PERGAMON

Mechatronics 11 (2001) 409–437

MECHATRONICS

Marker detection and trajectory generation algorithms for a multicamera based gait analysis system

M. Shahid Shafiq[a], S. Turgut Tümer[b,*], H. Cenk Güler[c]

[a]*Bertec Corporation, 6185 Huntley Road, Suite B, Worthington, OH 43229, USA*
[b]*Mechanical Engineering Department, Middle East Technical University, Ankara 06531, Turkey*
[c]*Department of Mechachinal Engineering, The Ohio State University, West 18th Avenue, Columbus, OH 43210, USA*

Abstract

The current gait analysis methodology requires tracking of markers placed on the body segments of the lower extremities, by using multiple camera images while the subject is walking. This study was aimed at the development of application software for the multicamera marker based gait analysis system established at the Middle East Technical University. The main functions of this software are pixel grouping, marker detection and generation of three-dimensional (3D) marker trajectories. The software removes noise and artefacts, and effectively detects the physical markers which may be closely spaced and even overlapped on some camera images. Then, the images in different cameras are matched by utilising the concept of epipolar lines, and 3D co-ordinates of markers are determined for each matching group at every field. Finally, matching of each marker in successive fields is accomplished by employing a set of extrapolation algorithms, thus yielding the required 3D marker trajectories. This paper presents theory, algorithms and implementation of the developed software, as well as an assessment of its performance. In addition to gait analysis for clinical and research purposes, the developed algorithms can find application in sports, 3D computer animation and motion analysis of mechanical constructions.

* Corresponding author. Tel.: +90-312-210-2544; fax: +90-312-210-1266.
E-mail address: tumer@metu.edu.tr (S.T. Tümer).

0957-4158/00/$ - see front matter
PII: S0957-4158(00)00026-X

Keywords: Gait analysis (software); Multicamera marker based motion analysis; Pixel grouping and marker detection; 3D reconstruction; Motion tracking; Trajectory generation; Image matching; Epipolar lines

1. Introduction

Gait analysis refers to the quantitative description of all the mechanical aspects of walking [1]. In general, clinical gait analysis is performed by collecting the motion data synchronously with kinetic data, and this information is then used in model based calculations to yield quantitative description of gait in the form of time-distance parameters and variations in joint angles, joint moments and joint powers. In addition, muscle activation levels measured by electromyography (EMG) provide useful supplementary information for the assessment of gait. Kinetic data, usually measured by force plates, can be acquired directly without much difficulty using commercially available equipment. Acquiring motion data, or data describing the position of various body segments as a function of time, constitutes the major task in gait analysis. Over the years, different techniques have been used for this purpose and now the automatic motion measurement by using video–computer based systems appear to represent the state-of-the-art technology [2]. Typically, spherical markers covered with reflective tapes are attached on various anatomical landmarks and a number of cameras, ranging from 4 to 7 for a three-dimensional (3D) analysis, observe these markers while the subject walks over a pathway which embeds two force plates. Light emitted by infrared light sources, consisting of light emitting diodes (LED) and placed near the cameras, is reflected by the markers and is detected by the cameras. The computer interfaced with cameras tracks the position of each marker during gait, and provides motion data for various body segments by utilising biomechanical models.

At Middle East Technical University (METU), we have established our own gait analysis facilities, called *Kiss* (*Ki*nematic *s*upport *s*ystem), using off-the-shelf equipment. Hardware consists of six Ikegami CCD cameras, two Bertec force plates and one Bortec Octopus EMG unit. Specially developed infrared light strobes are mounted around camera lenses with filters, so that only the infrared light reflected from 2.5 cm diameter spherical markers covered with retro-reflective material can be viewed. Motion of the markers is captured at 50 Hz with 752 (horizontal) $\times$ 291 (vertical) pixel resolution, by using even and odd fields of the interlaced camera separately. A synchronisation unit developed multiplexes the camera images on-line, and a frame grabber is used to transfer these images to the memory of the host computer. Along with the camera images, the force plate and EMG signals are also sampled synchronously by an A/D converter card. The equipment also includes a camera linearisation grid, four calibration rods

suspended from the ceiling, and a surveyor's telescope for their adjustment. The system is supported by two user-interfaced software. Data acquisition software *Kiss*-DAQ performs camera linearisation and calibration tasks, records and stores camera images in synchronisation with each other and with the force plate and EMG measurements and processes the image data to provide 3D marker trajectories. After on-line recording and pre-processing of image data, subsequent off-line processing terminates with the labelling of the marker trajectories by the system operator. The second software *Kiss*-GAIT accepts synchronously generated

