Designing Real-Time Multimedia Applications on Mobile Devices

Jiang Gao
Nokia Corporation
200 Mathilda Ave, Sunnyvale, CA 94086
jiang.gao@nokia.com

ABSTRACT

This paper explores the challenge of real-time image search and registration on mobile devices. We propose using the histogram of oriented gradients (HOG) features for characterizing image regions, and propose an algorithm based on the entropy of HOG to select "good" regions for image matching and registration. We also propose a novel implementation of a dual-mode mobile system based on a hybrid tracking and visual matching algorithm. We apply the algorithms to several mobile applications, including image matching for mobile visual search and panorama on mobile phones. The effectiveness of our approach is demonstrated on a large dataset.

Categories and Subject Descriptors

I.4.9 [**Image Processing and Computer Vision**]: Applications; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

General Terms

Algorithms, Performance, Design, Experimentation

Keywords

Image search; tracking; feature selection; registration

1. INTRODUCTION

Image recognition and registration are among the most commonly used technologies in mobile multimedia applications. With the introduction of robust local image features, both the accuracy and robustness of image recognition and registration have been improved greatly. However, these features are expensive to extract, and are too slow for real-time applications on a mobile device. For example, the popular MSER/SIFT algorithm extracts feature locations based on maximally stable extremal regions (MSER), and calculate a rich feature descriptor provided by SIFT[11]. However, the computation is too challenging for a mobile device. Furthemore, being characterization of local signatures, the feature descriptors can be locally ambiguous. For certain image regions, this problem becomes more prominent.

To make the local robust feature-based approaches more effective and working on mobile devices, we propose using histogram of oriented gradients (HOG) features to characterize image regions. We propose an algorithm based on the entropy of HOG to select "good" regions for image matching and registration. Based on the characterization of regions, we can fine tune the feature selection and manage the feature distributions in different regions, so that image registration and matching can become much more efficient and at the same time more reliable.

By applying the region characterization algorithm, we are able to select regions that contribute the most to image recognition, while filtering out regions either featureless or with ambiguous image features. In those regions, the expensive robust local image features are not extracted at all.

Our region categorization approach is different from texture classification. The most popular texture classification algorithms are based on techniques such as MRF modeling [12], or the integration of multi-scale filter outputs [10]. They usually have high computational complexity, which makes real-time applications difficult.

We also present a hybrid tracking and image matching algorithm that makes several multimedia applications feasible on a mobile platform. Tracking is utilized to select images not only for recognition,

MobiHeld '11, October 23, 2011, Cascais, Portugal.

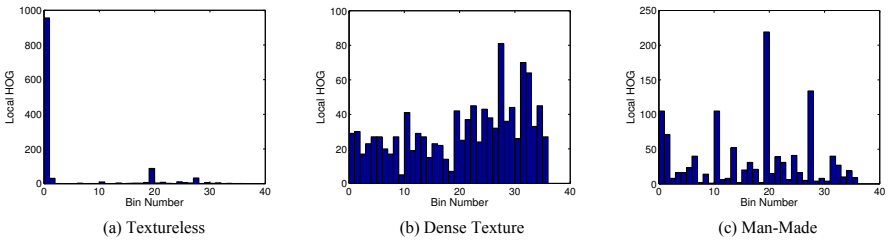

Figure 1: The typical orientation histogram of (a) A textureless image region; (b) One region with dense textures in it; (c) One image region with man-made structures in it. "Sparseness" is a character of the histogram features in (a) and (c).

Figure 2: A feature selection example. (a)-(b) Results of classified regions with dense textures (masked in green). Note these regions mostly correspond to trees /leaves in the images. (c) The originally detected local features. (d) The remaining features after applying the region-sensitive feature selection.

but also for image registration and panorama stitching. It also provides a smooth user interface.

