



ADAPTABLE XY SCANNER

Prof. Jiss Paul, Anish K.S, Arjun Muralidas, Dan Lawrence, Neenu Chandran, Veena G Krishna
Electronics and Communication Engineering Department,
Mar Athanasius College Of Engineering, Kothamangalam,
Ernakulam, Kerala

Abstract— Adaptable XY Scanner is consumer electronics product prototype, which is the fusion of a standalone XY scanner and a pen plotter, since both the products requires the same type of hardware. Both a plotter and a XY scanner needs to access points in a 2 dimensional space. While the pen plotter uses a pen to plot onto the plane the scanner utilizes a scan head to read/scan from the plane. The difference thus lies only in the head, as transducer based scan head or a pen mounted plotter head. The proposed system operates in two modes a scanning mode and plotting mode. The system allows the provision for inter changeability of the scan heads.

I. INTRODUCTION

The 21st century consists of all sorts of electronic gadgets. Whole market is over flooded with different varieties of electronic goods. They vary with price and uses. Scanner and plotter are one among those. They varied with mechanisms, materials used. One similarity between these devices is their hardware. Both these have a similar structure which access points in two dimensional spaces. Scanner and plotter are one among those.

A consumer electronics product that fuses a 2D scanner with a plotter together. With the provision for switching scan heads, the same hardware can be used for a wide range of applications. The basic hardware and associated firmware have two modes of operations, one scanning mode in which it will function as an image scanner that scans a 2D surface and a plotting mode in which the vector input to the system will be converted to drawings on paper by the pen plotting head of the device. The scanner is adaptive in the sense that head can be switched. If a laser head is used in place of pen holder laser etching/engraving can be carried out. Similarly an ultrasound transducer in place of the image scan head can convert the device to have NDT capabilities like crack detection in a metal sheet.

II. EXPERIMENTAL SETUP

This project is largely dependent on hardware; it is the uniformity of the hardware among scanners, plotters and laser cutters which made this project possible. Also being a hardware product based project this requires the hardware to be built prior to fine tuning of the software. Another point was that this project cannot be tested out in some simulator in a PC thus the initial approach was to build the hardware first.

The entire prototype is setup in a wooden base. A pair of stepper motor and lead screw assembly is used for X and Y directional motion on the wooden base. Aluminium channel and metal ball wheel is provided for the support of Y directional motor and the lead screw. Y directional carriage is given with showcase glass slider and rails for smooth motion with overhead wooden supporting beam.

The hardware requirement was to have a mechanism that can translate the scan head to any given point in a 2D plane and can trace fixed scan paths. The plotter head need to have the capability to raise and lower the pen. We dismantled an old printer + scanner unit to see the mechanism used within. It was belt drive with dc motor. Precise control was done with the help of feedback mechanism connected to shaft. But for us this was hard work, and no assurance was present for the precision. Also belt drives are more prone to sagging with the increase of scan head load and increase of scanning area. Possibility of slip also was there thus that choice was ruled out. Next obvious choice was to use stepper motors with lead screws.

The mechanism works as follows, one motor provides X axis motion and drives a horizontal traveler which holds the other stepper motor. The second motor drives the vertical traveler which comprises of the scanner head/pen plotter holding probe. Support rods and frames were used to prevent loading of the lead screws and unbalance. The bottom of the frames were screwed on to a wooden base. If vertical mounting was required then there has to be vacuum or suction mounting cups and associated pump with the hardware. The base plate can be wooden or transparent glass.

Scanner setup:-

The primary goal for this project is to generate a standalone scanner which can work as a plotter, by means of interchangeable heads. For the scanner, initially we need to have a stitching program for combining the acquired images. The second confusing fact is to design the interchangeable head for the scanner and plotter. From the earlier experimental results we came to know that webcam can be used for scanning and is easier to use. The scanner requirement is to have efficient, flexible software can directly interface with the webcam. The decision to use MATLAB was based on its capability for image processing and sending/receiving control

signals to and from arduino. The figure below shows the steps for the entire scanning process.

