Available online at http://www.journalijdr.com

IJDR

International Journal of
DEVELOPMENT RESEARCH

ISSN: 2230-9926

International Journal of Development Research
Vol. 07, Issue, 02, pp.11489-11498, February, 2017

Full Length Research Article

RETRIEVAL AND BLUR DETECTION OF IMAGES IN MOBILE DISPLAY DEVICES USING RECENT TECHNOLOGIES

***Srilekha, T. and Preethi, P.**

Department of Computer Science and Engineering, VSBCETC, Coimbatore 642109

ARTICLE INFO

Article History:

Received 19th November, 2016
Received in revised form
17th December, 2016
Accepted 24th January, 2017
Published online 28th February, 2017

Key Words:

Manifold algorithm,
Pixel separation,
Content based image retrieval,
Keyword Propagation,
Active learning.

ABSTRACT

Images of high (or) super resolution will be good to see. And having excellent clarity in all kinds of devices. Nowadays all mobile displays are not providing the users expecting clarity. Snaps taken from the mobile will be blurred (or) not clear sometimes. Reason for these problems are not using clarity lenses, shaking of mobile while taking snaps. For getting clarity in images first we should consider the pixels of the images. Pixels of an image should be very minute. If the image having very less spilt up in their pixels, then that image will look like blurred. So separation of pixels in images plays an important role here. For considering the pixels we should first use manifold algorithm. It split the images as pixels with good clarity and split as according to their texture also. In contrast to existing methods which train a binary classifier for each keyword, our keyword model is constructed in a straightforward manner by exploring the relationship among all images in the feature space in the learning stage. In relevance feedback, the feedback information can be naturally incorporated to refine the retrieval result by additional propagation processes. In order to speed up the convergence of the query concept, we adopt two active learning schemes to select images during relevance feedback. Furthermore, by means of keyword model update, the system can be self-improved constantly. The updating procedure can be performed on-line during relevance feedback without extra off-line training.

INTRODUCTION

Content-Based Image Retrieval (CBIR) is a long standing research problem in computer vision and information retrieval. Most of previous image retrieval techniques build on the assumption that the image space is Euclidean. However, in many cases, the image space might be a non-linear sub-manifold which is embedded in the ambient space. Intrinsically, there are two fundamental problems in image retrieval: 1) how do we represent an image? 2) How do we judge similarity? One possible solution to these two problems is to learn a mapping function from the low-level feature space to the high-level semantic space. The former is not always consistent with human perception while the latter is what image retrieval system desires to have. Specifically, if two images are semantically similar, then they are close to each other in semantic space. In this paper, our approach is to recover semantic structures hidden in the image feature space such as color, texture, etc. In recent years, much has been written about relevance feedback in content-based image retrieval from the perspective of machine learning, yet most learning methods only take into account current query session and the knowledge obtained from the past user interactions with the system is forgotten. To compare the effects of different learning techniques, a useful distinction can be made between short-term learning within a single query session and long-term learning over the course of many query sessions. Both short- and long-term learning processes are necessary for an image retrieval system though the former has been the primary focus of research so far. We present a long-term learning method which learns a radial basis function neural net-work for mapping the low-level image features to high level semantic features, based on user interactions in a relevance feedback driven query-by-example system. As we point out, the choice of the similarity measure is a deep question that lies at the core of image retrieval. In recent years, manifold learning has received lots of attention and been applied to face recognition, graphics, document representation, etc. These research efforts show that manifold structure is more powerful than Euclidean structure for data representation, even though there is no convincing evidence that such manifold structure is accurately present. Based on the assumption that the images reside on a low-dimensional sub manifold, a geometrically motivated relevance feedback scheme is proposed for image ranking, which is naturally

****Corresponding author: Srilekha, T.,***
Department of Computer Science and Engineering, VSBCETC, Coimbatore 642109.

conducted only on the image manifold in question rather than the total ambient space.

Relevance feedback on image manifold

In many cases, images may be visualized as points drawn on a low-dimensional manifold embedded in a high-dimensional Euclidean space. In this paper, our objective is to discover the image manifold by a locality-preserving mapping for image retrieval. We propose a geometrically motivated relevance feedback scheme for image ranking, which is conducted on the image manifold, rather than the total ambient space. 2.1 The Algorithm Let Ω denote the image database and R denote the set of query images and relevant images provided by the user. Our algorithm can be described as follows: 1. Candidate generation. For each image xi ∈ R, we find its knearest neighbors Ci = {y1, y2, …, yk}, yj∈Ω (those images in R are excluded from selection). Let C = C1 ∪ C2 ∪ … ∪ C|R|. We call C candidate image set. Note that R ∩ C = □. 2. Construct subgraph. Construct a graph G(V), where V=R ∪C.

where ε is a suitable constant. The choice of ε reflects our definition of locality. We put an edge between xi and xj if dist(xi, xj) ≠ ∞. Since the images in R are supposed to have some common semantics, we set their distances to zero. That is, dist(xi, xj) = 0, ∀xi, xj ∈ R. The constructed graph models the local geometrical structure of the image manifold. Distance measure on image manifold. To model the geodesic distances between all pairs of image points on the image manifold M, we find the shortest-path distances in the graph G. The length of a path in G is defined to be the sum of link weights along that path. We then compute the geodesic distance distG(xi, xj) (i.e. the shortest path length) between all pairs of vertices of i and j in G, using Floyd's O(|V| 3) algorithm. Retrieval based on geodesic distance. To retrieve the images most similar to the query, we simply sort them according to their geodesic distances to the query. The top N images are presented to the user. Update query example set. Add the relevant images provided by the user into R. Go back to step 1 until the user is satisfied.

