

# Content Based Image Retrieval (CBIR) Using Segmentation Process

R.Gnanaraja<sup>1</sup>, B. Jagadishkumar<sup>2</sup>, S.T. Premkumar<sup>3</sup>, B. Sunil kumar<sup>4</sup>

PG Scholar, Department of Computer Science and Engineering, Adithya Institute of Technology, Coimbatore, India<sup>1,2,3,4</sup>

**Abstract---** Mining of Structured representations in content based image retrieval is a popular research topic in many useful applications. In the last decade there has been an explosion of interest in mining time series data. Literally hundreds of papers have introduced new algorithms to index, classify, cluster and segment time series. The initial work focused mainly on values with tags, while most of the recent development focuses on discovering association rule among tree structured data objects to preserve the structural information. In this paper we combined the techniques of texture based segmentation algorithm, Blob reduction by identifying outlier's detection, SIFT algorithm to an automatic system to annotate and retrieve images. This paper tend to reveal a good behaviour in classification of our graph based solution on two publicly available databases and produce the images features with more enhancement by an efficient segmentation algorithm.

**Keywords-** Scale invariant feature transform (SIFT), Blob Reduction and Extraction, Segmentation, Global method, key point descriptor.

## I. INTRODUCTION

## II. OVERVIEW

Texture based segmentation is a topic where a lot of different approaches lead to more or less satisfying results. In general all of them try to match a particular feature or a feature vector which describes the analysed region. Subsequently a threshold or threshold vector is applied and a texture class is assigned to the region. This paper describes how data mining algorithms can be used advantageously for texture based segmentation. Using a reference image with Known texture, a model for a classier is trained.

That is applied to image regions of unknown texture. For the data mining it is necessary to calculate

many different features and rate them (e.g. by their information gain or correlation) accordingly. Only the best features selected this way are used to train a classifier, which is then used to segment subsequent images. Using this selected classifier, it is possible to determine the location where a texture occurs in the image.

The performance of the classifier is demonstrated for synthetic test images and the problem of detection of scratches on a metal sheet under inhomogeneous illumination. In this example only two reference images are classified manually to train the classifier and the rest is done automatically. So, no additional parameters or thresholds must be set for the scratch detection problem analysed. The process of image segmentation can be helped/enhanced for many applications if there is some application dependent domain knowledge that can be used in the process. Image retrieval has been an active research area over the last decades.

Most researchers provide an extensive description of image archives, various indexing methods and common searching tasks, using different techniques and technologies. Currently CBIR techniques can be classified into two categories: Global approach by using global visual features to describe images and Local approach by considering images as the combination of multiple objects, key points or regions.

The following flow chart explains the steps involved in the segmentation of the image and the next fig. 2 shows the result of the segmentation. A common approach to model image data is to extract a vector of features from each image in the database and then use the Euclidean distance between those feature vectors as similarity measure for images. But the effectiveness of this approach is highly dependent on the quality of the feature transformation which results in very high-dimensional feature vectors. Those extremely high-dimensional feature vectors cause many problems

commonly described by the term ‘curse of dimensionality’.

The segmentation steps involve the following:

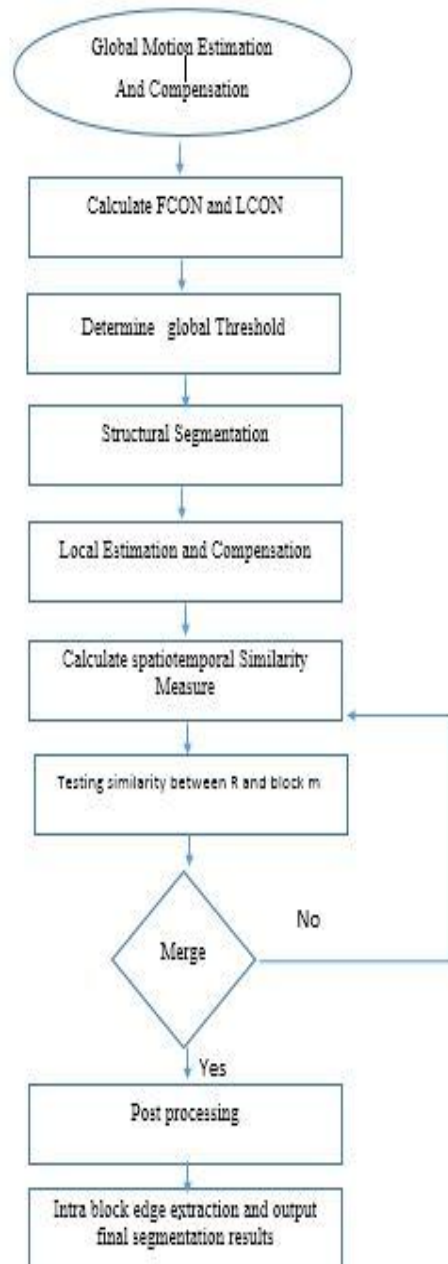


Fig. 1 steps involve the segmentation

Segmentation provides the image results as:

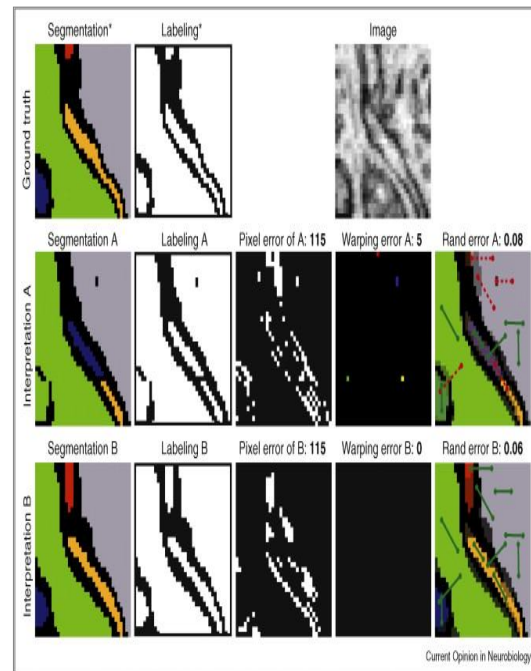


Fig. 2 The segmentation the image.

### III. OBJECTIVES

- ✦ Inspired by those works, we propose a colour image retrieval technique that utilizes the compressed information included in the constructed image dictionary.
- ✦ A given image is convolved with Gaussian filters at different scales, and then the difference of successive Gaussian blurred images are taken.
- ✦ By processing the segmentation algorithm after Gaussian filters, then apply the sift algorithm to mark the image in to various segments, and apply the blob technique to identify the differentiated image segments and produce it in more efficient and cleared image.

### IV. RELATED WORK

#### A. Global methods:

This technique deals with image globally and tries to characterize it by using visual/statistical features calculated from the entire image. Visual

features are classified into primitive features such as colour or shape, logical features such as identity of objects shown and abstract features such as significance of scenes depicted.

- **Colour:** In domain of photograph retrieval, colour has been the most effective feature and almost all systems employ colours. Although most of the images are in the red, green, blue (RGB) colour space. Color histograms are used to compare images in many applications. Their advantages are efficiency, and insensitivity to small changes in camera viewpoint. However, colour histograms lack spatial information, so images with very different appearances can have similar histograms.
- **Texture:** An **image texture** is a set of metrics calculated in image processing designed to quantify the perceived texture of an image. Image Texture gives us information about the spatial arrangement of colour or intensities in an image or selected region of an image. Some of the most common measures for capturing the texture of images are wavelets and Gabor filters. These texture measures try to capture the characteristics of the image or image parts with respect to changes in certain directions and the scale of the changes. This is most useful for regions or images with homogeneous texture. Again, invariances with respect to rotations of the image, shifts or scale changes can be included into the feature space.
- **Shape features:** There are many shape representation and description techniques in the literature. Shape description or representation is an important issue both in object recognition and classification. It has been used in CBIR in conjunction with colour and other features for indexing and retrieval. Many techniques, including chain code, polygonal approximations, curvature, Fourier descriptors and moment descriptors have been

proposed and used in various applications. The query images are represented by Fourier descriptors which serve powerful boundary-shape representation tools because of invariance property in affine transformation.