We discuss two mobile applications that benefit from the proposed algorithms. First, we developed a pilot visual search system. Based on the GPS/Cell ID information provided by the phones, the visual database for location-based point-of-interests (POI) is streamed to the phone. The algorithm on the phone matches the input images to the visual database, and shows the results to users. Previous works in this respect include [4, 6]. Next, we developed a panorama system on mobile phones. Our system uses a feature-based approach. In this application, not only the region selection algorithm is applied, we also apply the feature selection algorithm, so that an optimally distributed feature set is selected for more reliable image registration.

2. REGION CATEGORIZATION

To design an algorithm to select robust local image features, we start by looking for what types of regions do not contribute much to visual matching and registration. First, we can filter out regions with dense textures, for example, regions with tree leaves, grasses, ripples on the water, etc. Second, many of the textureless regions are also featureless, thus they are not good regions for extracting local features that characterize images. Examples include the sky, large patches of textureless regions on rocks or walls, etc. Third, regions which do not fall into the above two types are usually more useful for image matching and registration. Good examples are regions with man-made structures in a typical city environment. These regions usually have more consistent local gradient directions. Just think about architectural structures.

We propose to characterize the above three categories of regions by multiple directional edges in local regions. The orientation histogram [5] or histogram of oriented gradients (HOG) features [3] are direct measure of local edge orientations. With contrast normalization and orientation binning, HOG features are robust to illumination changes and other noises, and can be used as stable local signatures in images.

Fig. 1 shows a comparison of HOG features for different types of regions. Based on these analysis, we use the sparseness of HOG features as a reasonable measure to seperate different types of regions.

2.1 Entropy Thresholding

This algorithm classifies regions with dense textures based on the entropy of HOG, as defined in the following equations. The orientation histogram can be represented as

$$H(d) = \frac{1}{S} \sum_{(x,y)\in R} \delta\left(d - arctan(dy/dx)\right) \ , \qquad (1)$$

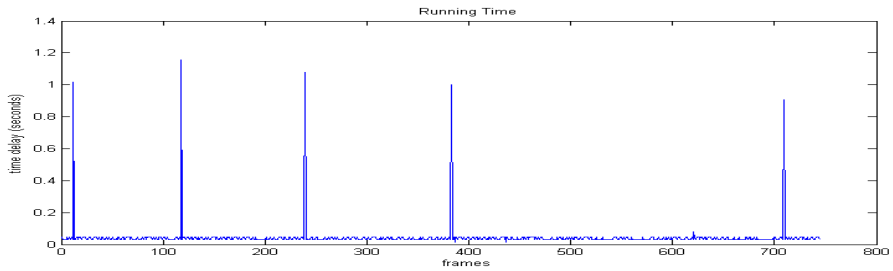

Figure 3: Running time of each frame for a typical video sequence. For most of the frames, the processing is very fast. These correspond to the tracking mode. For several of the frames, the processing times jumped to around 1 second. These are the frames the system performed image matching based on SURF features.

$$S = \sum_{(x,y)\in R} \mathbb{I}((x,y) \in R) \ , \tag{2}$$

where δ denotes the Kronecker function, $\mathbb{I}$ is the indicator function:

$$\mathbb{I}((x,y) \in R) = \begin{cases} 1 & (x,y) \in R, \\ 0 & (x,y) \notin R. \end{cases} \tag{3}$$

R is a subregion. $arctan(dy/dx)$ is the local orientation at pixel (x, y). We measure the orientation histogram using 36 bins. S is the normalization factor. The entropy is given by

$$Entropy(d) \quad = \quad -\sum_{d} [H(d) * log_2 H(d)] \ . \tag{4}$$

Entropies are calculated in each subregions. The subregions are classified as dense-textured with a threshold Thd:

$$Mask(x,y|(x,y) \in R) = \begin{cases} 1 & \text{if } Entropy(d) > Thd, \\ 0 & \text{else.} \end{cases} \tag{5}$$

We use a similar algorithm based on entrpy to classify textureless image regions.

3. FEATURE SELECTION

Based on the characterization of regions, we can manage the feature selection and distribution in subregions. This is different with the spatial-only feature selection where the overall feature distributions are balanced no matter what regions they fall in. The key new component of our approach is the region-sensitive feature selection algorithm.