Using manifold structure for image representation

In the previous section, we have described an algorithm to retrieve the user desired images by modeling the underlying geometrical structure of the image manifold. One problem of this algorithm is that, if the number of sample images is very small, then it is difficult to recover the image manifold. In this case, we propose a long-term learning approach to discover the true topology of the image manifold using user interactions. To be specific, we aim atmapping each image into a semantic space in which the distances between the images are consistent with human perception. The problem we are going to solve can be simply stated below:

Our proposed solution consists of three steps

- Inferring a semantic matrix Bm×m from user interactions, whose entries are the distances between pairs of images in semantic space T. m is the number of images in database.
- Find m points {z1, z2,…, zm}⊂ Rk which preserve pairwise distances specified in Bm×m. Laplacian eigenmaps [1] is used to find such an embedding. The space in which the m points {z1, z2,…, zm} are embedded is called LE semantic space in the rest of the paper. The user provided information is incorporated into the LE semantic space. Note that, the LE semantic space is only defined on the image database. In other words, for a new image outside the database, it is unclear how to evaluate its coordinates in the LE semantic space.
- Given m pair vectors, (xi , zi) (i = 1,2,…,m), where xi is the image representation in low-level feature space, and zi is the image representation in LE semantic space, train a radial basis function (RBF) neural network f that accurately predicts future z value given x. Hence f(x) is a semantic representation of x. The space obtained by f is called RBFNN semantic space. Note that, f(xi) ≈ zi. That is, RBFNN semantic space is an approximation of the LE semantic space. However, RBFNN semantic space is defined everywhere. That is, for any image (either inside or outside the database), its semantic representation can be obtained from the mapping function.

EXPERIMENTAL RESULTS

In this paper, we focus on image retrieval based on user's relevance feedback to improve the system's short-term and long-term performances. The user can submit a query image either inside or outside the database. The system first computes low-level features of the query image and then maps it into semantic space using the learned mapping function. The system retrieves and ranks the images in the database. Then, the user provides his judgment of the relevance of retrieval. With the user's relevance feedback, the system refines the search result iteratively until the user is satisfied. The accumulated relevance feedbacks are used to construct and update the semantic space, as described in Section 3. We performed several experiments to evaluate the effectiveness of our proposed approaches over a large image dataset. The image dataset we use consists of 3,000 images of 30 semantic categories from the Corel dataset. Each semantic category contains 100 images. The 3,000 images are divided into two subsets. The first subset consists of 2,700 images, and each semantic category contains 90 images.

The second subset consists of 300 images, and each semantic category contains 10 images. The first subset is used as training set for learning the optimal mapping function. The second subset is for evaluating the generalization capability of our learning framework. A retrieved image is considered correct if it belongs to the same category of the query image. Three types of color features (color histogram, color moment, color coherence) and three types of texture features (tamura coarseness histogram, tamura directionary, pyramid wavelet texture) are used in our system. The combined feature vector is 435-dimensional. We designed an automatic feedback scheme to model the short term retrieval process. We only require the user to provide positive examples. At each iteration, the system selects at most 5 correct images as positive examples (positive examples in the previous iterations are excluded from the selection). These automatic generated feedbacks are used as training data to perform short term learning. To model the long-term learning, we randomly select images from each category as the queries. For each query, a short-term learning process is performed and the feedbacks are used to construct the semantic space. The retrieval accuracy is defined as follows: N relevant images retrieved in top N returns Accuracy

= Four experiments are conducted. The experiment with the new retrieval algorithm on image manifold is discussed in Section 4.1. In Section 4.2, we show the image retrieval performance in the learned semantic spaces. The generalization capability is also evaluated. In Section 4.3 we further test the system's performance in semantic space with different dimensionalities. We compare our new algorithm with Rui's algorithm in semantic space 4.1 Retrieval on Image Manifold We compare the performance of our proposed retrieval algorithm on image manifold with the relevance feedback approach described in Rui. We didn't compare it to other image retrieval methods because our primary purpose is to analyze the geometrical structure of the image space. Specifically, we aim at comparing the Euclidean structure and manifold structure for data representation in image retrieval. The comparison was made in the lowlevel feature space with no semantic information involved. Figure 2 shows the experimental result by looking at the top 20 retrievals. As can be seen, our algorithm outperforms Rui's approach. One reason is that the image manifold is possibly highly nonlinear, while Rui's approach can only discover the linear structure.

Retrieval on Image Manifold

We compare the performance of our proposed retrieval algorithm on image manifold with the relevance feedback approach described in Rui. We didn't compare it to other image retrieval methods because our primary purpose is to analyze the geometrical structure of the image space. Specifically, we aim at comparing the Euclidean structure and manifold structure for data representation in image retrieval. The comparison was made in the low level feature space with no semantic information involved. As can be seen, our algorithm outperforms Rui's approach. One reason is that the image manifold is possibly highly nonlinear, while Rui's approach can only discover the linear structure.

MATERIALS AND METHODS

Mobile user guided adaptation system

To guarantee personalized media consumption with best possible perceptual experience in user-centric multimedia applications, both mobile device access environments and mobile user perceptual experiences are properly taken into consideration in this proposed scheme. The mobile environments include mobile device capabilities and mobile user interfaces, while mobile user perceptual experiences are highly affected by the semantics of media, user individual preference, and presentation of the adaptation results. Therefore, our target is to present the best possible adaptation results by preserving the quality of semantically important and user desired content under the limitation of mobile display capacities and interaction interfaces.