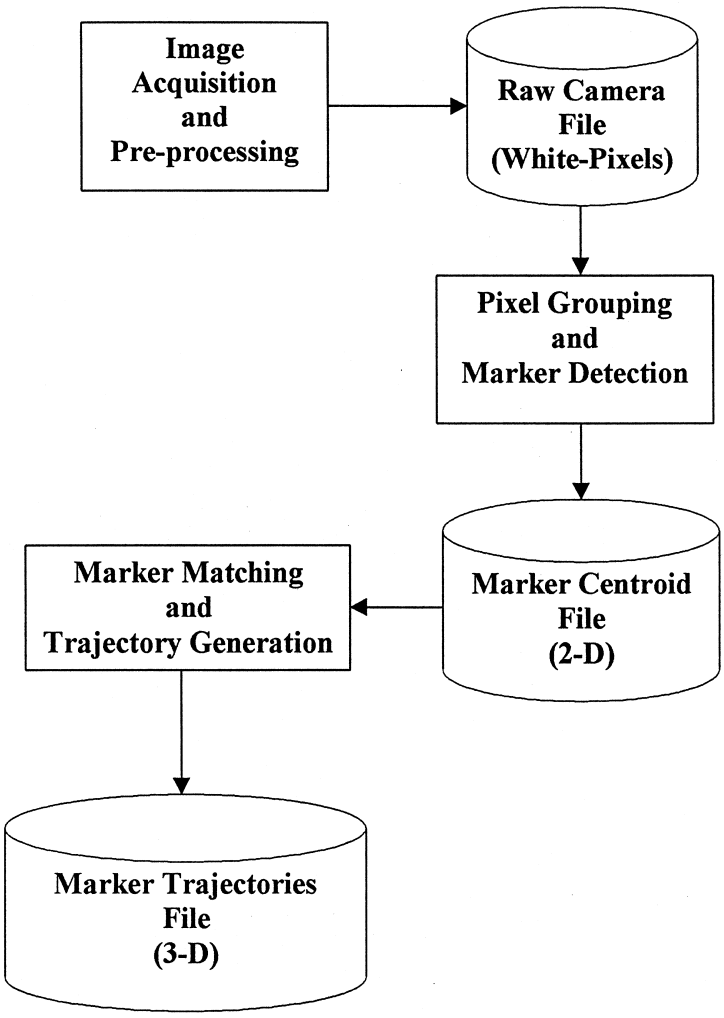


Fig. 1. Operation of *Kiss*-DAQ.

marker trajectories and force plate and EMG records as input and presents quantitative gait data through model based calculations.

Fig. 1 illustrates the operation of *Kiss*-DAQ for image processing. The first part of *Kiss*-DAQ operates on-line and pre-processes the acquired images of individual cameras to give digital image information in the form of white-pixels. The next step involves automatic detection of markers and determination of 2D co-ordinates of their centroids in each camera image. Finally, spatial and temporal matching of marker centroids among cameras and successive fields is accomplished and 3D marker trajectories are obtained.

This paper reports on the methods used in *Kiss*-DAQ for pixel grouping, marker detection and trajectory generation. Existing systems use either hardware or software solution for marker detection [3]. We have developed a software which retrieves data in the form of (x, y) co-ordinates of white-pixels in every image field and groups pixels that belong to the same marker image. Upon removal of noise and artefacts, centroids of marker images are calculated. Our trajectory generation algorithm involves 2D spatial matching of detected marker centroids based on the concept of epipolar lines, followed by 3D temporal matching via spatial extrapolation.

2. Pixel grouping and marker detection

We initially adopted the obvious and simple method of nearest neighbourhood

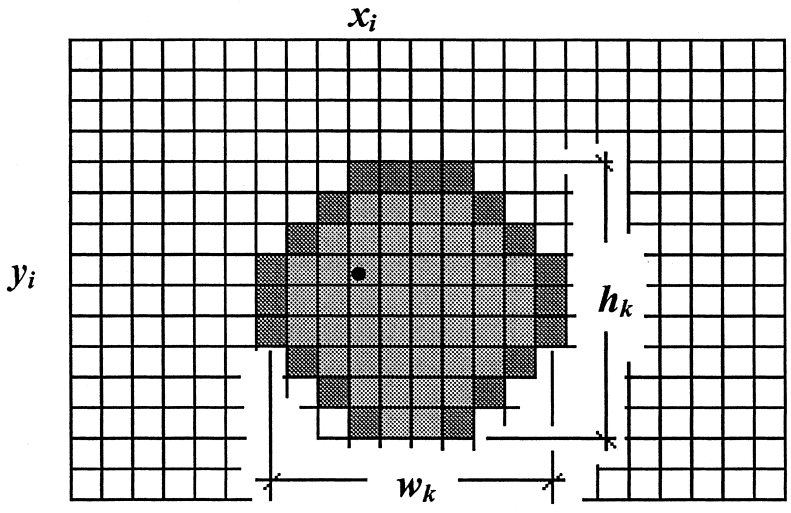


Fig. 2. Pixel grouping in a neighbourhood.