3.1 Region-Sensitive Feature Selection

The goal of region-sensitive feature selection can be abstracted as in the following. Suppose we categorized the subregions $R_k, k = 1, 2, \cdots, K$ in the image I. Let $p(i, j)$ be an indicator: if there is a feature point at pixel (i, j), $p(i, j) = 1$; Otherwise, $p(i, j) = 0$. Let N be the total number of feature points in image I:

$$N = \sum_{k=1}^{K} \left[\sum_{(i,j)\in R_k} p(i,j) \right] \ , \tag{6}$$

then we want to limit the total number of feature points within the regions with dense textures:

$$\sum_{\substack{(i,j)\in I \\ Mask(i,j)=1}} p(i,j) < N_t, \ \ N_t << N \ . \tag{7}$$

$Mask(i, j) = 1$ denotes pixel (i, j) belongs to a region with dense textures.More specifically, the procedure is: first, all the features are sorted according to how strong they represent a local feature. Then, only the top N_t features within the dense-textured regions are selected.

Fig. 2 shows an example of region-sensitive feature selection for image registration. Fig. 2(a)-(b) shows the detected subregions with dense textures (in green). Fig. 2(c) shows the originally detected local features in a photo. Note that many of the strong features are in the trees/leaves areas. As shown in Fig. 2(d). After feature selection, the remaining features correspond to geometrically important structures in the photos, which result in more robust feature registration as well as faster convergence of the RANSAC algorithm.

4. TWO MODES OF THE SYSTEM

On a camera phone, the viewfinder inputs a continuous video stream with a low resolution typically of 320 by 240. When a user captures a full resolution photo, it corresponds to the same field-of-view of the viewfinder frame, but with a much higher resolution.

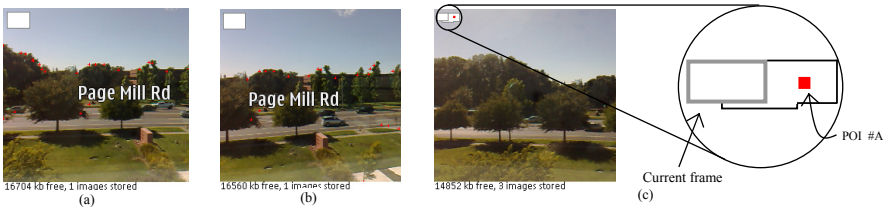

Figure 4: (a)-(b) On-device interface of the hybrid tracking and image matching system. The locations of the matched point-of-interest (POI) labels and feature points are adjusted using the motion parameters (translation, rotation, zoom) from the tracking module. (c) A screen shot showing the mini-map overlaid on the viewfinder screen. The previous key frames show up as rectangles in the mini-map, with their relatively locations preserved in the map. A new key frame is captured when there is enough new content in the current frame. The previous successful match (with a valid POI label) is marked by a red tag on the mini-map.

We developed an image processing pipeline which uses this viewfinder input to select viewfinder frames for image matching, and also estimates the overlapping regions for registration of image features. The key algorithm underlying these functionalities is the hybrid tracking and image matching algorithm.

4.1 Hybrid Tracking and Image Matching

Two modes: motion tracking and image matching co-exist in the system. The first mode uses a light-weight tracking algorithm. Image matching is based on robust local features, and is more computationally intensive [7].

As shown in Fig. 4, the tracking module estimates the motion model between the low-resolution viewfinder frames, and transforms and displays the previous recognition results to the next frame according to the motion model. The recognition results will be displayed even though image matching is not performed for each frame. Besides, the locations of matched frames (key frames) are displayed in a mini-map on screen.

When the tracking module decides that a large portion of the matched frame is out of the view, the system extracts and matches robust local features in the current viewfinder frame again.

The tracking algorithm essentially estimates the camera ego-motions on the mobile device. The problem of global motion estimation has been well studied in the past. The tracking algorithm in our system is an accelerated area-matching method, which extracts a set of point features from an image and performs local motion estimation on patches centered on these feature points. On a Nokia N95 mobile phone, which has a 330MHz ARM11 CPU, the tracking algorithm works at around 30fps.