System Components

As shown in Fig. 1(a), the proposed system consists of 1) an adaptation proxy to process user request and feedback as well as to carry out semantic extraction, user preference learning, and adaptation; and 2) a server/database hosting original consumer photo content. We assume the annotation of the server side media content is processed offline while the user request and feedback processing is carried out in real time.

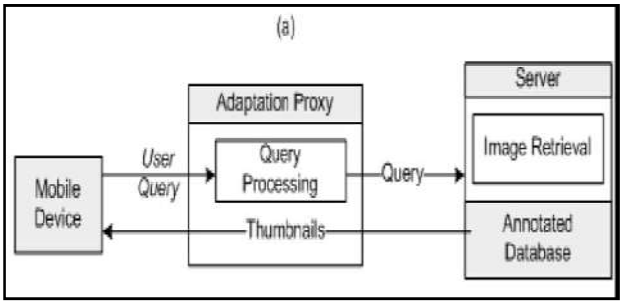


Fig. 1. Query Processing

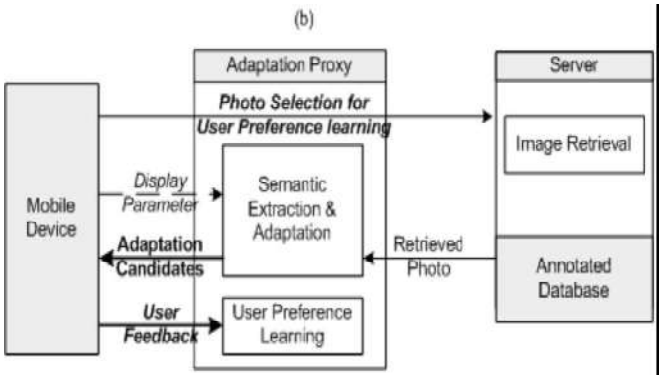


Fig. 2. Semantic Extraction and Adaptation Process

System Workflow

The proposed system works in the following manner. As shown in Fig. 1(b), a user first inputs the semantic request in the form of the keywords for the desired media content through the user interface at the mobile device. These semantic keywords represent the key semantic concept for the desired media content for retrieval and can be used to match the associated annotations representing the semantics of the media content in the database. Such a semantic request is then sent to the query processing module of the adaptation proxy. As most people would like to input the activities or events as the keywords, we assume the system takes queries in the form of events.

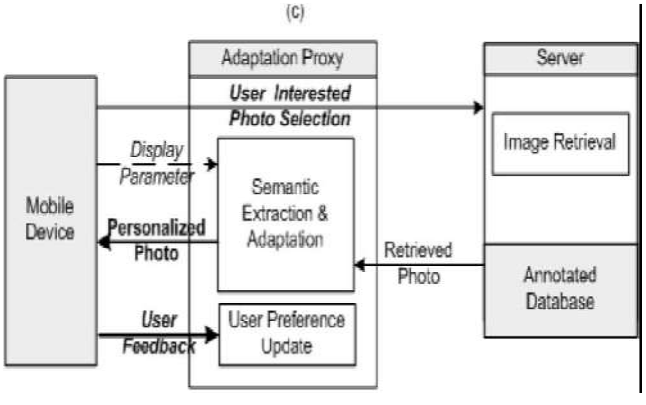


Fig. 3. Semantic Extraction and Adaption

User guided semantic extraction

Human tends to view and understand images in terms of people and objects associated with underlying concepts in the real world. Hence, semantic analysis is an indispensable step towards extracting semantically important objects and learning user preferences for proper content selection in image adaptation to improve mobile users' perceptual experiences. Semantic gap is still a big challenge in computer vision.

Fortunately, in our adaptation scenario, we do not need to perform a full semantic analysis for images, because for a given event, users tend to be interested in only a few objects. Hence, instead of carrying out full image semantic analysis, we extract key objects that users might desire. The key cue we can utilize to narrow down the semantic gap in our application is the user input queries. Although the limited mobile user interface usually does not allow very complicated input as query, the compact keywords provide simple yet useful information about the mobile users' intention. Although sometimes there is a departure between the mobile users' real intention and their query specification, the query is still informative enough to be utilized to extract semantically important objects as user preference relevant object candidates based on concept ontology. It is hence necessary to analyze the user supplied semantic request and determine related major semantic concepts in order to effectively extract semantically meaningful objects. Although the total number of semantic concepts will be numerous in the real world and in general image database, the concepts appearing in consumer domain contains only a small fraction of the general concepts. Moreover, it has been shown that most consumer photos are relevant to one of the events as defined. Therefore, we adopt these event definitions in the proposed system and focus on the semantic analysis utilizing the user provided event-based semantic keywords to extract the semantically important objects. The limited number of events also leads to an acceptable semantic analysis load.

Bottom-Up Low Level Feature Extraction

In the bottom-up approach, salient regions are generated based on low level features. First, a raw saliency map is calculated based on low level features such as intensity and color. Then, the image is segmented into regions to produce an efficient representation of the saliency maps for fusion with top-down semantic analysis. Finally, we represent the original image with regions of different saliency values by averaging the saliency within each region R and use the average value as the initial probability that the region belongs to the desired object.

