

Object Capturing In A Cluttered Scene By Using Point Feature Matching

Ch. Sravani*, B. Harikrishna**, K. Gayatri**, K. Anusha**, K. Pydiraju**

*Assistant Professor, ECE Department, Sri Sivani College Of Engineering

** IV/IV B. Tech, ECE Department, Sri Sivani College Of Engineering

ABSTRACT

Capturing means getting or catching. This project contains an algorithm for capturing a specific target based on the points which corresponds between reference and target image. It can capture the objects in-plane rotation and also effective to small amount of out-of plane rotation also.

This method of object capturing works best for objects that exhibit in a cluttered texture patterns, which give rise to unique point feature matches. When a part of object is occluded by other objects in the scene, only features of that part are missed. As long as there are enough features detected in the unoccluded part, the object can be captured. The local representation is based on the appearance. There is no need to extract geometric primitives (e.g. lines) which are generally hard to detect reliably.

Key words: object capture, matching technique, occlude, geometric primitives.

I. INTRODUCTION

An object capturing system finds objects in the real world from an image of the world, using object models which are known a priori. This task is surprisingly difficult. Humans perform object capturing[1] effortlessly and instantaneously. Algorithmic description of this task for implementation on machines has been very difficult. In this chapter we will discuss different steps in object capturing and introduce point feature matching technique that have been used for object capturing in many applications. We will discuss the different types of capturing tasks that a vision system may need to perform. We will analyze the complexity of these tasks and present approaches useful in different phases of the capturing task. The object capturing problem can be defined as a labelling problem based on models of known objects. Formally, given an image containing one or more objects of interest (and background) and a set of labels corresponding to a set of models known to the system, the system should assign correct labels to regions, or a set of regions, in the image. The object capturing[4] problem is closely tied to the segmentation problem: without at least a partial capturing of objects, segmentation cannot be done, and without segmentation, object capturing is not possible. In this chapter, we discuss basic aspects of object capturing. We present the architecture and main components of object capturing and discuss their role in object capturing systems of varying complexity.

II. RELATED WORK

In object capturing to capture a target image in a cluttered scene three methods are include. There are

- i. Appearance based methods
- ii. Geometry based methods
- iii. Recognition as a Correspondence of Local Features

RECOGNITION AS A CORRESPONDENCE OF LOCAL FEATURES:

Neither geometry-based nor appearance-based methods discussed previously do well as defined by the requirements stated in the beginning of the paper, i.e. the generality, robustness, and easy learning. The methods are also sensitive to occlusion of the objects, and to the unknown background, thus they are not robust. As an attempt to address the above mentioned issues, methods based on matching local features have been proposed. Objects are represented by a set of local features, which are automatically computed from the training images. The learned features are organised into a database. When recognising a query image, local features are extracted as in the training images. Similar features[2] are then retrieved from the database and the presence of objects is assessed in the terms of the number of local correspondences. Since it is not required that all local features match, the approaches are robust to occlusion and cluttered background. To recognise objects from different views, it is necessary to handle all variations in object appearance. The variations might be complex in general, but at the scale of the local features they can be modelled by simple, e.g. affine, transformations.

Thus, by allowing simple transformations at local scale, significant viewpoint invariance is achieved even for objects with complicated shapes.

In our project, the object can be captured by using point feature matching technique. It is one of the technique of the local features[5].

POINT FEATURE MATCHING TECHNIQUE: DEFINITION OF FEATURE:

Feature is defined as an "interesting" part of an image and features are used as a starting point for many computer vision algorithms. The desirable property for a feature detector is *repeatability*: whether or not the same feature will be detected in two or more different images of the same scene. Feature detection is computationally expensive and there are time constraints, a higher level algorithm may be used to guide the feature detection stage, so that only certain parts of the image are searched for features.

TYPES OF IMAGE FEATURES:

- Edges
- Corners / interest points
- Blobs / regions of interest or interest points
- Ridges

Feature Detection and Extraction:

A feature is an interesting part of an image, such as a corner, blob, edge, or line. Feature extraction[3] enables you to derive a set of feature vectors, also called descriptors, from a set of detected features. Computer vision system toolbox offers capabilities for feature detection and extraction that include: Corner detection, including Shi & Tomasi, Harris, and FAST methods

- BRISK, MSER, and SURF detection for blobs and regions
- Extraction of BRISK, FREAK, SURF, and simple pixel neighbourhood descriptors
- Histogram of Oriented Gradients (HOG) feature extraction.