Fig. 3 plots the timing result for a test sequence. It is obvious that by using the hybrid algorithm, the system is much more efficient, and can provide smoother output to the users, even though the image matching takes a perceivable period of time (around 1 second) for each frame.

5. APPLICATIONS

5.1 Image Search on A Mobile Phone

In a previous work we implemented the SURF algorithm [1] on mobile devices. It consists of three major steps, namely interest point extraction, repeatable angle computation and descriptor computation.

Using region categorization, if a local region is found to be textureless or with dense textures, SURF feature extraction is skipped for that region. This avoids extraction of expensive robust local features. We observed significant recognition speed up for most images, by using this strategy.

After feature extraction, the feature matching stage finds "similar" pairs from two sets of feature points, in terms of their descriptors. The spatial image transformation, which maps the input image to the template images, is inferred from two sets of coherent feature points using RANSAC. The speed and success rate of RANSAC also improve by using selected features based on region categorization.

By incorporating hybrid tracking and image matching, the system provides a much better user experience by showing the labeled results moving together with the previously recognized objects in each frame, even though the expensive image recognition is only performed for selected frames, as shown in Fig. 4.

5.2 Image Registration on Mobile Phones

We use a feature-based approach for panorama stitching on a mobile phone. The system detects Harris corner features [8]. Then, we apply region

Figure 5: Results on a test sequence captured by the phone pilot system. (a)-(b) The detected feature points for panorama stitching, without feature selections. (c)-(d) The classified subregions with dense textures. For panorama, only a limited number of strongest features are selected from these regions. For image matching, we do not extract SURF features from these regions. (e) The result of feature point registration. Note that very few or none of the corresponding features come from regions with dense textures, which are hard to match and also very expensive for geometric verification using RANSAC. (f)-(g) Stitching the images into a panorama. (g) Panorama image after blending, with recognized points of interest labelled.

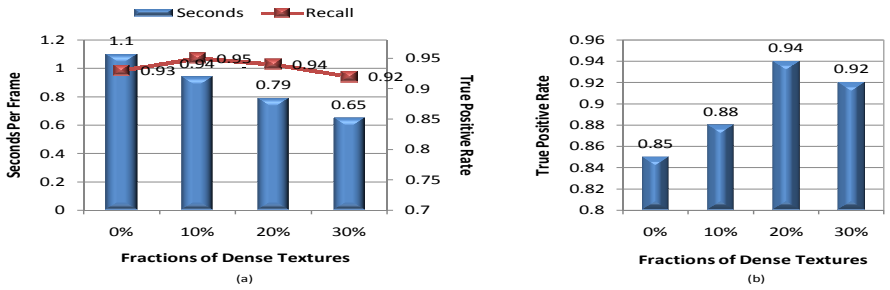

Figure 6: (a) Matching of images. The relationship between recall rates (with false positive rate less than 3 percent overall) and speed (Seconds-Per-Frame) with different proportions of subregions assigned as "unimportant" regions. (b) The true positive rates for correctly estimating the homography geometry for panorama stitching, with different proportions of subregions set as "unimportant" regions.

categorization and the feature selection strategy, as described in Section 2 and Section 3. Only selected features are registered and used to estimate geometric transforms between pictures. These are critical to not only speeding up the image registration process, but also eliminating most of the ambiguous image features, before establishing correspondence between them for photo registration.

The strategy of hybrid tracking and image matching is further extended to panorama stitching. For panorama, the tracker decides to capture a full resolution photo for stitching when the overlapping between current and previous captured frames exceeds a certain extend. Also for panorama, only the overlapping parts of the photos need to be analyzed. The boundaries of the overlapping parts are also estimated by the tracker. This is one of the major advantages of our system design that make the real-time performance possible.