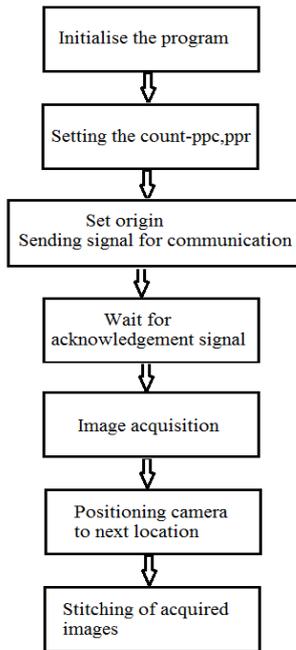


Figure 1: Flowchart for stitching

Plotter setup:-

For the plotter head as said earlier, servo motor is used to lift and lower the pen, designed with modifying the pen as in figure. The pen holder has nuts to adjust for pen size and for raising and lowering the pen, a small motor or solenoid used.



Figure 2 : Plotter Head



Figure 3 : Pen Head

III. CODING AND ALGORITHMS

A. Graphical User Interface(GUI)

Graphical user interface allows an interactive development environment through graphical icons and visual indicators. MATLAB is universally accepted as one of the most powerful data processing platform. Its connectivity with many advanced programming languages (like C , Java ,VB) and availability of a wide range of toolboxes make it popular, and the MATLAB-based GUI facilitates the user to control the operations on the entire system.

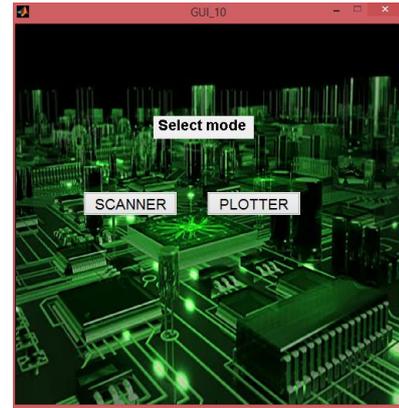


Figure 4: GUI 1



Figure 5: Scanner GUI

The proposed project design deals with a 2D scanner and plotter, GUI using MATLAB guide provides a better switching between the two.

The interfacing consists of a basic switching between scanner and plotter. As in figure, the first GUI consists of selection mode. When the selected mode is scanner, The GUI automatically replaces the first GUI with scanner GUI. As in figure 2, the scanner GUI initially asks for entering a name by which the scanned image to be saved. A SCAN push button is provided to start the image acquiring and stitching, and it will be displayed in the GUI window. When the selected mode is plotter, The GUI replaces with the plotter GUI as in figure 3. It consists of a list box from where the user can chose the G-code file. The provided RUN button initiates the plotting process by the selected G-code file.



Figure 6: Plotter GUI

B. *Stitching*

Image stitching is the process performed to generate one panoramic image from a series of smaller, overlapping images. Stitched images are used in applications such as interactive panoramic viewing of images, architectural walk-through, multi-node movies and other applications associated with modeling the 3D environment using images acquired from the real world. Most image processing techniques involve treating the image as a two-dimensional signal and applying standard signal processing techniques to it. Specifically, image stitching presents different stages to render two or more overlapping images into a seamless stitched image, from the detection of features to blending in a final image.

In this process, **Scale Invariant Feature Transform (SIFT) algorithm** [1] can be applied to perform the detection and matching control points step, due to its good properties. The process of create an automatic and effective whole stitching process leads to analyze different methods of the stitching stages. Several commercial and online software tools are available to perform the stitching process, offering diverse options in different situations.

This project's primary goal is to create a MATLAB script that will stitch two images together to create one larger image. In our experiments, we observe that a successful image stitching algorithm should not only create a smooth transition within the overlapped region but also preserve the salient features such as structural and intensity misalignment.

Due to the limited Field-Of-View (FOV) of a single camera, it is sometimes desired to extend the FOV using multiple cameras. Image stitching is one of the methods that can be used to exploit and remove the redundancy created by the overlapping FOV. However, the memory requirement and the amount of computation for conventional implementation of image stitching are very high. In this project, this problem is resolved by performing the image stitching in a strip-by-strip manner. Most of the existing methods of image stitching either produce a 'rough' stitch that cannot deal with stitching for tilted images or they require some user input.

