Image processing and computer vision techniques have been increasingly used in various applications, such as object recognition, face detection, and autonomous navigation.

In this project, we will focus on the application of image processing and computer vision techniques in detecting the validity of a Pepsi can.

The camera will capture an image of a Pepsi can and send it to MATLAB, where Surf and Orb algorithms will be used to detect the object.

The output of the algorithm will determine if the can is valid or invalid.

What Is Image Processing?

Image processing is the process of transforming an image into a digital form and performing certain operations to get some useful information from it. The image processing system usually treats all images as 2D signals when applying certain predetermined signal processing methods.

Types of Image Processing

There are five main types of image processing:

- Visualization - Find objects that are not visible in the image.
- Recognition - Distinguish or detect objects in the image.
- Sharpening and restoration - Create an enhanced image from the original image.
- Pattern recognition - Measure the various patterns around the objects in the image.
- Retrieval - Browse and search images from a large database of digital images that are like the original image.

Components of Image Processing

Computer

Hardware for Specialized Image Processing

Massive Storing

Camera

Software

Networking

Algorithms

It is essential to know that image processing algorithms have the most significant role in digital image processing.
Developers have been using and implementing multiple image processing algorithms to solve various tasks,
including digital image detection, image analysis, image reconstruction, image restoration, image enhancement, image data compression, spectral image estimation, and image estimation. Sometimes, the algorithms can be straight off the book or a more customized amalgamated version of several algorithm functions.

In the case of full image capture, image processing algorithms are generally classified into:

- Low-level methods, noise removal, and color enhancement
- Medium-level techniques, like binarization and compression
- Higher-level techniques involve detection, segmentation, and recognition algorithms extracting semantic information from the captured data

Types of Image Processing Algorithms

There are different types of image processing algorithms. The techniques used to process images are image generation and image analysis.

The basic idea behind this is converting an image from its original form into a digital image with a uniform layout.

Some of the conventional image processing algorithms are as follows:

Contrast Enhancement algorithm

- **Histogram equalization algorithm:** Using the histogram to improve image contrast.
- **Adaptive histogram equalization algorithm:** It is the histogram equalization that adapts to local changes in contrast.
- **Connected-component labeling algorithm:** It is about finding and labeling disjoint regions.

Elser difference-map algorithm

It is a search algorithm used for general constraint satisfaction problems. It was used initially for X- Ray diffraction microscopy.

Feature detection algorithm

consists of:

- **Marr–Hildreth algorithm:** It is an early edge detection algorithm.
- **Canny edge detector algorithm:** Canny edge detector is used for detecting a wide range of edges in images.
- **Generalized Hough transform algorithm.**
- **Hough transform algorithm.**
- **SIFT (Scale-invariant feature transform) algorithm:** SIFT is an algorithm to identify and define local features in images.
- **SURF (Speeded Up Robust Features) algorithm:** SURF is a robust local feature detector.
- **Richardson–Lucy deconvolution algorithm:** This is an image de-blurring algorithm.

Blind deconvolution algorithm:

Like Richardson–Lucy deconvolution algorithm, it is an image de- blurring algorithm when the point spread function is unknown.

Segmentation algorithms:

This algorithm parts a digital image into two or more regions. It consists of:

- Grow Cut algorithm
- Random walker algorithm
- Region growing algorithm
- Watershed transformation algorithm

It is to note that apart from the algorithms mentioned above, industries also create customized algorithms to address their needs.

They can be right from scratch or a combination of various algorithmic functions. It is safe to say that with the evolution of computer technology, image processing algorithms have provided sufficient opportunities for multiple researchers and developers to investigate, classify, characterize, and analyze various hordes of images.

