# Improving Histogram-Based Image Registration in Video Sequences through Warping

Xuesong Le

School of Information, communication, and Technology, Griffith University

Griffith University, Parklands Drive, Southport QLD 4222 Australia

Ruben Gonzalez

School of Information, communication, and Technology, Griffith University

Griffith University, Parklands Drive, Southport QLD 4222 Australia

*Abstract*—**This paper presents two computationally efficient dynamic-time warping algorithms for image registration in video sequence through histogram based image segmentation. The key idea is to warp the histogram in an input frame to create an ensemble of a reference frame. Any histogram based thresholding method can then be applied to create consistent regions in both the input and the ensemble reference frames. Experiments of the proposed algorithm are used to demonstrate that more consistent matches can found after thresholding.**

*Keywords-component; histogram warping, histogram thresholding; image registration.*

## I. INTRODUCTION

Compared to numerous feature point based algorithms surveyed in [1-3], the use of image segmentation as a step in image registration for video sequences has been little explored due to the computational requirements and the difficulty in obtaining segmentation consistency. Flusser and Dai [4-5] proposed to extract closed regions by contour tracing detected edges. Goshtasby [6] iteratively split an image into regions recursively until the variation of gray-level histogram in each sub-region is small. These region-based segmentation methods are not widely used in image registration of video sequences due to the computational complexity in part created by the post-processing requirements such as edge linking or boundary closing. In contrast, region-based segmentation methods, such as histogram splitting methods are very efficient because they require only one pass through the pixels in the image to perform segmentation. The key parameter in the histogram thresholding process is the choice of the threshold values.

Histogram threshold selection algorithms can be categorized into two groups: shape-based and clustering–based. The first category of methods achieves thresholding based on the shape properties of the histogram. It comprises three stages: recognizing the modes of the histogram, finding the valleys between the identified modes and finally assigning the valleys as the optimal thresholds. One disadvantage of the mode-seeking method is that it may be difficult to identify significant peaks and valleys in the image. Rosenfeld et al. [7] proposed a valley sharpening technique which restricts the histogram to the pixels with large absolute values of derivatives. Cheng and Sun [8] proposed a peak finding algorithm to locate the globally significant peaks of the histogram by removing small peaks, peaks which are too close to adjacent peaks, and peaks if valleys between them are not obvious. In the class of

clustering methods, gray levels are grouped under various clustering criteria. Otsu [9] selected the global optimal threshold by maximizing the weighted between-class variance or minimizing the weighted within-class variance. Kapur [10] considered a multi-level thresholding scheme where the sum of between-class entropies for two classes is maximized.

Recently, Goncalves [11] proposed an image registration method that uses histogram-based image segmentation but it can only handle translation and rotation because it is not robust to histogram changes and image scaling causes more significant change of the histogram. In this paper, we present a novel method that combines both dynamic time warping (DTW) methods and histogram thresholding techniques to solve this problem. The objective of warping two histograms is to minimise the histogram differences across frames, and thus provide higher correlation between images and creates increased segmentation consistency. In our method, the histogram of the second image is warped first to create the ensemble of the histogram in the first image before histogram thresholding. Two efficient dynamic time warping algorithms are proposed here to align the histogram sequences. The direct consequence is that each gray level in the second histogram is mapped to its correspondent in the second histogram. Then histogram thresholding techniques in [7-10] can be used to find optimal thresholds for the first histogram. After that, the same thresholds with a lookup table created during the warping can be applied on the second histogram. Finally, regions in both images are found by applying the thresholds. The significance of proposed histogram warping operation is that more consistent regions are found, which reduces the number of outliers in image registration and improves the registration accuracy.