for pixel grouping and marker detection. The available white-pixels were collected in different groups so that they may be identified as part of a specific marker, by using a predefined value for the range of neighbourhood. The grouped pixels were further analysed to remove the groups created due to artefacts or noise present in the digital image.

Considering that the pixels in a group g_k belong to a marker blob (Fig. 2), the height and width of that marker (in pixels) were calculated as,

$$w_k = \max(x_i) - \min(x_i); \qquad h_k = \max(y_i) - \min(y_i); \quad i = l, \ldots, n \tag{1}$$

where n is the total number of pixels in group g_k, x_i is the x co-ordinate of ith pixel, and y_i is the y co-ordinate of ith pixel. The markers that were wider and/or higher than the pre-set thresholds were split to make the size of the marker within an acceptable range. The thresholds for width and height were calculated adaptively for each field, based on the average of widths and heights of all the marker groups in that field. Then the co-ordinates of the centroid of an identified marker, which are used to represent that marker's image, were calculated from:

$$x_c = \frac{\sum_{i=l}^{n} x_i}{n}; \qquad y_c = \frac{\sum_{i=1}^{n} y_i}{n} \tag{2}$$

Tests conducted using the above method revealed that the results were satisfactory for well separated markers. For closely spaced markers, such as those on the feet, the method was susceptible to mis-grouping. With the application of marker size limit, larger blobs were divided harshly into smaller ones so that each image was within the size bounds. This produced many unwanted pixels as being candidates for a marker image. Moreover, the program was not efficient, so it became necessary to alter the algorithm to improve and enhance its performance.

The present algorithm employs a method in which a preliminary grouping of pixels is accomplished while scanning is being performed. Grouping during scanning is based on the closeness of mid-points of boundary pixels in every horizontal scan line. In this way, a number of groups each containing a number of mid-point co-ordinates are generated. These preliminary groups are then evaluated based on successive tests for noise removal and advanced analysis. This approach proved to be much more effective and fast with the computational time being reduced to almost half of the former method. In what follows the methods employed for preliminary grouping, noise removal and advanced analysis are briefly outlined, for which the detailed algorithms can be found in [4].

2.1. Preliminary grouping

During scanning, in addition to the boundary pixels P_i and P_{i+1} on each horizontal scan line, co-ordinates of their mid-points are also stored (Fig. 3). A

simple mean is used to calculate the mid-point co-ordinates:

$$x_{\mathrm{m}} = \frac{x_{P_i} + x_{P_{i+1}}}{2}; \qquad y_{\mathrm{m}} = \frac{y_{P_i} + y_{P_{i+1}}}{2} \tag{3}$$

At the beginning of the process, the mid-point co-ordinates of the first two boundary pixels of the first horizontal scan line containing white-pixels (P_1 and P_2) are assigned to the first group (g_1). Successive mid-point co-ordinates are compared with the values stored in previous groups, and if both x_{m} and y_{m} are sufficiently close to those stored in any of the previous group, the pair is also stored in that group. This process is performed for each horizontal scan line and as soon as a pair of boundary pixels is obtained, mid-point co-ordinates are calculated, compared with available groups and assigned to the matched one. If the new mid-point co-ordinates do not match with any of the previously available groups, a new group is created and assigned a new (next in series) group number.

2.2. Noise removal

The following criteria are employed to identify noise pixels:

- A group with width and height less than specified limits (former method)
- Lonely pixels
- Pixel groups with one pixel width
- Pixel groups with one pixel height

The last three criteria are implemented in the new scanning algorithm and they exhibited more effective noise removal as compared to the previous technique, which uses only the first criterion. The identified pixels are marked as noise and removed.

