A Faster Image Registration and Stitching Algorithm

N. Senarathne, S. Ransiri, A. Balasooriya, P. Arangala, C.R. De Silva and P. Samarakoon
Department of Computer Science & Engineering, University of Moratuwa, Sri Lanka.

Abstract - **This paper presents a new algorithm for Image Registration and Stitching. The algorithm is designed to be extremely efficient and fast in its execution and is intended for use in stitching images extracted from a video stream of a camera. This algorithm is not universally applicable to all the image registration and stitching problems. It is customized to be used to generate single images of surfaces such as a conveyor belt or undercarriage of vehicles, which cannot be captured by a single photo. The algorithm works by extracting edges of the two images to be registered. Then it selects a reference section from the first image and search in the second image where it finds the best match for that section. The best match is the least difference score. The section in the second image which gives the least difference score is then identified as the registration point between the two images. The speed of the algorithm is improved by using heuristics to restrict the search space in the second image. We present full details of how the extraction of the heuristic is done from the inputs and how it drastically reduces the execution time of the algorithm. The implementation of the algorithm was done in MATLAB and the details of the MATLAB functions are listed in the paper. The paper also contains a full section on comparing our algorithm with a set of existing algorithms. Our algorithm outperforms the existing ones for all the common image sizes.**

Index Terms - **Image pyramids, Image registration, Image stiching**

I. Introduction

Registration and stitching of images has always been a very interesting study in the field of Computer Vision. Different ways of registering images have been developed in the past. Most of these algorithms were designed for generic image registration meaning that they can be universally applied to any type of two overlapped images. But the problem with this approach is that the use of these generic algorithms for some of the special applications gives poor performance. For example, a surveillance system employing several cameras to capture the undercarriage of a vehicle has to register many image frames to generate the final image. One notable feature in these images is that the percentage of overlapping is not greater than 50% between two images for most of the time. But the generic algorithms does not take this information into account thus performs redundant operations. Therefore these generic algorithms, even though they give correct results, suffer from performance.

We have devised a new registration algorithm which is capable of registering any type of two images faster. The algorithm is based on extracting a heuristic to reduce the search space, thus removing redundant pixel operations.

Section 2 of this paper discusses the theory behind this new algorithm. Section 3 and section 4 provides an analytical view of this algorithm. It includes details about the basic implementation aspects of the proposed algorithm and a set of statistics that clearly explains and contrasts the new algorithm's performance with existing registration algorithms. Section 4 is followed by the conclusion and the acknowledgments.

A. *Requirement for faster image stitching algorithms*

Image registration and stitching has found its place in many industrial and scientific applications. Some of the popular applications are registering aerial photographs of cities and a more complex case of registering images of the earth's surface generated by geo-satellites [1]. Most of these applications concentrate on the accuracy rather than the performance. But some of the applications require the algorithm to work faster.

B. *Applications which require faster image registration*

Applications such as capturing undercarriage of vehicles for surveillance purposes require the registration and stitching algorithms to work faster. The check points must work faster to reduce the road traffic and a surveillance system employing image registration technique to inspect the undercarriage of a vehicle must produce the final registered images quickly for inspection.

Another example of a requirement for faster image registration algorithm is the artificial intelligence based application to check for unwanted objects in a conveyor belt. In this application, the conveyor belts full length causes problems for capturing the whole area in a single photo. So different image frames from a video stream should be used to capture the whole area and stitch them to represent the whole area in a single image. The registration algorithm must execute faster so that it is ready to capture the conveyor belt again in the next cycle.

Therefore there is a need for faster image registration and stitching algorithm and the algorithm proposed in this paper is vital step towards developing efficient and faster registration and stitching algorithms for industrial applications.

C. *Problems with existing algorithms for image stitching*

The existing image registration and stitching algorithms are designed to be universally applicable to registering of images in any orientation. Most of these algorithms are based on extracting features of the two images and matching the features across the two images [1].