Fig2.1 SURF (left), MSER (center), and corner detection (right) with Computer Vision System Toolbox. Using the same image, the three different feature types are detected and results are plotted over the original image.

Feature matching is the comparison of two sets of feature descriptors obtained from different images to provide point correspondences between images.

Our approach In this paper, we propose a SURF algorithm for extracting, description and matching the images.

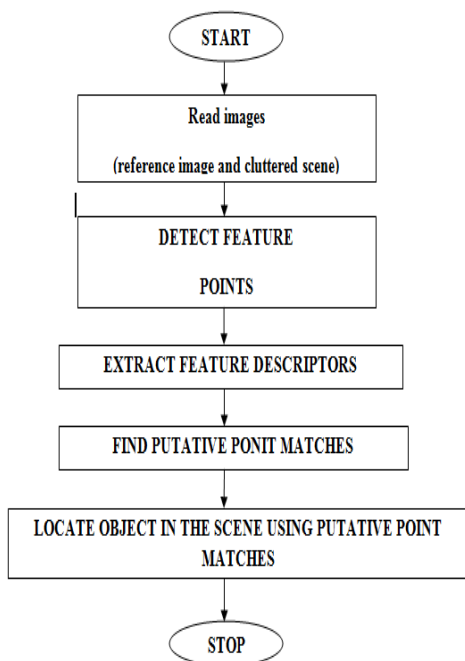
III. SURF FEATURE ALGORITHM

The SURF Algorithm SURF is developed by Bay et al. and it stands for Speeded Up Robust Features. SURF algorithm is actually based on the SIFT algorithm. It uses integral images and approximations for achieving higher speed than SIFT. These integral images are used for convolution. Like SIFT, SURF works in three main stages: extraction, description, and matching. The difference between SIFT and SURF is that SURF extracts the features from an image using integral images and box filters. The extraction of the key points from an image is a process that requires image filtering. SURF implements these filters using box filters. A very interesting pre-processing step is the conversion of the original image into a so-called integral image.

Integral images are very easily computed by adding the right pixel values. In an integral image every pixel is the sum of all pixels located in a rectangular window formed by that pixel and the origin, with the origin being the most top-left pixel. Box filters are used as an approximation of the exact filter masks. By using integral images together with box filters a major speed up is realized. Another difference in the extraction of key points is that SIFT rescales the image, while SURF changes the filter mask. The term box-space is used to distinguish it from the usual scale-space. While the scale space is obtained by convolution of the initial images with Gaussians, the discrete box-space is obtained by convolving the original image with box filters at several different discrete sizes. In the detection step, the local maxima of a Hessian-like operator, the Box Hessian operator, applied to the box-space are computed to select interest point candidates. These candidates are then validated if the response is above a given threshold. Both box size and location of these candidates are then refined using an iterated procedure fitting locally a quadratic function. Typically, a few hundreds of interest points are detected in a digital image of one mega-pixel. Therefore, SURF builds a descriptor that is invariant[8][9] to view- point changes of the local neighbourhood of the point of interest. Like in SIFT, the location of this point in the box- space provides invariance to scale and provides scale and translation invariance. To achieve rotation invariance, a dominant orientation is defined by considering the local gradient orientation distribution, estimated with Haar wavelets. Making use of a spatial localization grid[6][7], a 64- dimensional descriptor is then built, corresponding to a local histogram of the Haar wavelet responses[10].

IV. EXPERIMENTAL RESULTS

4.1. FLOW CHART:



4.1.1 Read Images:

4.1.1. a) Reference Image:



Fig4.1 reference image

4.1.1. b) Cluttered Scene:



Fig4.2 cluttered scene

4.1.2. Detect Feature Points:

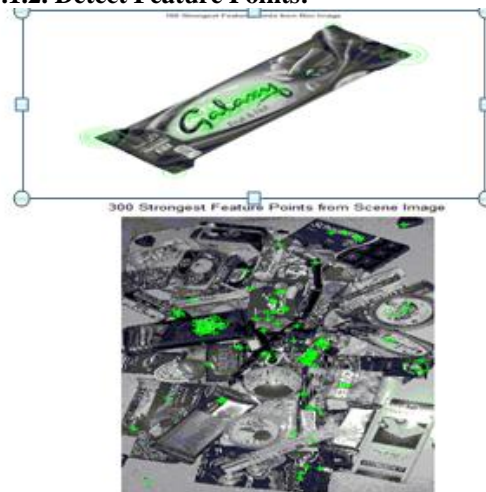


Fig4.3.strongest feature points from reference image and cluttered image.