User Guided Top-Down High Level Semantic Extraction

In the top-down approach, we develop the event specific classifier to obtain the high level features of the user interested objects with different semantic importance in the given events. We take SIFT [23] and color features as the event based high level semantic features. For each event, an event specific classifier is built offline on a training set that consists of photos of this event. The features for training include quantized color distribution of the 3-D histogram in HSI space and bag of words of quantized SIFT feature distributions in the training photos with known semantically important objects. Such generic high level semantic feature extraction scheme can be extended to any event by introducing corresponding training process. After the top-down training, the following probabilities are obtained: and, the probability of the SIFT word appears on the object and the non-object area, respectively; and, the conditional probability of the color bin on the object and non-object region, respectively. The training is carried out offline for each event and these probabilities will be used as conditional probability in the following Bayesian fusion module.

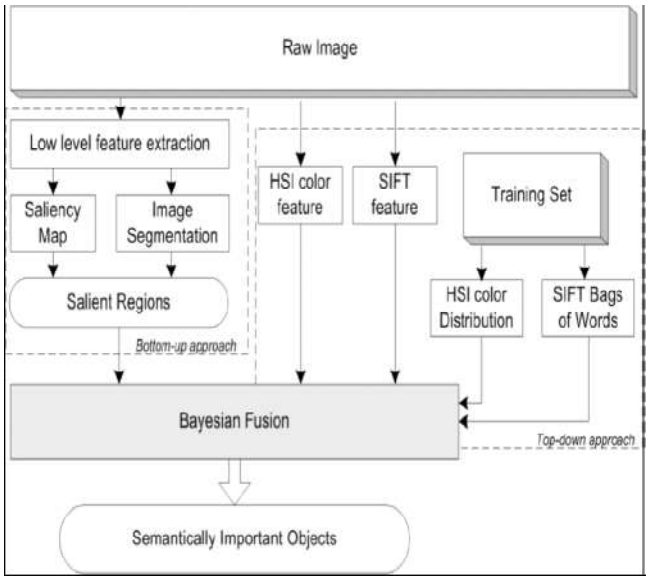


Fig. 4. Bayesian Fusion

Key Objects Localization by Bayesian Fusion

After obtaining the results of bottom-up salient regions and the top-down high level features of main objects using event specific classifier, we fuse them to find the semantically important objects that will likely match with the user's interests. The fusion is designed via Bayesian principle to obtain the posterior belief map of a class of semantic objects. For each region, we consider the bottom-up salient regions as *a priori* of the region belonging to the class of semantic objects. For the conditional probability, suppose the words are independent, the conditional probability of the region appears on the object class is the joint probability of all the words and colors in the region.

User Preference Learning for Adaptation

As discussed earlier, due to the limited mobile device user interfaces, it is usually not allowed to provide complicated inputs for mobile users to describe their desired content details very accurately. Moreover, because of mobile users' different background, they tend to have different interests in concepts even if they input the same query. Even for the same mobile user, his or her intention may vary at different time and circumstance. Therefore, to present content truly consistent with the mobile user's true interest, it is necessary to fine tune the importance of extracted objects and make them gradually matched to the mobile users' preference. However, mobile user preferences are subjective measures varied among individuals. To obtain the user subjective preference value (PV) upon different objects, the only way is to learn them from each individual user. In CBIR systems, to bridge the intention gap in retrieving more relevant images consistent with user's interests, relevance feedback techniques have been developed to capture the subjectivity of human perception of visual content by dynamically updating weights of different features based on the user's feedback. Weights of influential features will be tuned higher to retrieve more consistent images as user feedbacks their judgment towards previous retrieval results.

Feedback Process for PV Learning and Updating

Due to intrinsic characteristics of mobile device interfaces, simple and easy interaction schemes have to be designed for mobile users. Hence, for each query, the system only requires

the mobile user to grade a small batch of adaptation candidates initially and few adaptation results subsequently to learn the user's intention. For each grading, the user only needs to type a digit ranging from 1 to 9 to score whether it is consistent with his preference: 5 means no opinion; scores between 4 and 1 represent the degree of non-relevance, in which 4 is slightly non-relevant and 1 is highly non-relevant.

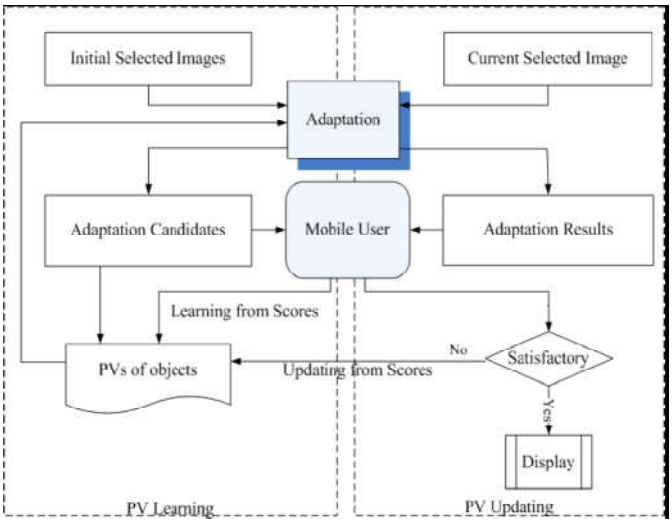
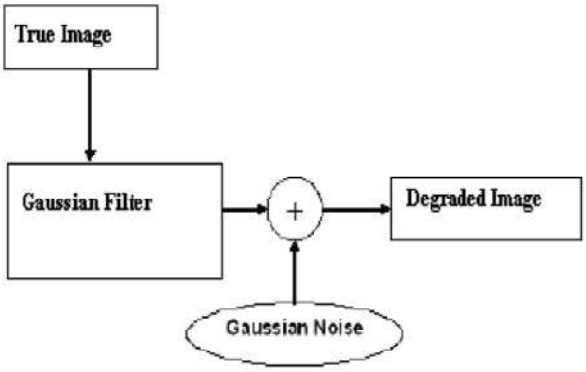


Fig. 5. Feedback process for PV learning and updating

User Preference Value Learning and Updating

To handle the variety of mobile user preference, PV learning is performed from the feedback upon the adaptation candidates for the initial selected four images to fine tune the importance of objects from their semantic importance inclined to the individual user preference values of objects. If the PVs learned are not consistent enough with the mobile user's intention, PV updating towards user preference will be further performed from the feedback upon subsequently selected images. Through a limited number of interactions, the PVs can converge to the mobile user intention and the system will be able to perform relevant adaptation for more images, since in one query, the user usually shows stable and coherent interests.