The geometry of the photo stitching problem is well understood, and consists of estimating the 3×3 camera matrix for each image [9, 2]. Assuming the camera rotates around its optical center, the transformation H that the photos undergo can be modeled based on the three rotation angles $[\theta_x, \theta_y, \theta_z]$ and the focal length f. RANSAC is applied to find geometrically consistent inliers in the matched pairs, H_{ij} between images is then estimated by least squares fit with all the inliers.

Fig. 5 shows results from one of the outdoor se-

quences captured using our phone pilot system. Using region categorization, we are able to avoid expensive feature extraction and matching/registration in most of the textureless regions and regions with dense textures. As shown in Fig. 5(e), the most important image features for matching and registration seldom come from the textureless regions or regions with dense textures.

6. ACCURACY AND PERFORMANCE

Fig. 6 summarizes experimental results on image matching and panorama stitching by setting the proportions of "unimportant" subregions in our algorithm. In this case we are setting the proportion of subregion to be categorized as regions with dense textures. Textureless regions are classified automatically. We also set other parameters so that the system maintains the false positive rate of lower than 3 percent. In Fig. 6(a), we tested image matching using three groups of 200 test images each, with slightly different image backgrounds in each group.

Next, we tested our algorithm on image registration in panorama, using 15 image sequences captured by the phone pilot system. Fig. 6(b) shows the true positive rate for correctly estimating the homography geometry, which results in the correct stitching of the panoramas. Results based on region categorization by filtering out "unimportant" subregions have much higher true positive rates, partly due to the fact that our testing dataset includes many pictures with dense texture or noisy regions.

In respect of the performance, on a Nokia N95 mobile phone, the panorama stitching runs at an average of 0.3 second per frame, and SURF-based image matching runs at an average of 0.8 second per frame, which is fine for daily use.

7. CONCLUSIONS

We studied the categorization of image regions for image recognition and registration. Based on the study, we proposed using the entropy of HOG features to characterize subregions, and developed several novel algorithms for region categorization and feature selection.

We also demonstrated a hybrid tracking and image matching algorithm and a general image processing pipeline, which provides an efficient framework for multimedia applications on a mobile phone.

The proposed algorithms were applied to image matching and registration on camera phones. In most of the experiments, the proposed algorithms gave better results, and provided a smoother user experience. Further evaluation and studies, combining with a user study, will be our future work.

8. REFERENCES

[1] H. Bay, T. Tuytelaars, and L. V. Gool. Surf: Speeded up robust features. *ECCV*, pages 404–417, 2006.

[2] M. Brown and D. G. Lowe. Recognizing panoramas. In *Proceedings of the 9th International Conference on Computer Vision*, pages 1218–1225, 2003.

[3] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. *IEEE Conference Computer Vision and Pattern Recognition*, 1:886–893, 2005.

[4] P. Fockler, T. Zeidler, B. Brombach, E. Bruns, and O. Bimber. Phoneguide: Museum guidance supported by on-device object recognition on mobile phones. *Proceedings of the 4th International Conference on Mobile and Ubiquitous Multimedia*, pages 3–10, 2005.

[5] W. T. Freeman and M. Roth. Orientation histograms for hand gesture recognition. *International Workshop on Automatic Face and Gesture Recognition*, pages 296–301, 1995.

[6] G. Fritz, C. Seifert, and L. Paletta. Urban object recognition from informative local features. *IEEE International Conference on Robotics and Automation*, pages 132–138, 2005.

[7] J. Gao. Hybrid tracking and visual search. *Proceedings of ACM SIGMM International Conference on Multimedia*, pages 909–912, 2008.

[8] C. Harris and M. Stephen. A combined corner and edge detector. *Proc. 4th Alvey Vision Conference*, 1988.

[9] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[10] T. Leung and J. Malik. Representing and recognizing the visual appearance of materials using three-dimensional textons. *International Journal of Computer Vision*, 43(1):29–44, June 2001.

[11] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. *IEEE Conference Computer Vision and Pattern Recognition*, 2006.

[12] S. C. Zhu, Y. N. Wu, and D. B. Mumford. Filters, random field and maximum entropy (frame): Towards a unified theory for texture modeling. *International Journal of Computer Vision*, 27(2):107–126, March 1998.