## II. DYNAMIC TIME WARPING IN HISTOGRAM ALIGNMENT

DTW is a technique that finds the optimal alignment between two time series if one time series is warped non-linearly by stretching or shrinking it along its time axis. This warping operation between two time series can then be used to find corresponding regions between the two time series. In our case, we are dealing with histograms rather than time series and so the DTW problem is expressed as follows: Given two histogram sequences F, and G, both of length 256,

$$F = f_0, f_1, ..., f_i, ... f_{255}, \quad G = g_0, g_1, ..., g_j, ... g_{255} \qquad (1)$$

Construct a warp path W:

$$W = w_0, w_1, ..., w_k, ... w_n \qquad (2)$$

where n is the length of the warp path and the $k^{th}$ element of the warp path is:

$$w_k = (i, j) \qquad (3)$$

where i is an index of histogram sequence F, and j is an index of histogram sequence G. The warp path starts at the beginning of each histogram sequence at $w_0=(0,0)$ and finishes at the end of both sequence at $w_n=(255,255)$. An example of warping histogram F to another histogram G is shown in Figure 1. If both of the histograms were identical, the warp path is a straight line joining $w_0$ ad $w_n$. There is a constraint on the warp path that forces i and j to be monotonically increasing in the warp path.

$$w_k = (i, j), \quad w_{k+1} = (i', j'), i \le i' \le i+1, j \le j' \le j+1 \qquad (4)$$

The minimum distance path, W, between F and G is defined as:

$$Dist(W) = \min(\sum_{k=0}^{k=n} Dist(w_{ki}, w_{kj})) \qquad (5)$$

where $Dist(w_{ki}, w_{kj})$ is the distance between two data point indexes (one from F and another from G) in the $k^{th}$ element of the warp path.

### A. -Dynamic Programming Based DTW

A direct solution to find the optimal warp path in equation 5 is through dynamic programming. A 256x256 two dimension cost function matrix D in figure 1, is constructed first where each element, denoted as D(i, j), is defined in equation 6. Each D(i,j) is the sum of the distance d(i,j) found in the current cell and the minimum of the adjacent elements.

$$D(i, j) = \begin{cases} \min[D(i, j-1), D(i-1, j), D(i-1, j-1)] + d(i, j) & if (i > 0, \wedge j > 0) \\ d(i, j) & else \end{cases}$$
$$d(i, j) = |f(i) - g(j)| \qquad (6)$$

After the cost matrix D is filled, backtracking along the minimum cost index pairs $(i, j)_k$ starting from $(255,255)_0$ to $(0,0)_n$ yields the DTW warping path W. A greedy search is performed that evaluates cells to the left, up, and diagonally to the top-left. Figure 1 below shows an example of a cost matrix and minimum-distance warp path traced through it from D(0,0) to D(255,255). If the warp path passes through a cell D(i,j) in the cost matrix, it means that $i^{th}$ point in histogram F is warped to the $j^{th}$ point in another histogram G.

The major disadvantage of the dynamic programming based DTW is that it requires high complexity of $O(N^2)$. A number of complexity reduction methods [12] have been proposed to speed up by a large constant factor, but those methods still runs in $O(N^2)$ time. To avoid the brute-force dynamic programming, a fast method is proposed in section 2.2 which directly aligns two histogram sequences in O(N) time.
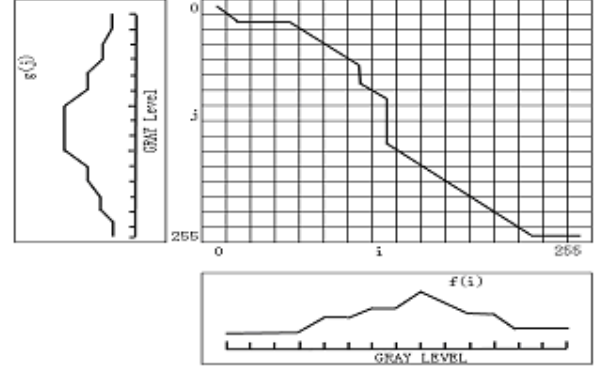


Figure 1: Illustration of warp path for histogram sequences F and G

### B. Integration Based DTW

A faster DTW approach is to use integration. In this approach, the histogram warping problem is described as follows: each gray level i of the second histogram sequence is mapped into a new gray level j in the first histogram sequence, such that the quantity:

$$\left| \sum_{k=0}^{i} cF[k] - \sum_{k=0}^{j} cG[k] \right|, \qquad (7)$$

$$\sum_{k=0}^{i} cF[k] = \sum_{k=0}^{i-1} cF[k] + f[i], \quad \sum_{k=0}^{i} cG[k] = \sum_{k=0}^{i-1} cG[k] + g[i] \qquad (8)$$

is minimized subject to j.