A common approach to model image data is to extract a vector of features from each image in the database (e.g. a colour histogram) and then use the Euclidean distance between those feature vectors as similarity measure for images. But the effectiveness of this approach is highly dependent on the quality of the feature transformation. Often it is necessary to extract many features from the database objects in order to describe them sufficiently, which results in very high-dimensional feature vectors. Those extremely high-dimensional feature vectors cause many problems commonly described by the term 'curse of dimensionality'. Especially for image data, the additional problem arises how to include the structural information contained in an image into the feature vector. As the structure of an image cannot be remodelled by a low-dimensional feature vector, the dimensionality Problem gets even worse.

#### **B.BLOB EXTRACTION:**

The detection and description of local image features can help in object recognition. The Scale Invariant Feature Transform (SIFT) features are local and based on the appearance of the object at particular interest points, and are invariant to image scale and rotation. They are also robust to changes in illumination, noise, and minor changes in viewpoint. In the field of computer vision, **blob detection** refers to mathematical methods that are aimed at detecting regions in a digital image that differ in properties, such as brightness or colour, compared to areas surrounding those regions. Informally, a **blob** is a region of a digital image in which some properties are constant or vary within a prescribed range of values; all the points in a blob can be considered in some sense to be similar to each other. There are several motivations for studying and developing blob detectors. One main reason is to provide complementary information about regions, which is not obtained from edge detectors or corner detectors.

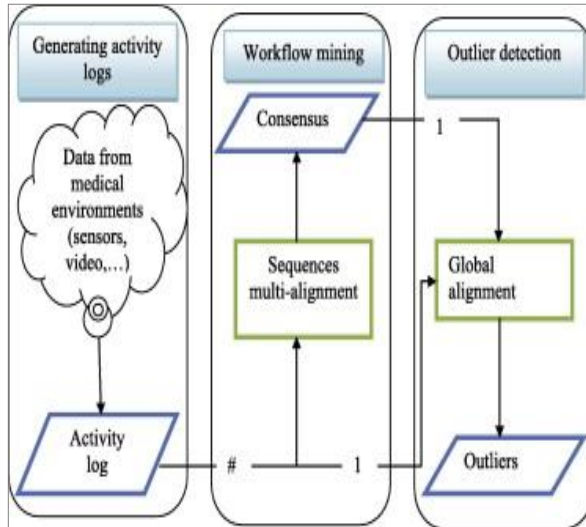


Fig. 3 The blob extraction.

In early work in the area, blob detection was used to obtain regions of interest for further processing. These regions could signal the presence of objects or parts of objects in the image domain with application to object recognition and/or object tracking. In other domains, such as histogram analysis, blob descriptors can also be used for peak detection with application to segmentation. Another common use of blob descriptors is as main primitives for texture analysis and texture recognition. In more recent work, blob descriptors have found increasingly popular use as interest points for wide baseline stereo matching and to signal the presence of informative image features for appearance-based object recognition based on local image statistics. There is also the related notion of ridge detection to signal the presence of elongated objects.

The Laplacian of Gaussian, one of the first and also most common blob detectors is based on the Laplacian of the Gaussian (Log). Given an input image  $f(x,y)$ , this image is convolved by a Gaussian kernel

$$g(x, y, t) = \frac{1}{2\pi t} e^{-(x^2+y^2)/(2t)}$$

At a certain scale  $t$  to give a scale space representation

$$L(x, y, t) = g(x, y, t) * f(x, y)$$

$$\nabla^2 L = L_{xx} + L_{yy}$$

Then, the Laplacian operator is computed, which usually results in strong positive responses for dark blobs of extent  $\sqrt{2t}$  and strong negative responses for bright blobs of similar size. A main problem when applying this operator at a single scale, however, is that the operator response is strongly dependent on the relationship between the size of the blob structures in the image domain and the size of the Gaussian kernel used for pre-smoothing. In order to automatically capture blobs of different (unknown) size in the image domain, a multi-scale approach is therefore necessary. A straightforward way to obtain a multi-scale blob detector with automatic scale selection is to consider the scale-normalized Laplacian operator

$$\nabla_{norm}^2 L(x, y, t) = t(L_{xx} + L_{yy})$$

And to detect scale-space maxima/minima, that are points that are simultaneously local maxima/minima of  $\nabla_{norm}^2 L$  with respect to both space and scale (Lundberg 1994, 1998). Thus, given a discrete two-

dimensional input image  $f(x, y)$  a three-dimensional discrete scale-space volume  $L(x, y, t)$  is computed and a point is regarded as a bright (dark) blob if the value at this point is greater (smaller) than the value in all its 26 neighbors. Thus, simultaneous selection of interest points  $(\hat{x}, \hat{y})$  and scales  $\hat{t}$  is performed according to

$$(\hat{x}, \hat{y}, \hat{t}) = \text{argmax}_{(x,y,t)} \text{local} (\nabla_{norm}^2 L(x, y, t))$$