Some of the popular features used in these applications such as lines and curves, detected using Hough Transforms [2] and object corners detected using Harris operator require extensive computational power. Using of these features provides high accuracy, but they cannot be used in high speed stitching applications.

These performance issues of the existing algorithms have increased the need of faster image registration and stitching algorithms for industrial imaging applications.

II. ALGORITHM

A. *General algorithm for registration in the horizontal direction*

The algorithm proposed in this paper is a modification to the image registration algorithm described in [3]. The problem with that algorithm was, even though it was able to correctly register two overlapping images, it required a huge amount of pixel operations. What we have proposed here is to reduce this huge number of pixel operations considerably and thus improve the total running time of the algorithm.

The new algorithm is based on extracting a heuristic from the two images to be stitched (from here onwards these two images are termed as L and R specifying the left image and the right image respectively), and use that to reduce the search space of the algorithm. The method used to extract the heuristic is to scale down the two images L and R by using Image Pyramid technique [4] (to SL and SR respectively) and use the basic algorithm proposed in [3] to register L and R as depicted in Fig. 1.

Downscaling of the image can be done by forming image pyramids. One thing that is important to note here is that, the intent of downscaling is to reduce the number of pixel operations. And it is advised not to use any pixel smoothing technique to compensate for the loss of information by forming image pyramids. This is because; the visual appearance of the scaled down images does not matter. What matters is that, even after downscaling, the basic algorithm should be able to register them correctly. Since the basic algorithm is a feature based registration algorithm, the edges in the images should be preserved even after downscaling. If a pixel smoothing technique is employed, some of the edges might disappear from the downscaled images (since we are using nearest neighbor method for resampling), thus resulting is an incorrect registration of SL and SR.

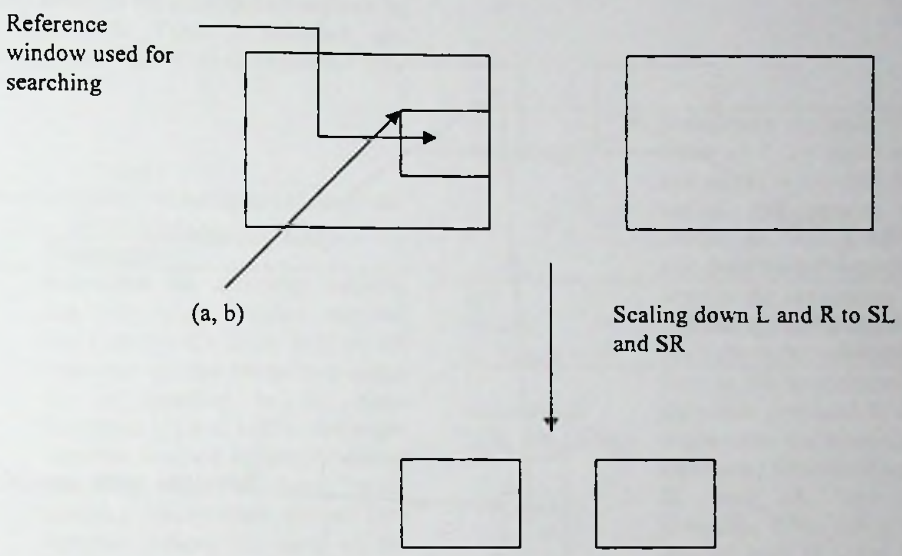

Fig. 1. - Rescaling the two original images to smaller ones. Reducing the size of the two images reduces the pixel operations exponentially thus increasing the performance of the matching operation.

After registering SL and SR, the registration point of this stage (MIN_U, MIN_V) is mapped back to the original two images L and R as shown in Fig. 2. The heuristic for L and R is the mapped registration point identified from the earlier step. This heuristic is used to define a region in image R, say X, which contains the registration point for L and R (An example of choosing an area X is given in the next section when an implementation of this algorithm is discussed). The area X is considerably smaller than the area of the whole image R. Then L and R are registered together, but this time using a search space restricted to the area X. Since it is guaranteed that registration point lies somewhere in X, this algorithm is an optimal algorithm. This algorithm is considerably faster since only area X is searched compared to the whole area of R previously, thus reducing the number of pixel operations dramatically.

