

## Point Feature Matching Adopting Walsh Transform

W. Abd-Almageed, E. ElKonyaly and S. Saraya  
Control and Computers Dept., Faculty of Engineering,  
Mansoura University, Egypt.

### ABSTRACT

This paper introduces a new method to solve the problem of matching correspondent feature points across two images containing a moving rigid object. Two successive time-varying images are used. Edge points are first extracted using a 3x3 Laplacian mask. Walsh transformation is then applied to the feature points in both images. The choice of Walsh transformation in contrast to other orthogonal transforms is a direct result of its computational simplicity and its interpretative meaning in terms of information contained in the spatial domain. Two premises are applied as matching rules. The first involves the speed of object and imaging system, while the other involves the selection of the best match from the set of candidate matches. Unlike other matching techniques, the computational complexity of the proposed technique does not grow up with to the number of detected feature points in either of the two images. This characteristic gives the technique a great flexibility. Experimental results are given and assessed in terms of both accuracy and computational complexity.

**Keywords:** Image Processing, Disparity, Feature Matching, Correspondence, Feature Points.

### I. INTRODUCTION.

Extracting the correspondent feature points out of two time-varying image sequences is one of the central and important problems in computer vision. Surface reconstruction from stereo images requires correspondent points from two images. Automatic navigation systems have to match images to digital maps. The motion estimation problem requires features matches of two or more images.

The main theme in the solution of time-varying feature matching problems is the selection of the most appropriate matching primitives and matching rules in order to identify correct matches. The method of selecting the most effective matching primitives and matching rules should have:

- a- simplicity of implementation.
- b- stability (insensitivity to small variations) of matching primitives.
- c- realistic processing time.

These constitute the key issues for a successful overall matching strategy [1].

The existing methods of matching can be largely divided into two categories; either the computation of flow field [10-12] or the establishment of discrete feature correspondences [13-20]. Weng *et al.* [20] present a point matching method which yields dense flow. In flow-based approaches, problems occur when motion is large. Many authors used feature matching approaches. Medioni and Nevatia [13] establish line correspondences based on relaxation. McIntosh and Much [14] and Liu and Huang [15] propose algorithm for finding line correspondences. Sethi and Jain [16] present an algorithm for finding smooth point trajectories over image sequences. Crowley *et al.* [17] and Deriche and Faugeras [18] present algorithm for tracking lines by using local affine models using four parameters for each line. Most

discrete feature matching methods are based on maximizing the compatibility of 2D attributes and relative 2D locations.

This paper is about finding correspondent pairs of feature points in two successive time-varying image sequences of a scene. The relationship between the viewing positions of the two images is unknown, unlike the case in stereo matching [24]. The objects in the scene may be stationary or moving, therefore we introduce two premises to determine unique matches.

The paper is arranged as follows. In section II, we discuss the Walsh-transformation We begin by a definition of the transform and its properties. The advantages of using Walsh-transformation is then highlighted. In section III, the problem of feature matching in time-varying image sequences and the proposed technique to do that are outlined. In section IV, experimental results and merits of the proposed approach are given. Section V concludes the findings of this paper.

## II. WALSH TRANSFORM

Image processing and image enhancement via transform techniques exploit the redundancy nearly found in all images. The motivation for using transforms other than Fourier is either to reduce the computation time for a given resolution or to increase the resolution without drastically increasing the computation time. Walsh transform has been used effectively to satisfy these requirements [1,2]. Like other real orthogonal transforms, Walsh transform of a 2D square image (N×N) is defined as:

$$W(u, v) = \sum_{x=0}^{N-1} H_f(x, u) \cdot \sum_{y=0}^{N-1} G(x, y) \cdot H_i(y, v) \quad (1)$$

for  $u, v, x, y=0, 1, \dots, N-1$

where

$G(x, y)$  is a 2D function (here, the image intensity function),

$H_f$  and  $H_i$  are the forward and inverse transformation kernels,

and  $(x, y)$  and  $(u, v)$  are the spatial and transform coordinates pairs.

The kernel  $H_f(x, u)$  of the Walsh transform is given by the relation:

$$H_f(x, u) = \frac{1}{N} \prod_{i=0}^{n-1} (-1)^{\beta_i(x) \cdot \beta_{r-1-i}(u)} \quad (2)$$

Where  $N=2^n$  and  $\beta_i(x)$  is the  $i^{\text{th}}$  bit value of  $x$ .