The mapping table can be constructed by a linear algorithm (O(N)) which does numerical integrations simultaneously on both histogram sequences. For each gray level i, in the first sequence, we integrate the specified histogram until a j is reached such that the above condition is satisfied. The integration process exploits the simple recurrence relations as i increases from one to the other. Thus, the overall cost in constructing the mapping table is linear in time.

The C code of warping histogram sequence G with reference to F is essentially a few lines of code:

```
void warp(int cF[], int cG[], int lookupTbl[])
{    int i,j=0;
    for(i=0; i<256; i++)
    {
        while( cF[j]< cG[i] && j<256) j++
```

```
    if( j == 0 )  lookupTbl[i] = 0;
    else if( j>255)  lookupTbl[i] = 255;
    else   // find which one is closer to cG[i]
      lookupTbl[i] = (cF[j] - cG[i]) > (cG[i] - cF[j-1]) ? j-1: j;
  }//End for
}
```

## III. PROPOSED HISTOGRAM THRESHOLDING ALGORITHMS

After the mapping of gray levels in both histograms is established, a histogram thresholding technique is still required to select optimal levels for the histogram in the first image. We propose three representative thresholding algorithms here and use them during experiments. Two of them are extensions of existing methods.

The problem of multi-level histogram thresholding can be stated as following: It assumes that there are M-1 thresholds, $\{t_1, t_2, ..., t_{M-1}\}$, which divide the original histogram into M classes: $C_1$ for $[0,...,t_1]$, $C_2$ for $[t_1+1, ..., t_2]$, ... , $C_i$ for $[t_{i-1}+1, ..., t_i]$, ..., and $C_M$ for $[t_{M-1}+1, ..., 255]$.

### A. Clustering-Based Thresholding

The proposed clustering algorithm is an extension of OTSU's bi-level thresholding method [9] to a multi-level thresholding method. The cumulative probabilities, $\omega_k$, mean, $\mu_k$, and  total variance, $\sigma_k^2$, mean for each class $C_k$ is defined as:

$$\omega_k = \sum_{i=C_k} p_i \text{, } \mu_k = \sum_{i=C_k} ip_i / \omega_k \text{, } \sigma_k^2 = \sum_{i=C_k} (i-\mu_k)^2 p_i / \omega_k \qquad (9)$$

In [11], the optimal threshold t in any class $C_i$ for $[t_{i-1}+1, ..., t_i]$ is the maximum weighted between-class variance, $\sigma_B^2$:

$$\sigma_B^2 = \omega_1 \omega_2 (\mu_1 - \mu_2)^2. \qquad (10)$$

where $\omega_1 = \sum_{j=t_{i-1}}^{t} p_j$, $\omega_2 = \sum_{j=t+1}^{t_i} p_j$, $\mu_1 = \sum_{j=t_{i-1}}^{t} jp_j / \omega_1$, and $\mu_2 = \sum_{j=t+1}^{t_i} jp_j / \omega_2$.

For this work, we have instead used local optimization to reduce processing time to search for the optimal multi-level thresholds. It is a divisive approach, which starts with a single class and recursively splits the single class into M classes. This algorithm can be described as following:

1  Start with a single class $C_0$ which has all intensity levels t=0… 255 and add to a list C with count m =0.

2  Compute total variance $\sigma_0^2$ in class $C_0$ with equation 3 and the optimal threshold t with equation 4.

3  Search each candidate class $c_i$ in list C and find the class $c_s$, which has the largest total variance $\sigma_i^2$ and the optimal threshold

$t_i$. Intensity gray level in class $c_i$ ranges from $t_s$ to $t_{s+1}$, where $0 \le t_1 < ... < t_s < t_{s+1} < ... < L$

4  Split the class $C_s$ into two new classes $C_m$, and $C_n$. Intensity gray level in class $c_m$ ranges from $t_s$ to $t_i$ and ranges from $t_i$ +1 to $t_{s+1}$ in class $c_i$.