B. Algorithm for registering in the vertical direction

This is a modified version of the above described algorithm. In this algorithm, the inputs are assumed to be image frames captured in the vertical direction as opposed to the earlier algorithm, which worked on horizontally aligned images. Here again, the same heuristic method is used to reduce the search space. Rescaling part is identical to the above situation. But, searching for the heuristic and feature tracking is modified and restricted more than the earlier situation and thus improves the performance furthermore. Fig. 3 illustrates the restricted search space in the heuristic extraction phase. In this situation, the only translation possible is in the vertical direction. So there is no need to move the reference section in the horizontal direction for searching. Therefore it dramatically reduces the number of search steps and thus increases the performance. Then the mapping of the heuristic to the two original images is pretty much similar to the earlier case.

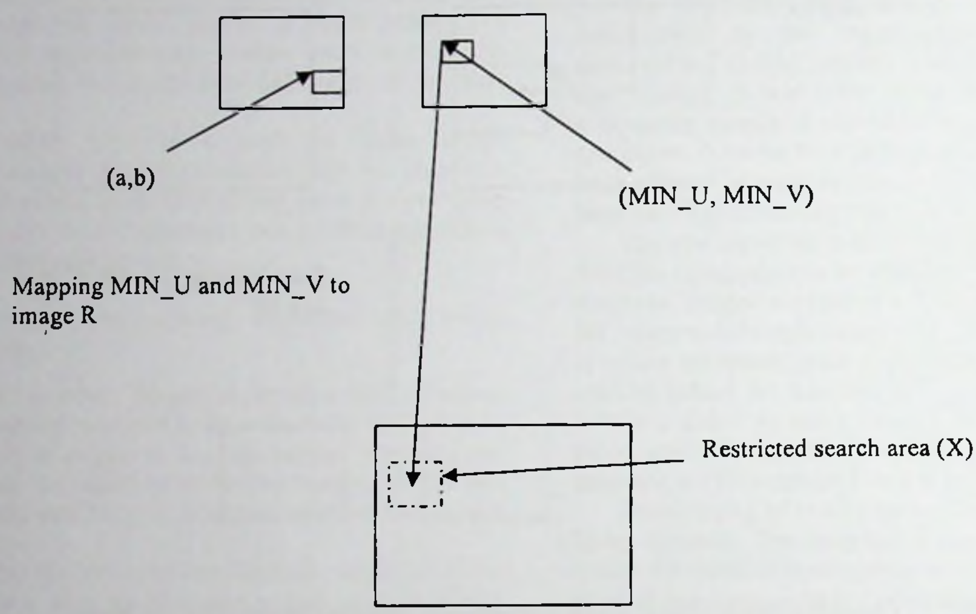

Fig. 2. – Mapping the heuristic to the original image and thus reducing the search space

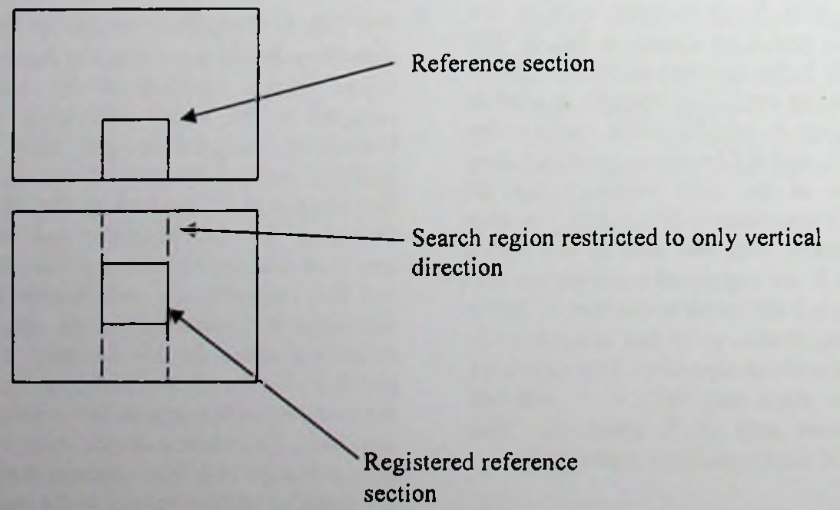

Fig. 3. – Restricting the search space to only a vertical bar when trying to identify the heuristic

III. Implementation

We have implemented the algorithms described in section 3 in MATLAB. Table I describes the MATLAB functions that were used to implement this algorithm.