The array formed by the Walsh transform kernel is a symmetric matrix whose rows and columns are orthogonal. These properties, which hold in general, lead to an inverse kernel  $H_i$  that is identical to the forward kernel  $H_f$ .

So, equation 1 can be rewritten as:

$$W(u, v) = \sum_{x=0}^{N-1} H_f(x, u) \cdot \sum_{y=0}^{N-1} G(x, y) \cdot H_f(y, v)^T \quad (3)$$

as  $H^{-1}=H^T = H$ .

A set of mutually orthonormal basis functions, with values +1 or -1 (ignoring the 1/N weight factor) constitutes Walsh transform kernels. Thus we can rewrite equation 3 in a simplified matrix form as follows:

$$W = H . G . H \quad (4)$$

Table 1 shows the values of the Walsh kernel excluding the 1/N weight factor.

**Table 1, Values of Walsh Transform Kernel for N=4**

X	0	1	2	3
u				
0	+1	+1	+1	+1
1	+1	+1	+1	+1
2	+1	+1	-1	-1
3	+1	+1	-1	-1

The coefficients of Walsh transform have the following properties:

1. They are mutually independent, i.e. the coefficients used in matching strategy are mutually independent for any neighborhood around a given feature point.
2. The computation of the coefficients is simple, involving only addition and subtraction operations.
3. Higher order coefficients yield smaller amplitudes and may be considered as matching primitives of less weight. Thus, the use of lower order Walsh coefficients as matching primitives in our proposed technique will enhance its performance.

### III. PROBLEM STATEMENT AND THE PROPOSED TECHNIQUE.

The problem to be discussed in this work is symbolically shown in fig. 1. The fig. shows two images constituting a time-varying sequence for an object  $O$  undergoing unknown motion. The problem is stated as follows: " Given two extracted sets of features  $\{a_0, a_1, a_2, \dots, a_n\}$  and  $\{b_0, b_1, b_2, \dots, b_n\}$  from two images 1 and 2 respectively, find each pair of features  $(a_i, b_j)$  that are the projections of the same physical point in the scene."

The proposed approach consists of four major stages:

*I- Feature Point Detection:*

*Detect distinct feature points.*

*II- Walsh-Transformation Application:*

*For detected feature points, apply Walsh-transformation.*

*III- Matching Premises Application:*

*Using Walsh-matrices calculated from stage II, find candidate matches.*

*IV- Validate a Potential Match:*

*Provides support through consistency check.*

Fig. 2 shows the pseudo code for the proposed technique. The following sections describe the implementation details.

***I- Feature Points Detection:***

A simple edge detector that uses a 3x3 mask is used. The edge detection kernel shown in fig. 3, detects all edges regardless of their orientation. Performing edge detection in one orientation only is only

a matter of removing symmetry from the kernel. Convoluting each image pixel with the detection kernel i.e.,

$$G_E(i, j) = \sum_{p=0}^{N-1} \left[ \sum_{q=0}^{M-1} K(p, q) \times G\left(i + p - \frac{N-1}{2}, j + q - \frac{M-1}{2}\right) \right] \quad (5)$$

for  $0 < i < x$  and  $0 < j < y$

where:  $G(i, j)$  is the gray level of the pixel (i,j) in the original image, and  $K(p, q)$  is the kernel value of the point (p,q), gives  $G_E(i, j)$  as a resultant value from the convolution operation.

In fact, the matching technique presented later, does not depend on any particular feature detector. However, a robust detection of distinct features across images will definitely ensure better matching results. Among the feature detectors that have been reported in literature, corner finding techniques ([21], [22], [23]) are recommended for use with our proposed technique.

### **II- Walsh-Transform Application:**

Using equation 4, the 2D Walsh transform is applied to all feature points extracted from the previous stage. The 2D Walsh attributes are used as matching primitives for the identification of a potential match by providing neighborhood information around located feature point in the image. As stated in section II, only low order Walsh attributes are taken into consideration as they contain larger amplitudes of the transformed image.