### C. SIFT Algorithm:

SIFT is an image processing algorithm which can be used to detect distinct features in an image. A "clustering" is essentially a set of such clusters, usually containing all objects in the data set. Additionally, it may specify the relationship of the clusters to each other, for example a hierarchy of clusters embedded in each other. Data Clustering is one of the challenging mining techniques exploited in the knowledge discovery process. Clustering huge amounts of data is a difficult task since the goal is to find a suitable partition in a unsupervised way (i.e. without any prior knowledge) trying to maximize the similarity of objects belonging to the same cluster and minimizing the similarity among objects in different clusters.



SIFT has four computational (reckon or calculate) Phases perform by SIFT, computations are expensive. Extracting the key-point is minimized by the use of cascading approach. The output of the SIFT algorithm is a set of key-point descriptors. Once such descriptor have generated for more than one image. The image matching or object matching is not only a part of SIFT algorithm. For matching we use a nearest neighbour search (NNS).

$$D(\mathbf{x}) = D + \frac{\partial D^T}{\partial \mathbf{x}} \mathbf{x} + \frac{1}{2} \mathbf{x}^T \frac{\partial^2 D}{\partial \mathbf{x}^2} \mathbf{x}$$

**D. Four Computational Phases**

Since we want to determine whether SIFT is a suitable algorithm to use as part of a CBIR service, we should know that how algorithm works.

**Phase 1: scale-space Extreme detection**

The first phase of the computation seeks to identify potential interest points. It will searches over all scales and image locations. The computation is accomplished by using a difference-of-Gaussian (DOG) function. The resulting interest points are invariant to scale and rotation, meaning that they are persistent across image scales and rotation.

**Phase 2: key point localization**

For interest point found in phase 1, a detailed model is created to determine location and scale. Key point is selected based on their stability, and a stable key point is thus a key point resistant to image distortion.

**Phase 3: orientation assignment**

For each of the key points identified in phase2, SIFT computes the direction of gradients around. One or more orientations are assigned to each key point based on local image gradient directions.

**Phase 4: key point descriptor**

The local image gradients are measured in the region around each key point. These are transformed into a representation that allows for significant levels of local shape distortion and change in illumination.

Hilbert transform of  $f_1(x, y)$  along the y axis:

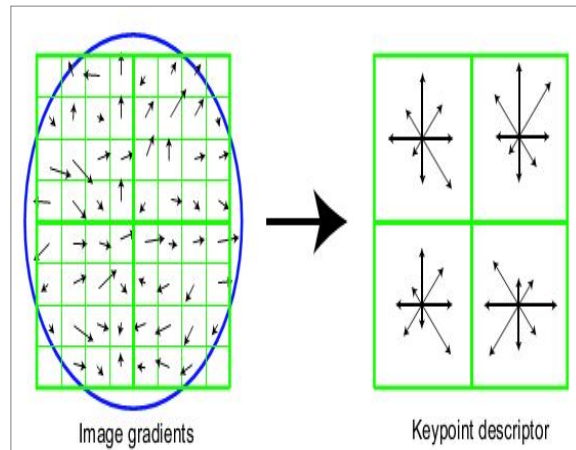


Fig. 4 the image gradients & key point descriptor.