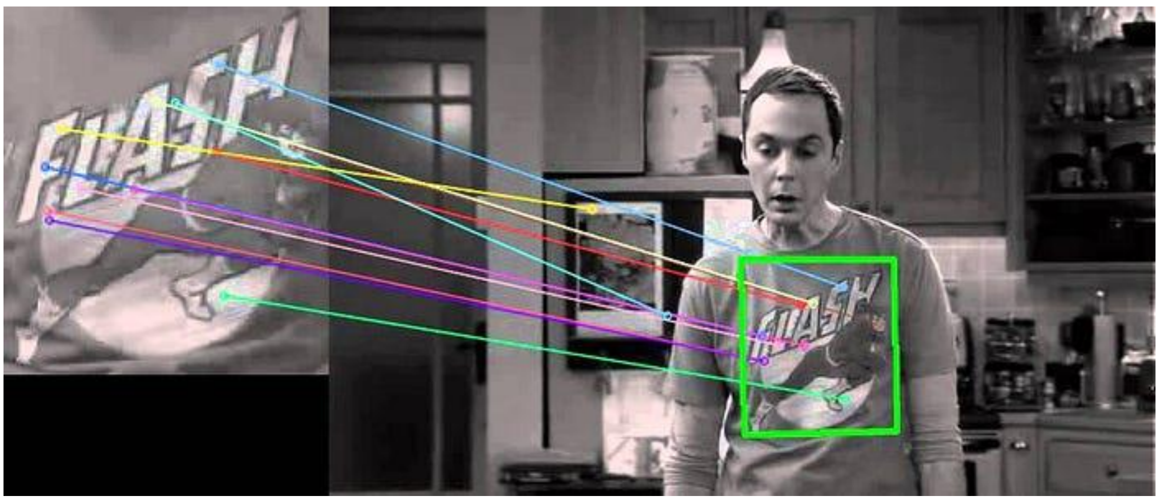
Used Algorithms => orb & surf

The SURF stands for Speeded Up Robust Features algorithm is a popular computer vision technique used for feature detection, description, and matching in images.

It is an improvement over the SIFT algorithm and is known for its robustness and efficiency.

SURF uses the Hessian matrix to identify key points in an image and then extracts features based on the intensity values and gradients of the surrounding pixels.

It is scale and rotation invariant, making it suitable for object detection in real-world scenarios.



SURF is composed of two steps.

- **Feature Extraction**
- **Feature Description**

How does SURF work?

SURF works by extracting key points or interest points from an image. These key points are areas in the image that have unique properties and can be easily distinguished from their surroundings. The algorithm then computes the descriptors for these key points, which are compact representations of their local neighborhood.

Key point detection: SURF uses a scale-space approach to detect key points at multiple scales.
It applies a series of Gaussian filters to the image at different scales and then identifies locations where the response to these filters is maximum or minimum.

These locations correspond to potential key points.

Orientation assignment: Once the key points are identified, SURF calculates their dominant orientation.

This allows the algorithm to make the descriptors invariant to image rotations.

The dominant orientation is determined by computing the gradient orientations within a neighborhood around each key point.

Descriptor calculation: After the key points and their orientations are determined, SURF constructs descriptors that capture information about the local neighborhood around each key point.

These descriptors are designed to be robust against changes in scale, rotation, and illumination.