Let a 2D neighborhood of a given point at row y and column x in an  $M \times N$  image is as follows:

$$G(x, y) = \begin{bmatrix} g(x, y) & g(x+1, y) & \text{L} & \text{L} & g(x+n-1, y) \\ g(x, y+1) & \text{O} & & & \\ \text{M} & & \text{O} & & \\ \text{M} & & & \text{O} & \\ g(x, y+n-1) & & & & g(x+n-1, y+n-1) \end{bmatrix} \quad (6)$$

where n is the neighborhood size (4x4 in our implementation),

$x=0,1,\dots,M-n$ ,

$y=0,1,\dots,N-n$

and  $g(i,j)$  denotes the gray level of the point (i,j).

At each feature point located in both images, the 2D Walsh attributes are obtained as follows:

$$W_{x_f, y_f} = H \cdot G_{x_f, y_f} \cdot H \quad (7)$$

subscript  $f$  denotes for feature point.

In our formulation, it is noted that the 2D matching strategy does not include those features located at row and column levels greater than  $M-n$  and  $N-n$ . This restriction can be remedied using 1D Walsh transform.

Only the first row values  $W(0, j)$ ,  $j=0,1,2,3$  of the Walsh matrix are the important values as they convey the greater energy quantity of the 2D image signal, so we can discard other values of the Walsh matrix and compute only the first row in the Walsh matrix by proper implementation. This finding reduces the calculation time drastically.

### **III- Matching Premises Application:**

#### **1- Finding all candidate matches:**

The first premise; used in the proposed technique, is simply stated as follows:

“Due to the very high speed of the currently used imaging systems, 30 frames/sec., and the limited velocity of the object or of the camera, the valid match for a given feature point is most probably located in a limited neighborhood around the feature point.” Our experimental results showed that a 17x17 neighborhood is completely sufficient for good matches (refer to sec IV.) This premise is named *the velocity check*. This premise limits the search space for the valid matches, so it will directly reduce the time required for finding the correct match because of reducing the number of candidate matches. We list all found features in the selected neighborhood in image 2 as candidate matches for the tested feature of image 1.

#### **2- Find a Potential Match**

The list of candidate matches found from step 1, is tested to find the best match. The second premise states that “feature point  $f_2(x_2, y_2)$  in image 2 is said to match the feature point  $f_1(x_1, y_1)$  if:

$$\sum_{i=0}^3 (W_{f_2}(0, i) - W_{f_1}(0, i))^2 = \min \left( \sum_{i=0}^3 (W_{f_2}(0, i) - W_{f_1}(0, i))^2, f_2 = 0, 1, 2, \dots, n_{f_2} \right) \quad (8)$$

where  $n_{f_2}$  is the number of features in the selected neighborhood of image 2”. This feature is denoted by  $f_{2_m}$

#### **IV- Validate a Potential Match:**

With the features found from step 2 in stage III, perform the following consistency check,

$$\sum_{i=0}^3 (W_{f_{2_m}}(0, i) - W_{f_1}(0, i))^2 = \min \left( \sum_{i=0}^3 (W_{f_2}(0, i) - W_{f_1}(0, i))^2, f_2 = 0, 1, 2, \dots, n_{f_1} \right) \quad (9)$$

where  $n_{f_1}$  is the number of features found in image 1 when applying velocity check to  $f_{2_m}$ .

## **IV- EXPERIMENTAL RESULTS AND IMPLEMENTATION DETAILS**

The computer used in the experiments of this work is IBM-PC compatible machine with P-60 processor, 8MB RAM and 1 MB video memory. All images used were of size 128x128 and 320x200 pixels with a resolution of 8-bits per pixel. All images were obtained from the anonymous file transfer protocol server of the university of Massachusetts (vis-ftp.cs.umass.edu).

Limited memory availability is one of the major problems arises when implementing image processing applications on PCs. The simulation programs are coded with C language and WATCOM C/C++ compiler that supports DOS4/GW DOS extender. Thus, the processor is able to switch into its protected mode. Working in the protected mode makes all system RAM available to use by our software. This gives a great flexibility to the developed software which contains huge structures of image data, Walsh transform attributes and feature locations. This could not be performed without WATCOM C/C++ compiler.

The new technique is applied to a large variety of time-varying image sequences. Two experimental results are presented here. We visually check the results by watching the alternating frames of matched features.

Two threshold values were set when applying the proposed technique namely; the threshold value of the edge detector and the size of the neighborhood of the *velocity check* premise. Our experiments showed that appropriate values for these thresholds are 50 and 17x17 respectively. In the following, we discuss the results from experiments.