Approaches for image stitching optimize the search for the best correlation point. The approach offered in this project makes the selection of the best correlation point in the following way. Based on knowledge about the expected overlap when using the motorized stage, it would be straight forward to find the best correlation point in the ideal case. However, the overlap area is not perfect, and certainly not to an accuracy of one pixel, due to deviations in stage position from the ideal and due to stage/camera misalignment. Our algorithm offers a way to overcome this problem by searching the small area around the expected central overlap pixel in order to find the best correlation point. Positioning of acquired images with a manual stage is much less accurate, so there is a need to search a wider area in order to find the best cross-correlation point.

The figure below shows the flowchart showing the steps for image stitching,

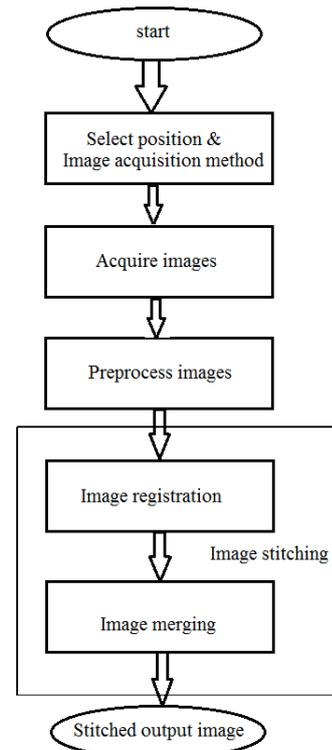


Figure 7: Flowchart for image stitching

The 3 important stages in the process of stitching images are:

1. *Image Acquisition:*

The images that need to be stitched are acquired using a camera mounted on a scan head. The camera movement is adjusted so that to get different overlapping sample images.

2. *Image Registration:*

The process of image registration aims to find the translations to align two or more overlapping images such that the projection from the view point through any position in the aligned images into the 3D world is unique. Image registration consists of four main steps; the set of features includes the intensity values, contours, textures and so on. A feature set must be selected for each image registration method. A similarity measure is done which returns a scalar value that provides an indication of the similarities between two features.

3. *Image Merging:*

Image merging is the process of adjusting the values of pixels in two registered images, such that when the images are joined, the transition from one image to the next is invisible. It also ensures that the new image has a quality comparable to that of the original images used. Image merging can be carried out by making the seam invisible in the output image. The seam is the line that is visible at the point where the two images overlap.

Methodology:-

Scale Invariant Feature Algorithm:-

The SIFT algorithm (Scale Invariant Feature Transform) proposed by Lowe is an approach for extracting distinctive invariant features from images. It has been successfully applied to a variety of computer vision problems based on feature matching including object recognition, pose-estimation, image retrieval and many others. However, in real world applications there is still a need for improvement of the algorithm's robustness with respect to the correct matching of SIFT features. In this paper, an improvement of the original SIFT algorithm providing more reliable feature matching for the purpose of object recognition is proposed[1]. Following are the major stages of computation used to generate the set of image features:

1. Scale-space extrema detection: The first stage of computation searches over all scales and image locations. It is implemented efficiently by using a difference-of-Gaussian function to identify potential interest points that are invariant to scale and orientation.
2. Keypoint localization: At each candidate location, a detailed model is fit to determine location and scale. Keypoints are selected based on measures of their stability.
3. Orientation assignment: One or more orientations are assigned to each keypoint location based on local image gradient directions. All future operations are performed on image data that has been transformed relative to the assigned orientation, scale, and location for each feature, thereby providing invariance to these transformations.
4. Keypoint descriptor: The local image gradients are measured at the selected scale in the region around each keypoint. These are transformed into a representation that allows for significant levels of local shape distortion and change in illumination. This approach has been named the Scale Invariant Feature Transform (SIFT), as it transforms image data into scale invariant coordinates relative to local features[4].