Advantages of SURF

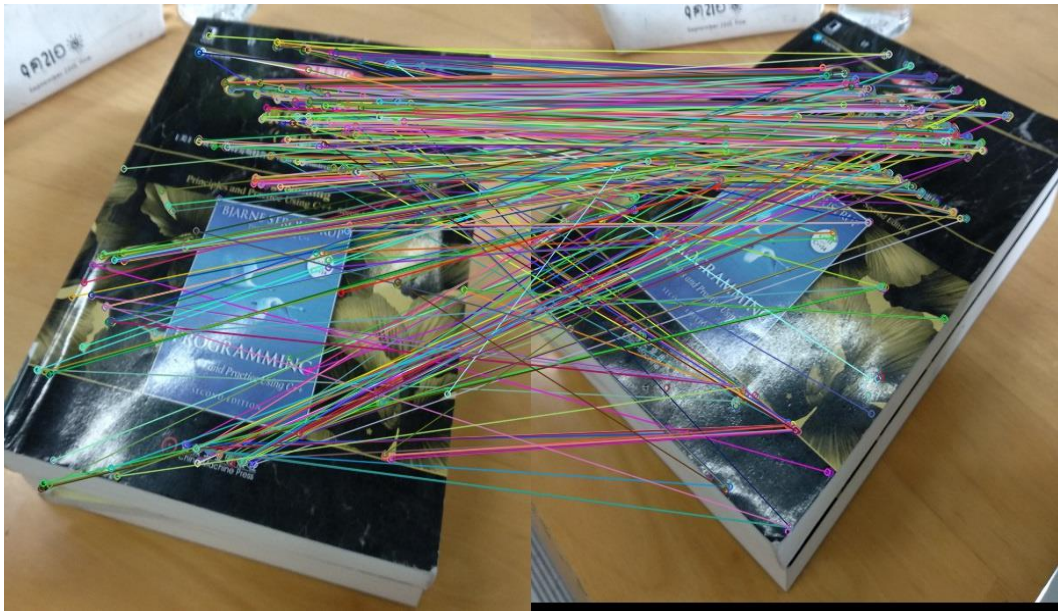
- SURF is highly efficient due to its use of integral images, which enable fast calculations of box filters.
- The algorithm is also robust against geometric transformations such as rotation, scaling, and affine transformations.
- SURF can handle images with significant noise and occlusion.

Limitations of SURF

- SURF may not perform well in images with repetitive patterns or textures as it relies on unique key points.
- The algorithm is sensitive to changes in viewpoint and lighting conditions.

ORB algorithm

ORB stands for Oriented FAST and Rotated BRIEF is a feature detection and description algorithm used in image processing and computer vision applications.
It is an extension of the FAST (Features from Accelerated Segment Test) algorithm and the BRIEF (Binary Robust Independent Elementary Features) descriptor.
Oriented FAST and Rotated BRIEF (ORB) was developed at OpenCV labs by Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary R. Bradski in 2011, as an efficient and viable alternative to SIFT and SURF



The ORB image matching algorithm is generally divided into three steps:

feature point extraction, generating feature point descriptors and feature point matching

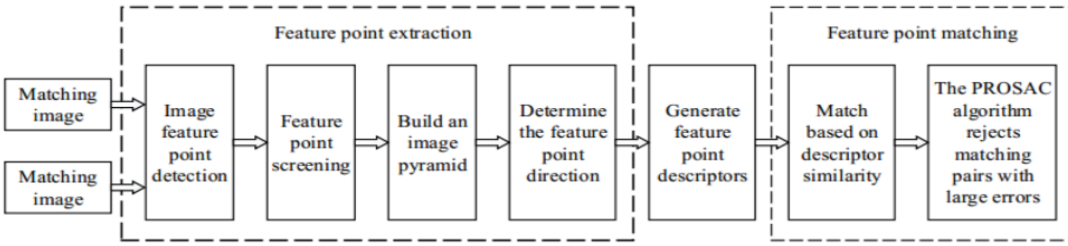


Fig.1 Image matching flow chart based on ORB algorithm

ORB performs as well as SIFT on the task of feature detection (and is better than SURF) while being almost two orders of magnitude faster.
ORB builds on the well-known FAST key point detector and the BRIEF descriptor.
Both techniques are attractive because of their good performance and low cost. ORB's main contributions are as follows:

- The addition of a fast and accurate orientation component to FAST
- The efficient computation of oriented BRIEF features
- Analysis of variance and correlation of oriented BRIEF features
- A learning method for decorrelating BRIEF features under rotational invariance, leading to better performance in nearest neighbor applications.