5  Compute total variance $\sigma_m^2$, $\sigma_n^2$ and the optimal thresholds, $t_m$, and $t_n$ for two classes.

6  If any new created class $c_j$ has satisfied following criteria $\sigma_j^2 < \sigma_{min}^2 \vee size(c_j) < min\_size$, classify this class as a none-divisible class. Otherwise this class is assigned as candidate class.

7  Remove $c_s$ from list C and add two split classes $c_m$ and $c_n$ to list C. Increment m by 1.

8  If the number of classes including all none-divisible classes and candidate classes has reached maximum threshold M or the largest variance is below some threshold, then stop, else go back to step 3.

### B. Entropy-Based Thresholding

The proposed entropy algorithm is another modification of [10] where the selection of optimal threshold in any class $C_i$ for $[t_{i-1}+1, ..., t_i]$ is defined as the maximum between-class entropy in equation 11.

$$e_B = -\{ \sum_{j=t_{i-1}}^{t} ((p_j/\omega_1)*\log_2(p_j/\omega_1)) + \sum_{j=t+1}^{t_i} ((p_j/\omega_1)*\log_2(p_j/\omega_1))\}. \qquad (11)$$

where $\omega_1 = \sum_{j=t_{i-1}}^{t} p_j$, $\omega_2 = \sum_{j=t+1}^{t_i} p_j$,

### C. Peak-Seeking-Based Thresholding

The selection of thresholds in [8] is to seek most significant peaks through several stages of peak detections and is described as the following steps:

1  Find all peaks.

2  Find significant peaks.

3  Remove any small peaks.

4  If two peaks are close, the peak with bigger value is chosen.

5  Remove a peak if the valley between two peaks is not obvious.

6  The minimum values between any adjacent peaks are selected as the thresholds.

After thresholds of the histogram, F, in the first image are selected, every gray level, $i$, is labeled as one of the following:

$$L = \{l_{0,(0,t_0-1)}, l_{1,(t_0,t_1-1)}, ..., l_{j,(t_{j-1},t_j-1)}, ..., l_{m,(t_{m1},t_m-1)}\}, \qquad (12)$$

$$L_F(i) = l_{j,(t_{j-1},t_j-1)}, \quad t_{j-1} \le i \le t_j - 1, \qquad (13)$$

where $L_f(i)$ represents the j$^{th}$ label value which gray level, *i*, is assigned to and *i* is within the range $[t_{j-1}, t_j-1]$.

The process of assigning labels to gray levels in the second histogram, G, is a lookup operation, described as follows

$$L_G(j) = L_F(lookupTbl[j]), \tag{14}$$

where lookupTbl represents mapping of i$^{th}$ gray level in histogram G to j$^{th}$ gray level in histogram F created in section 2.

Labeling each pixel in the images is another lookup operation which retrieves a label value through its gray level directly. Finally a connected component analysis is performed on both labeled images to extract regions where pixels have been assigned to the same value.

## IV. EXPERIMENT RESULTS

To test performance of the proposed algorithm in histogram-based segmentation, various image sequences in Figure 2 are used in experiments. Each type has been further tested under varying scaling, rotation, and translation camera transformations.



**Fig. 2.** (a) Hall (b) Lounge (c) Ladies (d) Stairs

Testing images for rotation tests were selected every 20 frames from the input video sequences. The maximum frame interval range is 100 frames, approximating 10 degree rotations which are reasonable in video sequences. Images in translation tests were selected every 20 frames among 100 frames as well. The horizontal translation was measured as percentage of amount of translation over the image width. The maximum translation range in the testing sequences was 15%.