Table I
Summary of the functions in the implementation of the algorithm

Function	Description
S = calcSimilarity(wd1, wd2)	Calculates the similarity between the two image widows supplied (wd1 and wd2). Here wd1 is the reference window taken from image L, as specified in the basic algorithm [1] and wd2 is the target window selected iteratively during searching stage. "wd1" and "wd2" contains binary edge values. This function returns the sum of the absolute values of differences in each pixel location between wd1 and wd2
[a, b, MIN_U, MIN_V] = imreg(L, R, XRange, YRange)	Register the two images L and R using the restricted search spaced defined by XRange and YRange. This function implements the idea proposed by [1]. XRange and YRange are optional parameters. If they are omitted, a full search is done on the entire image region of R. XRange and YRange are not specified in the heuristic extraction stage. XRange is a structure containing two fields, "start" and "end" which correspond to the restricted search space's starting and ending rows of image R. YRange is similar to XRange, but has the column addresses instead of row addresses. This function iteratively calls calcSimilarity in the area specified and returns the registration point MIN_U and MIN_V extracted from image R which corresponds to the least difference value. "a" and "b" represents the starting point of the reference window taken from image L.
SI = rescaleimg(I)	Scale down the input image I by a factor of 9, by reducing the width and length to 1/3rd of their original values, and returns the Scaled Image SI. MATLAB's imresize also does the same operation.
I = imstch(X, Y, a, b, u, v)	Stitches the two images X and Y by registering the point (a,b) with (u,v) and returns the result as I.
I = imcombine(L, R, fnh_imreg)	This is the implementation of the algorithm proposed in section 2. It implements the heuristic extraction method by first rescaling L and R to SL and SR, and then calls imreg(SL, SR). The result is then mapped back to L and R to get the heuristic. Using these values, the fields in the XRange and YRange structure are updated. Then imreg is called with L, R, XRange and YRange as arguments to have the final output. fnh_imreg is the function handler that points to an implementation of 'imreg'. Several implementations of imreg with minor changes were used in the testing purposes, and their results are listed in section 4
I = vimcombine(T,B)	Similar to "imcombine" in operation but only in the vertical direction
I = vimstch	Similar to "imstch" in operation. Used for stitching in the vertical direction.

`imcombine` uses following code to map the registration points between SL and SR to L and R. Constant values 3 and 6 are due to using of three as the scaling ratio in the image pyramid.

```
XRange.start = MIN_U*3 - 3;
%% If the calculated starting position
%%is less than 1
if(XRange.start < 1)
  XRange.start = 1;
end

XRange.end = XRange.start + 6;
%% If the calculated ending position
%% exceeds the height of IMG2
if(XRange.end + WINDOWHGT > RHGT)
   XRange.end = RHGT - WINDOWHGT;
end

YRange.start = MIN_V*3 - 3;
%% If the calculated starting position %%
is less than 1
if(YRange.start < 1)
  YRange.start = 1;
end

YRange.end = YRange.start + 6;
if(YRange.end + WINDOWLEN > RLEN)
%%If the the calculated ending %%position
exeeds the length of IMG2
    YRange.end = RLEN - WINDOWLEN;
end
```

According to this code, there are only 36 (6x6) possible locations to search for similarity instead of larger number of possible locations when basic algorithm was employed.

IV. Results

The above implementation has several variants. The function `imreg` is implemented in four flavors for extracting statistical information.