Figures 4 and 5 show the images sequences used in our experiments. Fig. 4a and 5a show four images of the image sequences on which the technique is applied. Images are of indoor scenes, of 128x128 size. Extracted edge feature points from the first two successive frames are shown in figs. 4b and 5b. The matching results between features of the first two frames are shown in figs. 4c and 5c. Tables 2 and 3, summarizes the performance results of the proposed technique. The new technique is also applied to other pairs of images as well as images of size 320x200 of the same sequences and the same results are obtained.

**Table 2, Results of the applied technique to Cones sequence.**

Frame Pair	First Frame Features	Second Frame Features	Matched Features	Matching Time
1,2	1434	1537	772	2.1
2,3	1537	1512	226	2.53
3,4	1512	1611	1256	1.93

**Table 3, Results of the applied technique to Lense sequence.**

Frame Pair	First Frame Features	Second Frame Features	Matched Features	Matching Time
1,2	1220	1287	794	1.33
2,3	1287	1511	921	1.45
3,4	1551	1331	434	1.88

## V- PERFORMANCE EVALUATION

A series of simulation experiments was carried out in order to compare the proposed algorithm to the algorithms presented in [1 2]. Table 4 shows the results obtained by the algorithm presented by M. Adjouadi, F. Candocia and J. Riley [1, 2].

**Table 4, Results of the algorithm presented in [1],**

Image Sequence	Features Extracted	Features Matched	Execution Time (Sec.)
Synthetic	520	520	4.74
Stapler	886	718	4.77
Computers	5336	3726	8.94
Cup & Sunglasses	1721	1309	5.24

By comparing the above results, to the results of our proposed algorithm in tables 2 and 3, we can note an improved performance by applying the proposed algorithm. It's also important to note that, the algorithm proposed in [1], was applied to stereo pairs of images, which limits the search space for the correct match to a single direction. This is because of the priory knowledge of the two cameras alignment. This of course reduces the computational time dramatically. In our algorithm, we work on two successive images with no apriori knowledge about the camera position in the two time instants. It's obvious from tables 2,3 and 4 that a better performance is obtained by applying the proposed algorithm.

Also, comparing the results to the results obtained by applying the algorithm proposed in [25] that, our proposed algorithm gives higher number of detected features per image which increases the number of matched features between the pair of images.

A higher number of matched features gives better performance when using to solve the problem of motion parameters estimation. This is because it reduces the effect of noise in images even with no prior filtration. Table 5 shows some results obtained in [25].

Sequence No.	Number of features in image 1	Number of features in image 2	Number of matched Features.
1	302	302	147
2	301	302	128
3	244	255	56
4	195	186	77

**Table 5, Results of the algorithm of [25].**

## VI-CONCLUSION

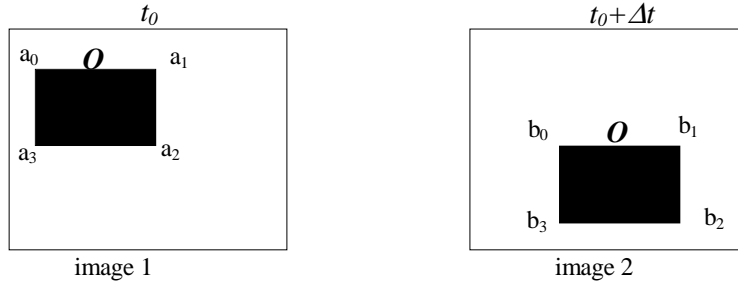
We have developed a new method which matches feature points across time-varying image sequences. In our method we used the merits of Walsh transform which gives a reduced processing time in contrast to other transforms. We have also presented the results of two experiments to demonstrate the advantages of our technique. Although feature points were defined as edge points in this paper, other types of features could be used which gives flexibility to the proposed technique. By considering corner points as feature points, better performance of the method could be obtained.

Parallel implementation of the technique will, of course, reduces dramatically the computational time required. This can be accomplished by paralleling feature extraction and Walsh transform calculation stages.