Degradation model for blurring image

In degradation model for blurring image, the image is blurred using filters and an additive noise. The image can be degraded done by using Gaussian Filter and Gaussian Noise. Gaussian Filter represents the Point Spread Function which is a blurring function. The degraded image can be express by the equation f = g * h + n; Where * is the convolution operator, g is the clear image to recover, f is the observed blurred image, h is the blur kernel (or point spread function) and n is the noise. The below Fig.1 represents the formation of degradation model. Image deblurring can be performed by the technique, Gaussian Blur. They are the convolution of the image with 2-D Gaussian function.

Gaussian filter

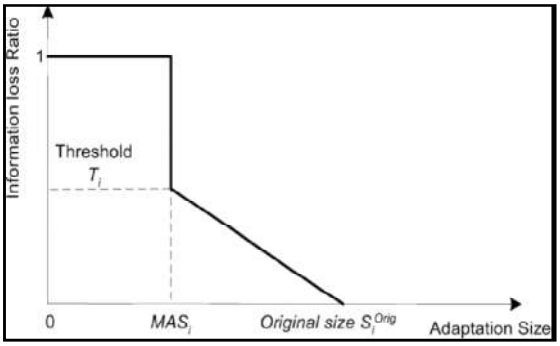
Gaussian filter is useful for blur an image by Gaussian function. It requires two specifications such as mean and variance. They are weighted blurring. Gaussian function is of the following form where σ is variance and x and y are the distance from the origin in the x axis and y axis respectively.

Overall Architecture And Deblurring Algorithm

The original image is degraded or blurred using degradation model to produce the blurred image. The blurred image should be an input to the deblurring algorithm. Various algorithms are available for deblurring. In this paper, we are going to use blind deconvolution algorithm. The result of this algorithm produces the deblurring image which can be compared with our original image. This algorithm can be achieved based on MLE (Maximum Likelihood Estimation).

User Centric Semantic Adaptation

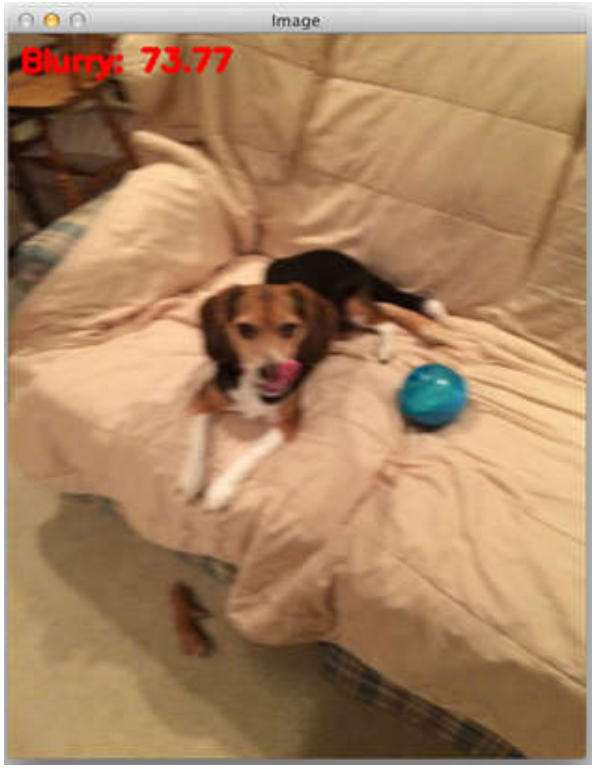
Given the key semantically meaningful objects contained in a consumer photo, the relevance of different objects for a given user preference can be varied. Moreover, the relevance of the same object for different mobile users may vary substantially Furthermore, there are a variety of mobile devices with various capacities for different users. Given these different PVs of different objects for various mobile users as well as the variety of mobile display capacities, the adaptation module has to decide what content to adopt adaptively according to these varying conditions. The goal of user centric adaptation is to simultaneously panelize the selection of contents not preferred by the user and preserve the user preferred objects with high quality depending on the degree of their relevance to user, under the limited mobile display constraints. By the integration of OSI and feedback, we have already obtained such relevance of different objects to different mobile users which are denoted as PVs. In the following step, we utilize the PVs of objects to guide the adaptation to provide the mobile user the best possible perceptual experience. The optimal adaptation is performed and presented by formulating it into an information fidelity (IF) maximization problem as discussed below.