Random Sample Consensus algorithm-

The Random sample Consensus algorithm (RANSAC) proposed by Fischler and Bolles is a general parameter estimation approach designed to cope with a large proportion of outliers in the input data. Unlike many of the common robust estimation techniques such as M-estimators and least median squares that have been adopted by the computer vision community from the statistics literature, RANSAC was developed from within the computer vision community. RANSAC is a re-sampling technique that generates candidate solutions by using the minimum number observations (data points) required to estimate the underlying model parameters. As pointed out by Fischler and Bolles, unlike conventional sampling techniques that use as much of the data as possible to obtain an initial solution and then proceed to prune outliers, RANSAC uses the smallest set possible and proceeds to enlarge this set with consistent data points[5].

The basic algorithm is summarized as follows:

- 1) Select randomly the minimum number of points required to determine the model parameters.
- 2) Solve for the parameters of the model.

- 3) Determine how many points from the set of all points fit with a predefined tolerance.
 - 4) If the fraction of the number of inliers over the total number points in the set exceeds a predefined threshold τ , re-estimate the model parameters using all the identified inliers and terminate.
 - 5) Otherwise, repeat steps 1 through 4 (maximum of N times).
- An advantage of RANSAC is its ability to do robust estimation of the model parameters, i.e., it can estimate the parameters with a high degree of accuracy even when a significant number of outliers are present in the data set. A disadvantage of RANSAC is that there is no upper bound on the time it takes to compute these parameters. When the number of iterations computed is limited the solution obtained may not be optimal, and it may not even be one that fits the data in a good way. In this way RANSAC offers a trade-off; by computing a greater number of iterations the probability of a reasonable model being produced is increased. Another disadvantage of RANSAC is that it requires the setting of problem-specific thresholds.



Figure 8 : Input image 1

RANSAC can only estimate one model for a particular data set. As for any one-model approach when two (or more) model instances exist, RANSAC may fail to find either one. The Hough transform is an alternative robust estimation technique that may be useful when more than one model instance is present. The figure 7 below depicts key point detection [3].



Figure 9 : Input image 2



Figure 10 : Stitched output image

The image stitcher provides a cost effective and flexible alternative to acquire panoramic images using a panoramic camera. The panoramic images stitched by a stitcher can also be used in applications where the camera is unable to obtain a full view of the object of interest. The full view of the object can be constructed using the image stitcher using overlapping regional images acquired for the object.

C. G-Code processor

G codes are universally accepted standard codes used in CNC machines, 3D printers etc. There are multitude of applications available in the market that manipulate G-codes for specific task at hand. The MATLAB based GUI developed by us does the same function and is compatible with any standard G-code format. The G-code processor code exists partially in both the MATLAB and arduino codes. Inkscape has a comprehensive tool set to edit paths, as they are the basic element of a vector file. It is a free and open-source vector graphics editor, used to create or edit vector graphics such as illustrations, diagrams, line arts, charts, logos and complex paintings. Inkscape can render the primitive vector shapes, text and regions containing raster graphics. It supports image tracing, enabling the editor to create vector graphics from photos and other raster sources. Created shapes can be subjected to further transformations, such as moving, rotating, scaling and skewing.

Inkscape's primary vector graphics format is Scalable Vector Graphics (SVG), while Inkscape can import and export several formats, all editing workflow inevitably occur within the constraints of the SVG format.

D. Serial communication

The communication between the MATLAB front end GUI and the G-Code processor/Motor control algorithm within the arduino is of optimum importance to ensure the proper working of the device. The baudrate was set at 96000 bps but this doesn't ensure the synchronisation between the two parallel program codes. To ensure this a particular loop is set up at the power on in arduino and upon the initialisation phase of GUI. Which keeps on sending a particular character, to the other end and waits for an acknowledgement. Thus problem with initial synchronisation was avoided.

Next issue in communication arise ironically due to one safety feature setup within the MATLAB. Upon sending an

instruction to move according to the G-codes and waiting for an acknowledgement for the movement to complete, a time-out error occurred. In order to avoid that, a particular delay calculation was carried out within the MATLAB and acknowledgement search was done only after the estimated delay.