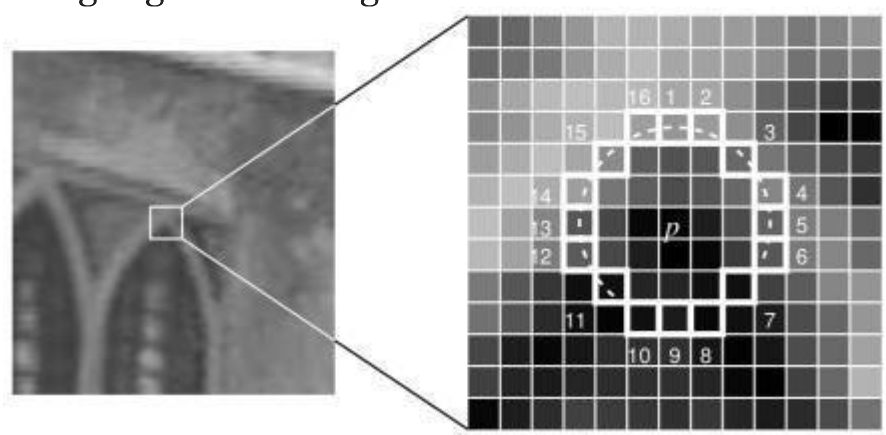
Fast (Features from Accelerated and Segments Test)

Given a pixel p in an array fast compares the brightness of p to surrounding 16 pixels that are in a small circle around p.

Pixels in the circle is then sorted into three classes (lighter than p, darker than p or similar to p).

If more than 8 pixels are darker or brighter than p than it is selected as a key point.

So key points found by fast gives us information of the location of determining edges in an image.

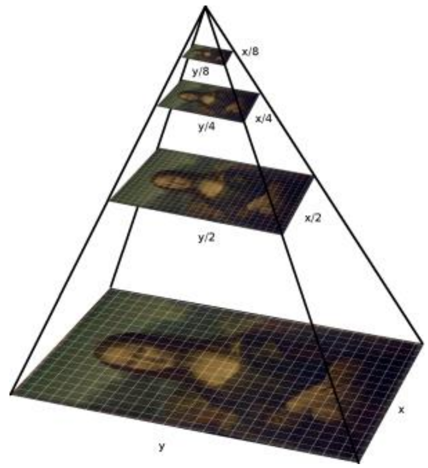


However, FAST features do not have an orientation component and multiscale features so orb algorithm uses a multiscale image pyramid.

An image pyramid is a multiscale representation of a single image, that consist of sequences of images all of which are versions of the image at different resolutions.

Each level in the pyramid contains the down sampled version of the image than the previous level.

Once orb has created a pyramid it uses the fast algorithm to detect key points in the image. By detecting key points at each level orb is effectively locating key points at a different scale. In this way, ORB is partial scale invariant.



After locating key points orb now assign an orientation to each key point like left or right facing depending on how the levels of intensity change around that key point.

For detecting intensity change orb uses intensity centroid.

The intensity centroid assumes that a corner's intensity is offset from its center, and this vector may be used to impute an orientation.

First, the moments of a patch are defined as:

$$m_{pq} = \sum_{x,y} x^p y^q I(x, y)$$

ORB descriptor-Patch's moment's definition

With these moments we can find the centroid, the "center of mass" of the patch as:

$$C = \left(\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}} \right)$$

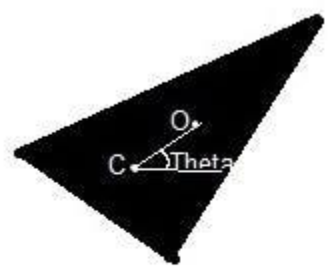
ORB descriptor — Center of the mass of the patch

We can construct a vector from the corner's center O to the centroid -OC. The orientation of the patch is then given by:

$$\theta = \operatorname{atan2}(m_{01}, m_{10})$$

ORB descriptor — Orientation of the patch

Here is an illustration to help explain the method:



Once we've calculated the orientation of the patch, we can rotate it to a canonical rotation and then compute the descriptor, thus obtaining some rotation invariance.

Implementation

For detecting the validity of a Pepsi can the camera will be fixed on the conveyer then it will take the reference image and store it in order to increase the algorithms accuracy also to increase the computational speed make it more faster then, conveyer move to the first stopping point where the camera will image the distorted or the valid can and send it to MATLAB, where Surf and Orb algorithms will be used to detect and compare the reference and the object.
The output of the algorithm will determine if the can is valid or invalid.
The output will be a threshold which represent the maximum accepted distortion in captured can.

Appendix B

- **Reference code**
- **Main code**