The SIFT detector extracts from an image a collection of frames or key points. These are oriented disks attached to blob-alike structures of the image. As the image translates, rotates and scales, the frames track these blobs and thus the deformation. By canonization, i.e. by mapping the frames to a reference (a canonical disk), the effect of such deformation on the feature appearance is removed. The SIFT descriptor is a coarse description of the edge found in the frame. Log is expensive, so let's approximate it using the heat-diffusion equation

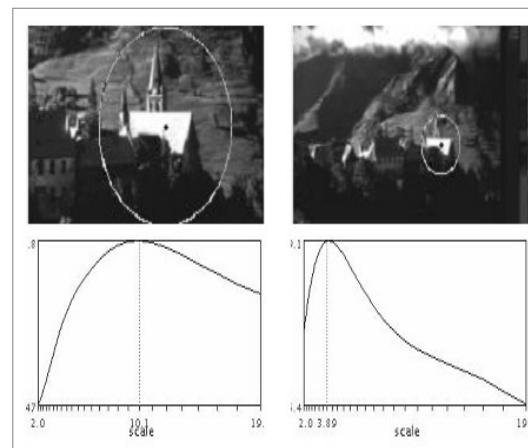


Fig. 5 the fine image enhancement.

The graph results are shown here:

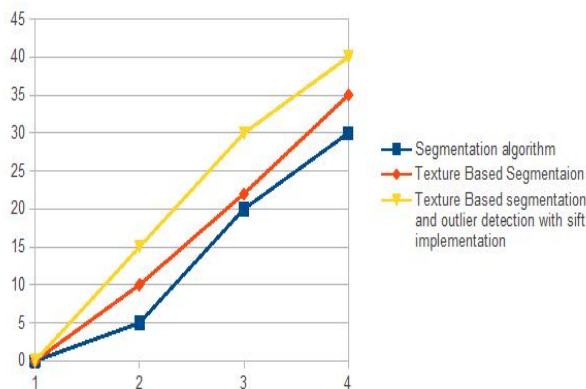


Fig. 6 the graphical representation image enhancement.

This graph represents the fine image enhancement by using the texture based segmentation and outlier's detection with the implementation of sift algorithm:

## V. CONCLUSION

In this paper, a graph based representation was proposed in a CBIR context. From a partition into regions processed by an efficient segmentation algorithm, a Region Adjacency Graph was built to consider spatial relationships between regions. Each region is characterized using a set of features based on the Colour, Texture and Shape. Inspired and stimulated by the results of independence between segmentation and sift algorithm methods, an interesting work will come up. It would aim at speeding up the system by computing the segment images and results in producing the sharpening and clarity in image. In future work the hidden markov queues can be applied in segmentation algorithm by identifying the various segments in an image.

## REFERENCES

- [1]. I. L. Coman, I.K. Sethi, Mining association rules between low-level image features and high-level concepts, in: Proceedings of the SPIE Data Mining and Knowledge Discovery, vol. III, 2001, pp. 279–290.
- [2]. K. Barnard, D. Forsyth, Learning the semantics of words and

pictures, in: IEEE International Conference on Computer Vision (ICCV), vol. 2, 2001, pp. 408–415. <http://dx.doi.org/10.1109/ICCV.2001.937654>, Url:<<http://dx.doi.org/10.1109/ICCV.2001.937654>>.

- [3]. J. Shi, J. Malik, Normalized cuts and image segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (8) (2000) 888–905. <<http://citeseer.ist.psu.edu/shi97normalized.html>>.
- [4]. B. Zitov, J. Flusser, Image registration methods: a survey, Image and Vision Computing 21 (2003) 977–1000.
- [5]. Y. Rui, T.S. Huang, M. Ortega, S. Mehrotra, Relevance feedback: a power tool for interactive content-based image retrieval, IEEE Transactions on Circuits and Systems for Video Technology 644–655.
- [6]. A.Smeulders, M. Worring, S. Santini, A. Gupta, R. Jain, Content-based image retrieval at the end of the early years, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (12) (2000) 1349–1380.
- [7]. E. Acuna, C. Rodriguez, The treatment of missing values and its effect in the classifier accuracy, Classification, Clustering and Data Mining Applications (2004) 639–648.
- [8]. G Smith, I. Burns, Measuring texture classification algorithms, Pattern Recognition Letters 18 (1997) 1495–1501.
- [9]. B.S. Manjunath, W.Y. Ma, Texture features for browsing and retrieval of image data, IEEE Trans. Pattern Anal. Mach. Intell. 18 (8) (1996) 837–842, <http://dx.doi.org/10.1109/34.531803>.