IV. EXPERIMENTAL RESULTS AND DISCUSSIONS

The proposed standalone prototype had undergone many experiments to note down its adaptability with different physical situation and its tolerance level for different parameters.

The stitching process starts with the matching key points. So to provide sufficient matching points, the amount of overlap between successive images was important. But when the overlap increases, the overhead in the stitching algorithm increases. Also more images, more storage and more time for image acquisition and execution of stitching algorithm is required. Decreasing overlap creates chances of error since a reduction in the matching key points, especially when the external lighting changes. So optimum overlap value which satisfies indoor and outdoor lighting and as well as night environment is required for smooth working of the system. We took 20 image samples of varying texture and color content, such as pixels containing text information as black and white and a few where essentially monochromatic. And subjected them to scanning under varying ambient conditions and overlap. The figure 9 shows the adaptability of the system for various physical situations.

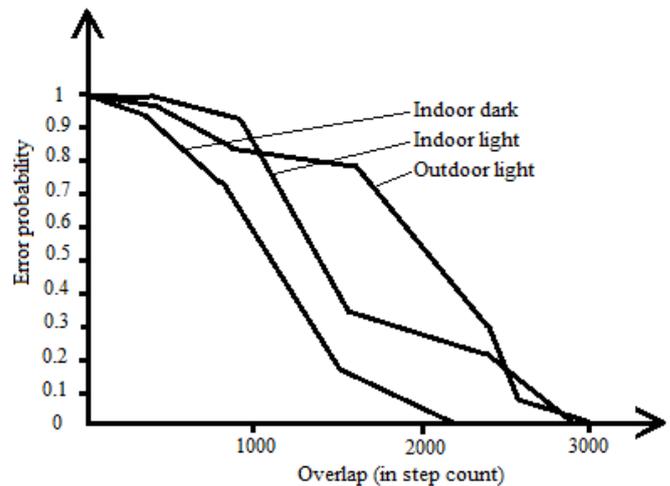


Figure 11: Overlap-Error probability graph

Each curve depicts a plot against overlap and error probability for a specific ambient lighting value.

Tests were carried out to determine the maximum and the optimum speed levels of the motors as well as to calculate the pitch and steps per cm values. The step per cm was determined by drawing a 1000 step line and calculating average values. The maximum step is obtained with an inter-step delay of 0.5ms in the arduino codes. 1ms was selected as the optimum value for drawing, and for the return to origin motion, 0.5ms was used.



V. CONCLUSION

The proposed system is low cost, eco-friendly prototype which can access points in two dimensional spaces, hence works as a plotter and scanner by interchanging the heads. The device works in all environments irrespective of the shape and structure of scanning object and also independent of the environment. Based on the working behind the plotter, the same system can be implemented for PCB printing using a conductive ink. Also the system can be extended for laser engravers and also PCB drills. Coming to the industrial point of view, it can be used for non-destructive testing (NDT) applications.

VI. REFERENCE

[1] Y. Yu, K. Huang, and T. Tan, "A Harris-like scale invariant feature detector," in *Proc. Asian Conf. Comput. Vis.*, 2009, pp. 586–595.

[2] J. M. Morel and G. Yu, "Asift: A new framework for fully affine invariant image comparison," *SIAM J. Imag. Sci.*, vol. 2, no. 2, pp.438–469, Apr. 2009.

[3] J. Rabin, J. Delon, Y. Gousseau, and L. Moisan, "RANSAC: A robust algorithm for the recognition of multiple objects," in *Proc. 3D'PVT*, 2010.

[4] M. Krulakova, Matrix technique of image processing in Matlab, ICC'2010: proceedings of 11th International Carpathian Control Conference, 26-28 May, 2010, Eger, Hungary, Rekatel 2010

[5] Image stitching using MATLAB , Tejasha Patil, Shweta Mishra ,Poorva Chaudhari , Shalaka Khandale International Journal of Engineering Trends and Technology- Volume4Issue3.