In test cases of rotation, and translation, we found that both proposed warped method and non-warped methods were able to provide consistent matched regions because the change of the histogram is limited. Figure 3 shows segmented regions using integrated-warped method, and non-warped method for image Hall_r200 and Hall_r400 under 10's degree rotation respectively.
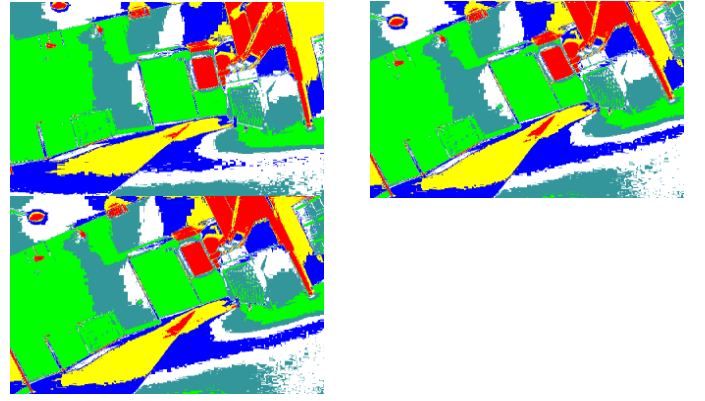


**Figure 3.** (a) segmented regions in Hall_r200 (b) warped segmented regions in Hall_r400 (c) non-warped segmented regions in Hall_r400

TABLE I. MATCHING RESULTS FOR SCALING

| | Ladies | | | Lounge | | |
|---|---|---|---|---|---|---|
| | integ | dp | non-warped | integ | dp | non-warped |
| **scaling ratio 1.3** | | | | | | |
| Peak-SeekingBased | 21 | 24 | 20 | 17 | 20 | 10 |
| Entropy-Based | 28 | 30 | 22 | 18 | 21 | 9 |
| Clustering-Based | 30 | 31 | 24 | 15 | 23 | 13 |
| **scaling ratio 1.4** | | | | | | |
| Peak-Seeking-Based | 19 | 23 | 12 | 15 | 20 | 6 |
| Entropy-Based | 29 | 28 | 15 | 17 | 18 | 10 |
| Clustering-Based | 27 | 28 | 25 | 17 | 20 | 13 |
| | **Hall** | | | **Stair** | | |
| **scaling ratio 1.3** | | | | | | |
| Peak-Seeking-Based | 21 | 20 | 12 | 12 | 12 | 7 |
| Entropy-Based | 17 | 16 | 17 | 10 | 13 | 10 |
| Clustering-Based | 20 | 19 | 16 | 10 | 7 | 10 |
| **scaling ratio 1.4** | | | | | | |
| Peak-Seeking-Based | 18 | 19 | 9 | 11 | 10 | 4 |
| Entropy-Based | 17 | 16 | 15 | 10 | 11 | 10 |
| Clustering-Based | 18 | 20 | 11 | 10 | 7 | 11 |

In test cases of scaling, we selected test images with zooming ratio up to 1.4. Table 1 shows the number of possible matched regions for Hall sequence under 1.3 and 1.4 scaling. The warping methods performed better than non-warped histogram-based methods 95% of the time. And the warping methods achieved matching accuracy on average 40% higher than those non-warped methods. In these experiments, the dynamic programming based DTW, which is 430 times slower

than the integration-based method shown in Table 2, performed only 6% better on average. Experiments results in Figure 4 show segmented regions using dynamic-warped method, integrated-warped method, and non-warped method for image Hall_1 and Hall_4 under 1.4 scaling respectively. Table 2 lists the average running time on above video sequences between dynamic-warped method and integration-based method. Experiments run on a PC with 3.00GHZ Intel Pentium 4 CPU and 1G Memory.

