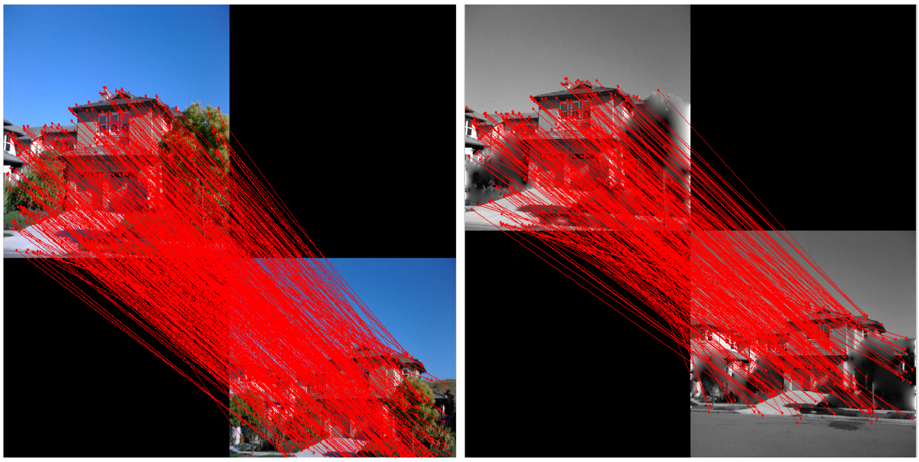

Fig. 5. Image matching of SIFT features with and without preprocessing. Left panel is the matching between two original images without segmentation and inpainting (matching rate 53.80 %). Right panel is the matching between two pre-processed images with trees regions removed (matching rate 67.08 %) and 17.11 % total cpu saving.

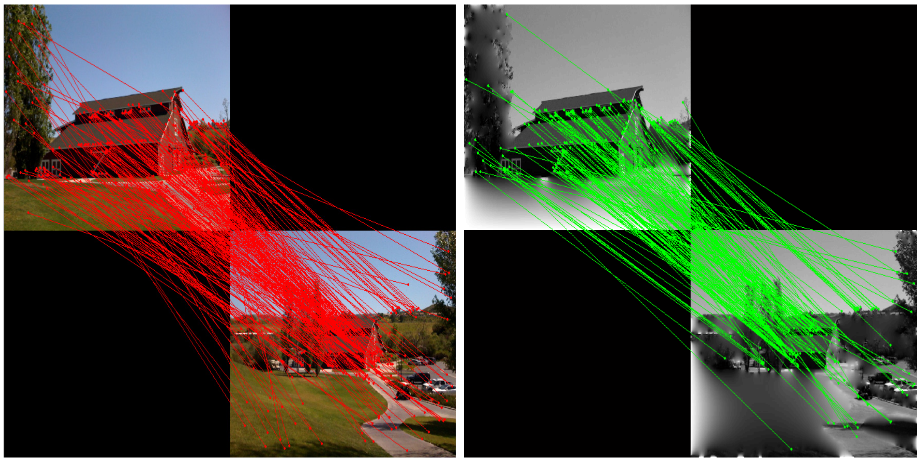

Fig. 6. Near (upper left) and far (lower right) views of a farm house in each panel. The left panel shows matching of SIFT features on the original images, with matching rate 28.73 % (77 correct matches, 191 false matches). The right panel shows SIFT feature matching on preprocessed images, with matching rate 37.17 % (71 correct matches, 120 false matches) and 29.51% total cpu saving.

is a far view. The far view contains more objects in the environment, such as additional trees occluding part of the building, and cars in the parking lot. Near (upper left) and far (lower right) views of a farm house are shown in Fig. 6. The left panel shows matching of SIFT features on the original images, with matching rate 28.73 % (77 correct matches, 191 false matches), and cpu time is 57.30 seconds. The right panel shows SIFT feature matching on the preprocessed images, with matching rate 37.17 % (71 correct matches, 120 false matches), and cpu time is 40.39 seconds. The improvement is consistent across 20 other images of landscaped buildings from street scenes in residential and commercial areas as long as the tree/bush/grass regions occupy roughly the same percentage of pixels in the images.

5 Concluding Remarks

We introduced a novel and efficient color-entropy segmentation for content adaptive image matching of natural building images. Experimental results showed a robust increase in matching rate by approximately 10% and decrease in cpu time (from 20 % to 30%) with pre-processing time included. A future line of work will study how to avoid modifying the images. Instead, one may design a weighting function to adapt the matching score. The weighting function is negatively correlated with the entropy-color measure and is essentially supported away from the tree regions. We also plan to study content adaptive preprocessing for more recent versions of corner detectors [7], and perform more extensive experiments on landscaped building images preferably on a public database.

The work was partially supported by NSF grant DMS-0911277.

References

1. Bertalmío, M., Sapiro, G., Caselles, V., Ballester, C.: Image inpaiting. In: Proc. of SIGGRAPH 2000, New Orleans, USA, pp. 417–424 (July 2000)
2. Chan, T., Shen, J.: Mathematical Models for Local Nontexture Inpaintings. SIAM J. Appl. Math. 62, 1019–1043 (2002)
3. Geman, G., Geman, D.: Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. IEEE Trans. Pattern Anal. Machine Intell. 6, 721–741 (1984)
4. L*a*b color space: `http://en.wikipedia.org/wiki/Lab_color_space`
5. Lowe, D.: Object recognition from local scale-invariant features. In: Proceedings of the International Conference on Computer Vision, vol. 2, pp. 1150–157 (1999)
6. Lowe, D.: Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision 60(2), 91–110 (2004)
7. Rosten, E., Porter, P., Drummond, T.: Faster and Better: A Machine Learning Approach to Corner Detection. IEEE Transactions on Pattern Analysis and Machine Intelligence 32(1), 105–119 (2010)