TABLE II
DIFFERENT IMPLEMENTATIONS OF imreg

`imreg`	registration algorithm by [1]
`imreg2`	registration by [1] with search space reduced to first half(in the horizontal direction) of image R
`Imreg3`	This method is only used with the algorithm proposed by this paper. It extracts features only from the restricted area in the R image
`Imreg4`	This method is also only used with the algorithm proposed by this paper. It extracts features only from the reference window in image L, instead of the whole image and only from the restricted area of image R

`imcombine` also has two implementations

TABLE III
VARIANTS IN imcombine IMPLEMENTATION

`imcombine`	Calls either `imreg` or `imreg2` on images L and R
`Imcombine2`	Calls `imreg`, `imreg3` or `imreg4` on images SL and SR first, and then calls the same version of `imreg` on L and R in the second step with the range heuristic.

The following table contains the results that we have obtained by running five different stitching algorithms to stitch two images each with dimension 500x487. Fifty samples were taken for each algorithm.

TABLE IV
RESULTS OF RUNNING DIFFERENT VERSIONS OF REGISTRATION ALGORITHMS ON THE SAME TWO IMAGES

Stitching Algorithm	**Mean time to stitch two images (seconds)**	**Standard deviation (σ)**
Basic Algorithm (imcombine + imreg)	137.1	1.49
Basic Algorithm with search space reduced by half (imcombine + imreg2)	47.15	0.04
Heuristic search algorithm (imcombine2 + imreg)	3.32	0.017
Enhanced heuristic search algorithm with partially restricting the feature extracting area (imcombine2 + imreg3)	2.1	0.015
Enhanced heuristic search algorithm with fully restricting the feature extracting area (imcombine2 + imreg4)	1.79	0.016

The following graph represents the actual timing recorded for the above five algorithms with different final image sizes. All of the measurements were taken by taking two images which are equal in dimension.