TABLE II.         RUNNING TIME COMPARISON BETWEEN INTEGRATION-BASED DTW AND DYNAMIC PROGRAMMING-BASED DTW

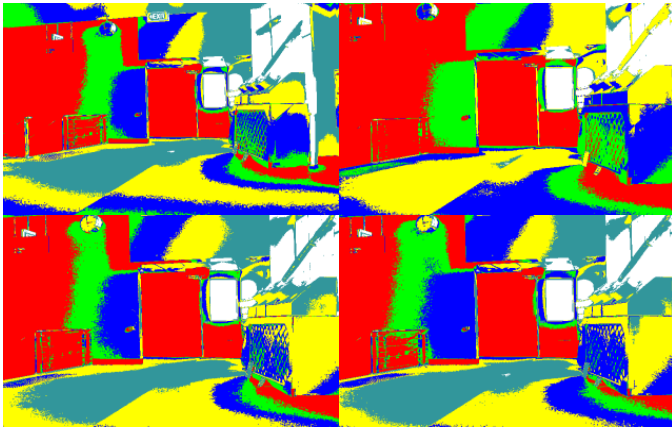|  | Integration based DTW | Dynamic programming based DTW |
|---|---|---|
| comp. time (ms) | 0.0287 | 12.5309 |



**Figure 4.** (a) segmented regions in Hall_1 (b) non-warped segmened regions in Hall_4 (c) dp-warped segmented regions in Hall_4 (d) integ-warped segmented regions in Hall_4

## V. CONCLUSION

This paper has presented methods for improving histogram-based segmentation consistency in video sequences for image sequence registration using a DTW approach. The use of DTW is able to improve segmentation consistency by up to 40%. This is necessary for making histogram segmentation based image sequence registration robust against image scaling. While the classical dynamic programming based DTW marginally outperforms the proposed integration-based approach, it is about 430 times slower. Designed for low computation requirements, the proposed technique is effectively able to handle image sequences with significant histrogram changes due to either scaling or lighting changes. Future work includes application of the technique to region based image sequence registration methods. .

## REFERENCES

[1] L.G.Brown, A survey of image registration techniques, ACM Computing Surveys 24(4), pp.325-376, December 1992.

[2] B. Zitová, Jan Flusser: Image registration methods: a survey. Image Vision Comput. 21(11): 977-1000 (2003).

[3] J. B. A. Maintz and M. A. Viergever, A survey of medical image registration, Medical Image Analysis 2(1), 1998.

[4] J.Flusser, T. Suk, A Moment-based Approach to Registration of Images with affine Geometric Distortion.", IEEE Trans on Geoscience and Remote Sensing, Vol 32. No.2 March, 1994.

[5] X. Dai and S. Khorram, A feature-based image registration algorithm using improved chain-code representation combined with invariant moments, IEEE Trans. Geosci. Remote Sensing, vol.37, Sept. 1999.

[6] A. Goshtasby, G. C. Stockman, and C. V. Page, A region-based approach to digital image registration with subpixel accuracy, IEEE Trans. Geosci. Remote Sensing, vol. 24, no. 3, pp. 390–399, 1986.

[7] J. S. Weszka, R. N. Nagel, and A. Rosenfeld, "A threshold selection technique", IEEE Trans. Comput., vol. C-23, pp. 1322-1326, 1974.

[8] H. D. Cheng and Y. Sun, "A hierarchical approach to color image segmentation using homogeneity," IEEE Trans. Image Process., vol. 9, no.12, pp. 2071–2082, Dec. 2000.

[9] N. Otsu, A threshold selection method from gray-level histograms, IEEE Transactions on Systems Man Cybernet SMC-9, 1979, pp. 62–66.

[10] J.N. Kapur, P.K. Sahoo, A.K.C. Wong, "A New Method for Gray-Level Picture Thresholding Using the Entropy of the Histogram," Graphical Models and Image Processing, 29 (1985) 273-285.

[11] H. Gonçalves, J. A. Gonçalves, and L. Corte-Real, "HAIRIS: A method for automatic image registration through histogram-based image segmentation," IEEE Trans. Image Process., vol. 20, no. 3, pp. 776–789, 2011.

[12] Chu, S., E. Keogh, D. Hart & Michael Pazzani. "Iterative Deepening Dynamic Time Warping for Time Series." In Proc. of the Second SIAM Intl. Conf. on Data Mining.Arlington, Virginia, 2002.