4.1.3. Extract Feature descriptors:

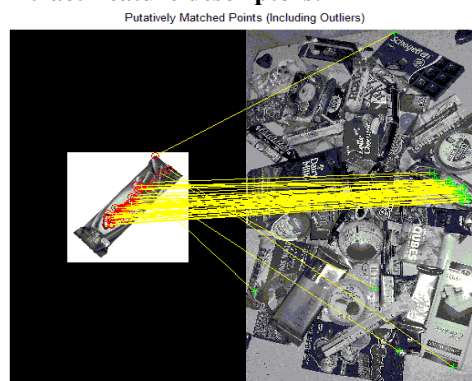


Fig4.4. extract feature descriptors from reference image and cluttered scene

4.1.4. Find Putative Matched Points:

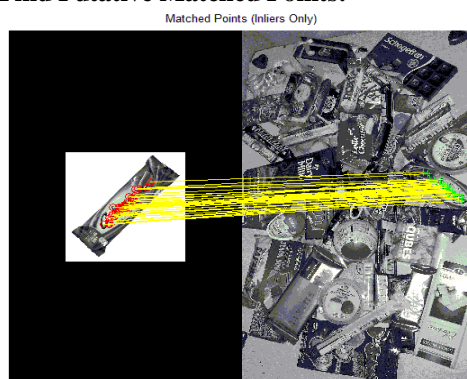


Fig4.5. matched points

4.1.5. Locate Object In The Scene:

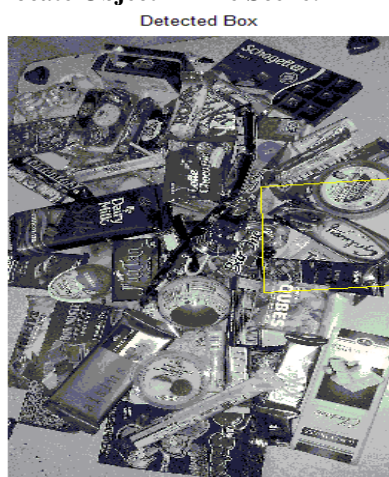


Fig4.7.captured the target image in a cluttered scene

V. CONCLUSION

This method of object capturing works best for objects that exhibit non-repeating texture patterns, which give rise to unique feature matches. This technique is not likely to work well for uniformly-coloured objects, or for objects containing repeating patterns. Note that this algorithm is designed for detecting a specific object, for example, the elephant in the reference image, rather than any elephant. For detecting objects of a particular category, such as people or faces, see vision.CascadeObjectDetector.

REFERENCE

- [1] G.R. Bradski "Real Time Face and Object Tracking as a Component of a Perceptual User Interface", Proceedings of the 4th IEEE Workshop on Applications of Computer Vision, 1998.
- [2] Viola, Paul A. and Jones, Michael J. "Rapid Object Detection using a Boosted Cascade of Simple Features", IEEE CVPR, 2001.
- [3] M. Brown and D. Lowe. Invariant features from interest point groups. In BMVC02, 2002.
- [4] D. Jugessur and G. Dudek. Local appearance for robust object recognition. In Computer Vision and Pattern Recognition.
- [5] D. G. Lowe. Distinctive image features from scale-invariant key points. International Journal on Computer Vision, 2004.
- [6] D.G. Lowe. Object recognition from local scale-invariant features In ICCV99, pages 1150–1157, 1999.
- [7] Lowe, D: Object recognition from local scale-invariant features. In: ICCV. (1999).
- [8] Carneiro, G., Jepson, A.: Multi-scale phase-based local features. In: CVPR (1). (2003) 736 -743.
- [9] Mindru, F., Tuytelaars, T., Van Gool, L., Moons, T.: Moment invariants for recognition under changing viewpoint and illumination. CVIU 94 (2004) 3–27.
- [10] Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. PAMI 27 (2005) 1615–1630.