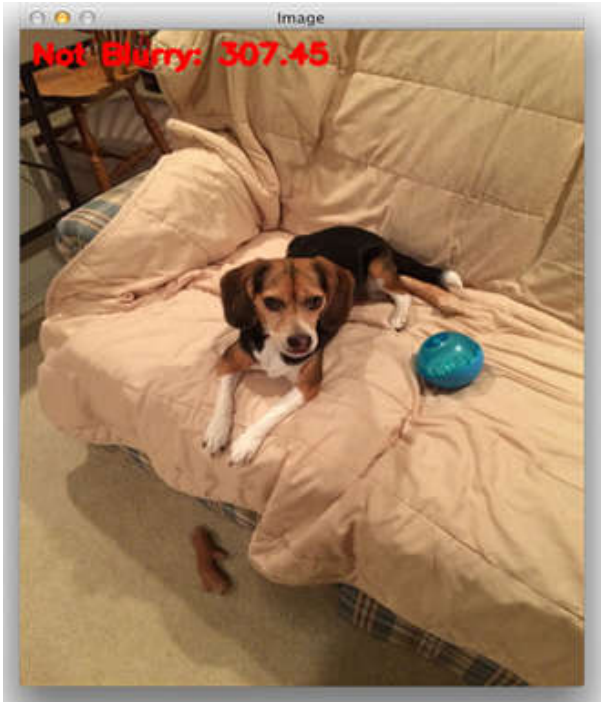


Experiments

Blur detection with Open CV Shell

The subsequent image in the dataset is marked as "blurry". However, this image contains dramatic amounts of blur.



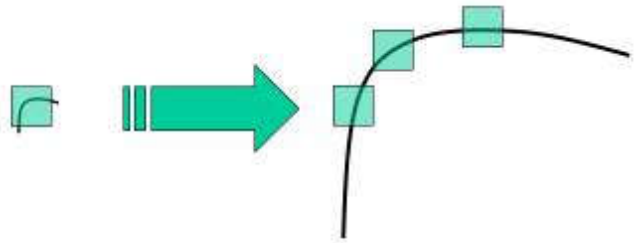


Detecting the amount of blur in an image using the variance of Laplacian

We implemented the *variance* of *Laplacian* method to give us a single floating point value to represent the "blurriness" of an image. This method is fast, simple, and easy to apply — we simply convolve our input image with the Laplacian operator and compute the variance. If the variance falls below a predefined threshold, we mark the image as "blurry". It's important to note that threshold is a critical parameter to tune correctly and you'll often need to tune it on a per-dataset basis. Too small of a value, and you'll accidentally mark images as blurry when they are not. With too large of a threshold, you'll mark images as non-blurry when in fact they are.

SIFT (Scale-Invariant Feature Transform)

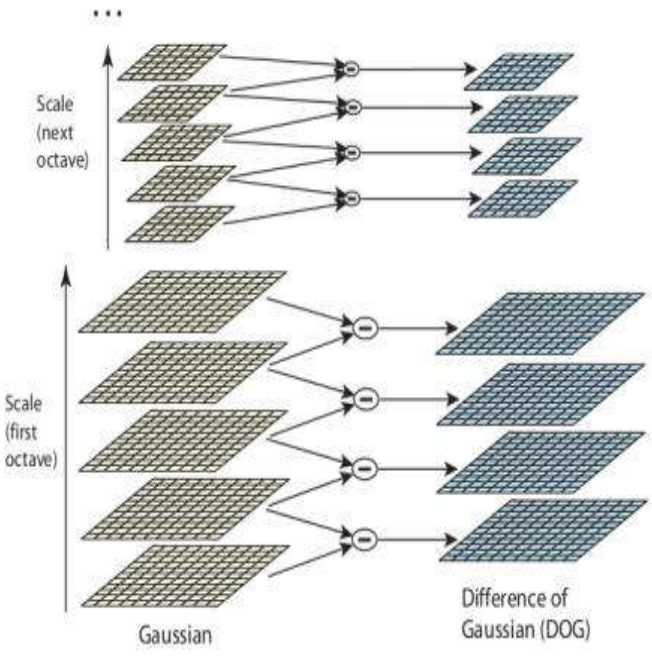
They are rotation-invariant, which means, even if the image is rotated, we can find the same corners. It is obvious because corners remain corners in rotated image also. But what about scaling? A corner may not be a corner if the image is scaled. For example, check a simple image below. A corner in a small image within a small window is flat when it is zoomed in the same window. So Harris corner is not scale invariant.



Scale-space Extrema Detection

From the image above, it is obvious that we can't use the same window to detect keypoints with different scale. It is OK with small corner. But to detect larger corners we need larger windows. For this, scale-space filtering is used. In it, Laplacian of Gaussian is found for the image with various σ values. LoG acts as a blob detector which detects blobs in various sizes due to change in σ. In short, σ acts as a scaling parameter. For eg, in the above image, gaussian kernel with low σ gives high value for small corner while guassian kernel with high σ fits well for larger corner. So, we can find the local maxima across the scale and space which gives us a list of (x,y,σ) values which means there is a potential keypoint at (x,y) at σ scale. But this LoG is a little costly, so SIFT algorithm uses Difference of Gaussians which is an approximation of LoG. Difference of Gaussian is obtained as the difference of Gaussian blurring of an image with two different σ, let it be σ and kσ. This process is done for different octaves of the image in Gaussian Pyramid.

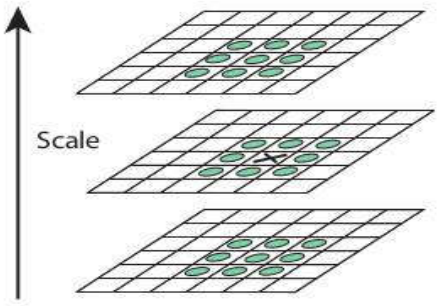
It is represented in below image:



Once this DoG are found, images are searched for local extrema over scale and space. For eg, one pixel in an image is compared with its 8 neighbours as well as 9 pixels in next scale and 9 pixels in previous scales. If it is a local extrema, it is a potential keypoint. It basically means that keypoint is best represented in that scale. It is shown in below image

Key point Localization

Once potential keypoints locations are found, they have to be refined to get more accurate results. They used Taylor series expansion of scale space to get more accurate location of extrema, and if the intensity at this extrema is less than a threshold value (0.03 as per the paper), it is rejected. This threshold is called contrast.