## REFERENCES

1. Adjouadi, Frank Candocia and John Riley, "Exploiting Walsh-Based Attributes to Stereo Vision," *IEEE Trans. Signal Processing*, vol. 44, no. 2, pp. 409-420, Feb. 1996.
2. Adjouadi and Frank Candocia, "A Stereo Matching Paradigm Based on the Walsh Transformation," *IEEE Trans. Pattern Anal. and Machine Intell.*, vol. 16, no. 12, PP. 1212-1218, Dec. 1994.
3. S. Huang and Arun N. Netravali, "Motion and Structure from Feature Correspondence: A Review," *Proc. IEEE*, vol. 82, no. 2, PP. 252-268, Feb. 1994.
4. M. Haralick and Linda G. Shapiro, "Computer and Robot Vision," *Addison-Wesely Publishing Co.*, 1993.
5. Dick Oliver *et al.*, "Tricks of The Graphics Gurus," *SAMS Publishing*, 1993.
6. J. Awcock and R. Thomas, "Applied Image Processing," *Macmillan New Electronics Series*, 1995.

7. Ibrahim Sezan and Reginald L. Lagendijk, "Motion Analysis and Image Sequence Processing," *Kluwer Academic Publishers*, 1993.
8. G. Beauchamp, "Walsh Functions and Their Applications," *Academic Press*, 1975.
9. David W. Murray and Bernard F. Buxton, "Experiments in the Machine Interpretation of Visual Motion," *The MIT Press*, 1990.
10. Horn and B. Schunck, "Determining Optical Flow," *Artif. Intell.*, 17, 1981, PP. 185-203.
11. Black and P. Anandan, "A Model for the Detection of Motion Over Time," *Proc. ICCV, Osaka, Japan*, 1990, PP. 33-37.
12. F. J. L. Baron, "Performance of Optical Flow Technique," *Proc. IEEE Conf. on Computer Vision and Pattern Recog., Champaign, IL*, June 1992, pp. 236-242.
13. Medioni and R. Nevatia, "Matching Images with Linear Features," *IEEE Trans. Pattern Anal. and Machine Intell.*, PAMI-6, Nov. 1984, pp. 675-686.
14. McIntosh and K. Much, "Matching Straight Lines," *Computer Vision, Graphics, Image Process., Pattern Recog.*, 43, 1988, pp. 386-408.
15. Liu and T. Huang, "Determining Straight Line Correspondences for Intensity Images," *Pattern Recog.*, 24, 1991, pp. 489-504.
16. Sethi and R. Jain, "Finding Trajectories of Feature Points in a monocular image sequence," *IEEE Trans. Pattern Anal. and Machine Intell.*, PAMI-9, 1987, pp. 56-73.
17. S. J. Crowley and C. Discours, "Measuring Image Flow by Tracking Edge-Lines," *Proc. Int. Conf. Computer Vision*, 1988, pp. 658-664.
18. Deriche and O. Faugeras, "Tracking Line Segments," *Proc. European Conf. Computer Vision*, 1990, pp. 259-268.
19. Venkateswar and R. Chellappa, Hierarchical Feature Based Matching for Motion Correspondence," *Proc. IEEE Workshop on Visual Motion*, Oct. 1991, pp. 280-285.
20. A. J. Weng and T. S. Huang, "Matching Two Perspective Views," *IEEE Trans. Pattern Anal. and Machine Intell.*, PAMI-14, Aug. 1992, pp. 806-825.
21. A. Zuniga and R. M. Haralick, "Corner Detection Using Facet Model," *Proc. Computer Vision, Pattern Recog.* 7, 1983, pp. 30-3.
22. Kitchen and A. Rosenfeld, "Gray-Level Corner Detection," *Pattern Recognit. Lett.*, vol. 1, pp. 95-102, 1982.
23. Singh and M. Shneier, "Gray Level Corner Detection: A Generalization and A Real Time Implementation," *Computer Vision, Graphics, Image Processing*, vol. 51, pp. 54-69, 1990.
24. ElKonyaly, S. Saraya and Y. Atwa, "Competitive Stereo Correspondence Framework," *SPIE Conf. Oct. 1995*, vol. 2588, pp. 519-527.
25. Sull and N. Ahuja, "Integrated Matching and Segmentation of Multiple Features in Two Views," *Computer Vision and Image Understanding*, Vol. 62, No. 3, Nov. 1995, pp. 279-297.