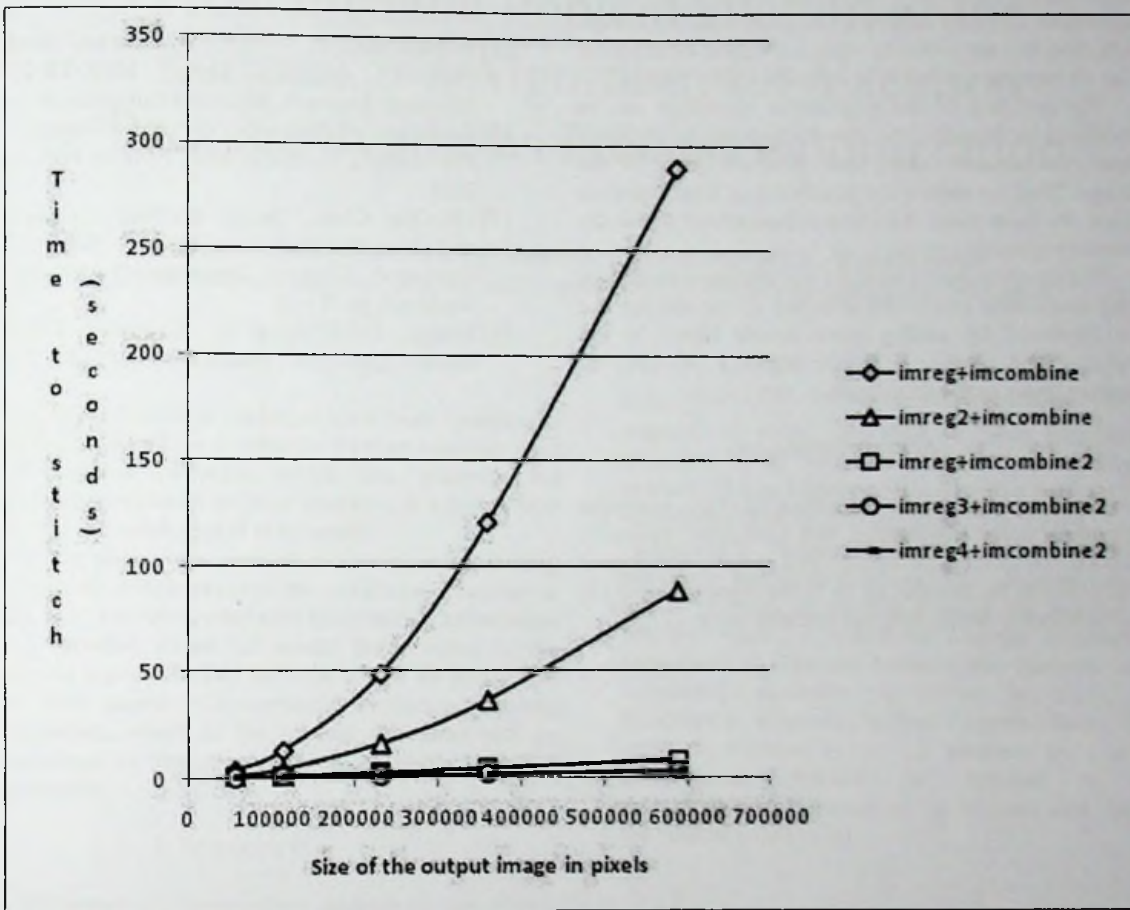

Fig. 4. Comparison of the performance of the proposed algorithm with existing algorithms

V. Conclusion

The basic principle behind making image registration faster is to identify certain heuristics which reduces the search space for the algorithm. We have successfully shown that our algorithm is better for registering images quickly.

As it was listed in table IV, there is a significant reduction in the time to stitch two images by using our algorithm compared to the existing generic algorithm. It also shows that the use of linear operators to extract features only from the required areas also helps to improve the performance of the algorithm.

The Fig. 4 shows that our algorithm perform almost linearly up to all of the common image sizes used so far, compared to the exponential performance of the basic algorithm. As it is depicted in the graph, when the search space is reduced by half, the algorithm's performance doubles, but still it behaves exponentially for larger image sizes. All three algorithms which used heuristics approach, which is suggested in this paper (using imcombine2), behaves linearly up to the common image sizes used.

The bottom line is that use of heuristic approach to reduce the search space for image registration greatly improves the performances of the algorithms.

VI. Future Work

The algorithm was designed only for the translational motion model. But as it is explained in [1], there are many other image motions such as Euclidean motion, affine motion and projective motion. Adapting this algorithm to different motion models in imaging will be an important improvement to this algorithm.

The proposed algorithm works only when the images do not reflect the depth of the objects in the frame. If the camera used to capture the area is close to the objects, the image reflects the depth of the objects and it introduces a curved nature in the corners of the image. Adapting this algorithm to work with these situations is also an important task that is intended by the authors.

The accuracy of the registration algorithm can be increased by repeating the above algorithm for different reference sections taken from different parts of the image. Then we reduce the possibility of ill-registration since we have more than one measurement about the correct registration point.

The image pyramid used for our implementation had only one coarse level. The accuracy of registration can be improved by adding more coarse layers to the pyramid and conduct a feature tracking [4] from all coarser layers to the finer layer.

ACKNOWLEDGMENT

Dr. Chathura De Silva for answering all of our questions regarding image registration and Computer Vision in general and guiding us through this project. Mr. Prasad Samarakoon for introducing us to the research paper [3] and constantly monitoring our research work.

REFERENCES

[1] Richard Szeliski, *"Image Alignment and Stitching, A Tutorial"*, Technical Report MSR-TR-2004-92, Microsoft Research, Microsoft Corporation.

[2] Gonzalez, Rafael C, et al, *"Digital Image Processing"*, Second Edition, Prentice-Hall of India, 2007.

[3] Chia-Yen Chen, *"Image Stitching – Comparisons and New Techniques"*, Technical Report, CITR, Computer Science Department, University of Auckland, pp. 8 - 13

[4] Forsyth, David A, et al, *"Computer Vision – A Modern Approach"*, Prentice-Hall of India, 2006.