Threshold in OpenCV. DoG has higher response for edges, so edges also need to be removed. For this, a concept similar to Harris corner detector is used. They used a 2x2 Hessian matrix (H) to compute the pricipal curvature. We know from Harris corner detector that for edges, one eigen value is larger than the other. So here they used a simple function, If this ratio is greater than a threshold, called edgeThreshold in OpenCV, that keypoint is discarded. It is given as 10 in paper. So it eliminates any low-contrast keypoints and edge keypoints and what remains is strong interest points.

Orientation Assignment

Now an orientation is assigned to each keypoint to achieve invariance to image rotation. A neigbourhood is taken around the keypoint location depending on the scale, and the gradient magnitude and direction is calculated in that region. An orientation histogram with 36 bins covering 360 degrees is created. (It is weighted by gradient magnitude and gaussian-weighted circular window with σ equal to 1.5 times the scale of keypoint. The highest peak in the histogram is taken and any peak above 80% of it is also considered to calculate the orientation. It creates keypoints with same location and scale, but different directions. It contribute to stability of matching.

Keypoint Descriptor

Now keypoint descriptor is created. A 16x16 neighbourhood around the keypoint is taken. It is devided into 16 sub-blocks of 4x4 size. For each sub-block, 8 bin orientation histogram is created. So a total of 128 bin values are available. It is represented as a vector to form keypoint descriptor. In addition to this, several measures are taken to achieve robustness against illumination changes, rotation etc.

Keypoint Matching

Keypoints between two images are matched by identifying their nearest neighbours. But in some cases, the second closest-match may be very near to the first. It may happen due to noise or some other reasons. In that case, ratio of closest-distance to second-closest distance is taken. If it is greater than 0.8, they are rejected. It eliminaters around 90% of false matches while discards only 5% correct matches, as per the paper. So this is a summary of SIFT algorithm. For more details and understanding, reading the original paper is highly recommended. Remember one thing, this algorithm is patented.

SIFT in OpenCV

So now let's see SIFT functionalities available in OpenCV. Let's start with keypoint detection and draw them. First we have to construct a SIFT object. We can pass different parameters to it which are optional and they are well explained in docs. sift.detect() function finds the keypoint in the images. You can pass a mask if you want to search only a part of image. Each keypoint is a special structure which has many attributes like its (x,y) coordinates, size of the meaningful neighbourhood, angle which specifies its orientation, response that specifies strength of keypoints etc. opencv also provides cv2.drawkeypoints() function which draws the small circles on the locations of keypoints. if you pass a flag, cv2.draw_matches_flags_draw_rich_keypoints to it, it will draw a circle with size of keypoint and it will even show its orientation.

Now to calculate the descriptor, OpenCV provides two methods.

- Since you already found keypoints, you can call sift.compute() which computes the descriptors from the keypoints we have found. Eg: kp,des = sift.compute(gray,kp)
- If you didn't find keypoints, directly find keypoints and descriptors in a single step with the function, sift.detectAndCompute().

We will see the second method:

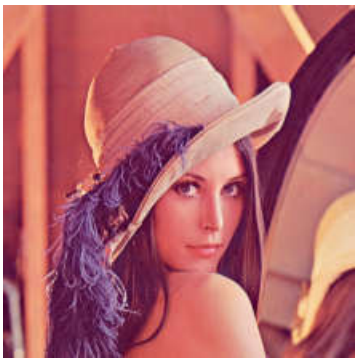
```
sift = cv2.xfeatures2d.SIFT_create()
kp, des = sift.detectAndCompute(gray,None)
```

Here kp will be a list of keypoints and des is a numpy array of shape Number_of_Keypoints×128.

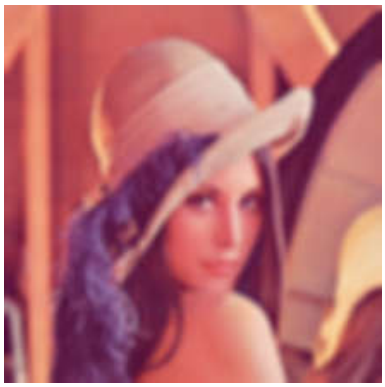
So we got keypoints, descriptors etc. Now we want to see how to match keypoints in different images.