**Fig. 1, Two images of an object  $O$  undergoing unknown motion showing two sets of features  $a_i, b_j$  extracted from the two images.**

```

BEGIN
  Detect Image Point Features.
  DO
    IF The Current Point is a Point Feature
      THEN Calculate Walsh-Transform for The Point
    WHILE NOT End of Image Points
      DO
        Apply Velocity check
        Get the Set of all Candidate Matches
        Validate Match.
      WHILE NOT End of Image Features
  END

```

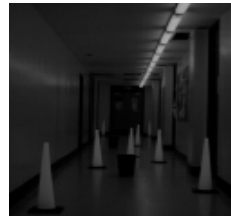
**Fig. 2. Pseudo code representing the proposed algorithm**

0	-1	0
-1	4	-1
0	-1	0

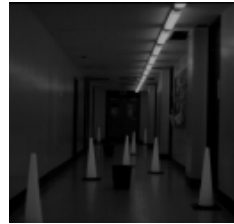
**Fig. 3, Edge Detection Convolution Kernel.**



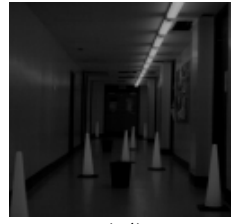
(a1)



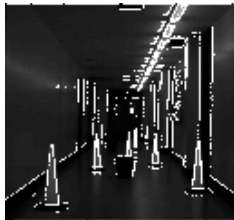
(a2)



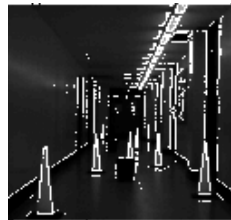
(a3)



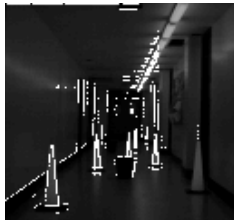
(a4)



(b1)



(b2)

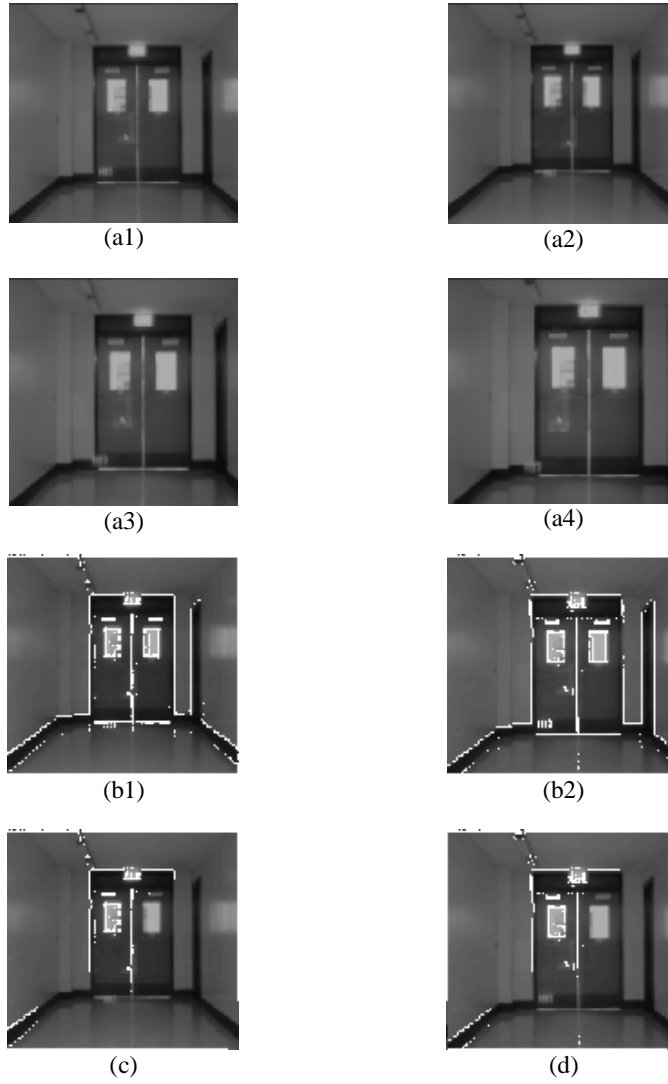


(c)



(d)

**Fig. (4).** (a1, a2, a3 and a4) show four frames of image sequence no. 1, (b1 and b2) show extracted edge feature points from the first two frames and (c and d) show matched point features between the two frames.



**Fig. (5).** (a1, a2, a3 and a4) show four frames of image sequence no. 2, (b1 and b2) show extracted edge feature points from the first two frames and (c and d) show matched point features between the two frames.