RESULTS

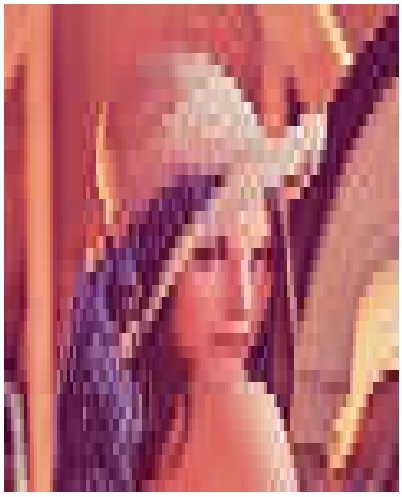
Original Image



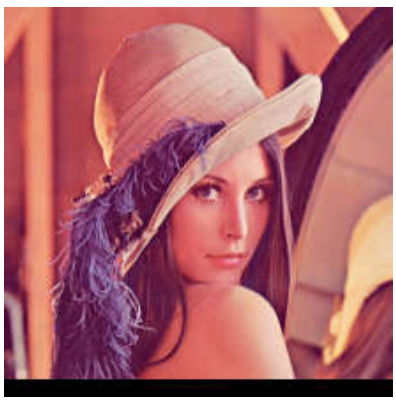
Blur image



Resized image



Shift image



Coding

```
#include <opencv2/core.hpp>
#include <opencv2/core/ocl.hpp>
#include <opencv2/highgui.hpp>
#include <opencv2/img_hash.hpp>
#include <opencv2/imgproc.hpp>
#include <iostream>
void compute(cv::Ptr<cv::img_hash::ImgHashBase> algo)
{ auto input = cv::imread("lena.png");
   cv::Mat similar_img;
   //detect similiar image after blur attack
   cv::GaussianBlur(input, similar_img, {7,7}, 2, 2);
   cv::imwrite("lena_blur.png", similar_img);
   cv::Mat hash_input, hash_similar;
   algo->compute(input, hash_input);
   algo->compute(similar_img, hash_similar);
   std::cout<<"gaussian blur attack : "<<
            algo->compare(hash_input,
hash_similar)<<std::endl;
   //detect similar image after shift attack
   similar_img.setTo(0);
   input(cv::Rect(0,10, input.cols,input.rows-10)).
   copyTo(similar_img(cv::Rect(0,0,input.cols,input.rows-
10)));
   cv::imwrite("lena_shift.png", similar_img);
   algo->compute(similar_img, hash_similar);
   std::cout<<"shift attack : "<<
            algo->compare(hash_input,
hash_similar)<<std::endl;
   //detect similar image after resize
   cv::resize(input, similar_img, {120, 40});
   cv::imwrite("lena_resize.png", similar_img);
   algo->compute(similar_img, hash_similar);
   std::cout<<"resize attack : "<<
            algo->compare(hash_input,
hash_similar)<<std::endl;
}
int main()
{
   using namespace cv::img_hash;
   //disable opencl acceleration may(or may not) boost up
speed of img_hash
   cv::ocl::setUseOpenCL(false);
   //if the value after compare <= 8, that means the images
   //very similar to each other
   compute(ColorMomentHash::create());
   //there are other algorithms you can try out
   //every algorithms have their pros and cons
   compute(AverageHash::create());
   compute(PHash::create());
   compute(MarrHildrethHash::create());
   compute(RadialVarianceHash::create());
   //BlockMeanHash support mode 0 and mode 1, they
associate to
   //mode 1 and mode 2 of PHash library
   compute(BlockMeanHash::create(0));
   compute(BlockMeanHash::create(1));
}
```

DISCUSSION

In the above method first the image will be retrieved from the particular snap (or) album. That image will be correctly retrieved using manifold ranking of blocks.

In that method the corresponding image will be split as number of pixels based on the color and texture. After that the image will be going under the Bayesian fusion algorithm for entire clearance of an image. This algorithm makes the image to be clear when we zoom that in mobile display devices. In many of the mobile displays, images will not be clear while zooming. For that we are SIFT and Open CV methods for detecting blur image from the snap. Open CV method detecting and classifying the blur image for making clearance. Open CV method first considers the original image. Then that image will be resized for detecting. Generally the image will be resized for detecting the blur part easily. Finally the blur part will be removed successfully. These all methods provides clear and unblurred image totally and even while zooming an image. So both operations are carried out here. Those are images will be looking clear when we zoom. And blur part also removed. Finally the image will be displayed with excellent clarity normally as well as while zooming. This method having one drawback. We can get the clear image after zooming that. But while zooming, the image will not be clear. It will be looking like shaky image. After zooming only we can get the image with high resolution. So research is based on while we zooming also the image should be looking clear. While zooming we will move the particular of the image up and down, left and right.

At that exact time the image will not be looking clear. So the main aim is to bring the clarity during the movement of an image while zooming.

REFERENCES

Suh, B., Ling, H. Bederson, B. B. and Jacobs, D. W. 2003. "Automatic thumbnail cropping and its effectiveness," in *Proc. ACMSymp. User Interface Software and Technology*, pp. 95–104.

Chen, L. Q., Xie, X., Fan, X., Ma, W. Y., Zhang, H. J. and H. Q. Zhou, 2003. "A visual attention model for adapting image on small displays," *ACM Multimedia Syst. J.*, vol. 9, no. 4, pp. 353–364, Oct. 2003.

Liu, H., Jiang, S., Huang, Q., Xu, C., and Gao, W. 2007. "Region-based visual attention analysis with its application in image browsing on small displays," *ACM Multimedia*, pp. 305–308.

Lagendijk, R. L. 2000. Basic Method for Image Restoration and identification, Academic Press.

Mallat, S. and Hwang, W.L. 1992. "Singularity Detection and Processing with Wavelet," IEEE Trans. On Information Theory, March, pp.617-643.

Marichal, X., Ma, W.Y. and Zhang, H.J. "Blur Determination in the Compressed Domain Using DCT Information," Proceedings of the IEEE ICIP'99, pp. 386-390.

Sheppard, K., Marcellin, D.G., Marcellin, M.W. and Hunt, B.R. 2001. "Blur identification from vector quantizer encoder distortion," IEEE Trans. On Image Processing, March, pp. 465-470.

Pavlovic, G. and Tekalp, A.M. 1992. "Maximum likelihood parametric blur identification based on a continuous spatial domain model," IEEE Trans. On Image Processing, October 1992, pp. 496-504.

****