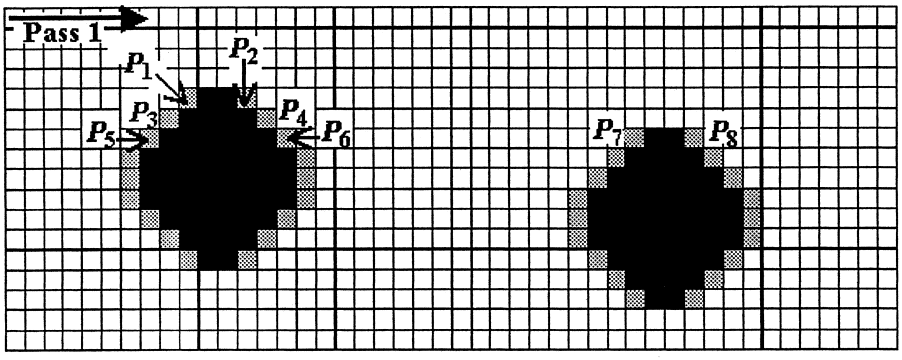


Fig. 3. Preliminary grouping during scanning.

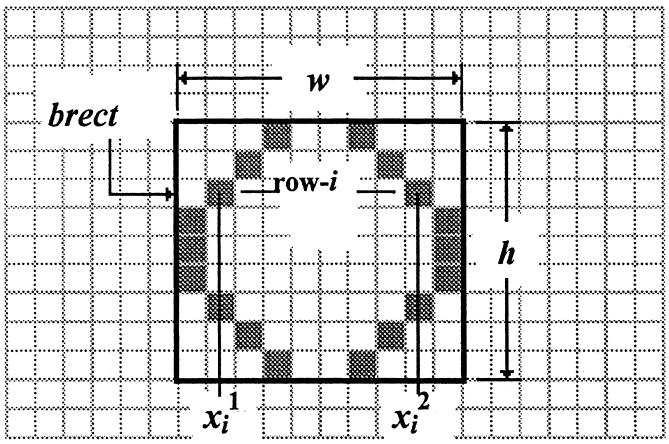


Fig. 4. Bounding rectangle (*brect*) and pixels in a group.

2.3. Advanced analysis

Upon noise removal, additional tests are performed to improve the reliability and exactness of the detected groups. In addition to the centroid (x_c, y_c), these tests require the bounding rectangle, *brect*, of width (w) and height (h), to be defined for each detected group (Fig. 4). This enables to find the total number of pixels (*TotalPixels*) that a group can have and the number of white-pixels (*WhitePixels*) in that group:

$$TotalPixels = w \times h \tag{4}$$

$$WhitePixels = \sum_n \left(x_i^2 - x_i^1 + 1\right) \tag{5}$$

where x_i^1 and x_i^2 are the boundary co-ordinates of each element (row of pixels) in

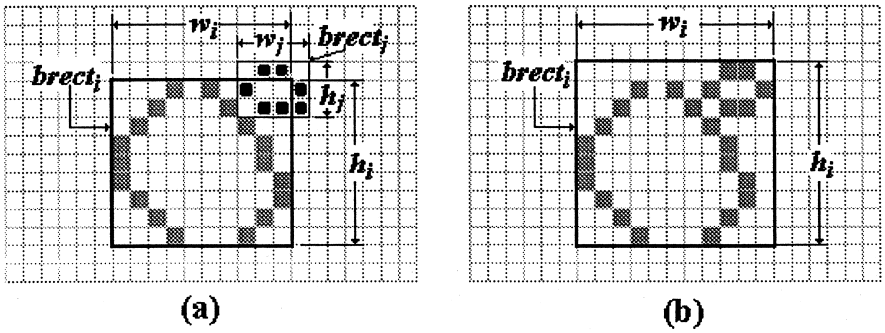


Fig. 5. Merging of sub-marker groups.

a group. These parameters are used in the following fine-tuning tests performed for each group.

2.3.1. Sub-marker detection

Small marker groups which (almost) lie within the bounding rectangle of another larger group are merged with the big group to create a single group. Fig. 5(a) shows a typical example. Most of $brect_j$ lies within $brect_i$, hence, they are merged to create a single group as shown in Fig. 5(b).

A marker is declared to be small if it is smaller than a specified size and is composed of only two rows. This additional condition prevents losing any useful information while tracking the motion.

2.3.2. Overlapped markers

Two kinds of overlapped groups frequently arise; upper-left (*ul*) and upper-right (*ur*) types as illustrated in Fig. 6. In both the cases, two separate markers are included in a single group. The existence of such groups are indicated by a small *WhitePixels*/*TotalPixels* ratio, and the following equations are used to detect the kind of overlapping:

$$\mathrm{GrpOvlp}_{ul} = \begin{cases} abs\left(x_m^l - x_m^u\right) > abs\left(x_m^u - x_1^1\right) \\ abs\left(x_m^l - x_m^u\right) > 3 \\ \left(x_2^1 < x_1^n\right) \vee abs\left(x_2^1 - x_1^n\right) < w \end{cases}$$

$$\mathrm{GrpOvlp}_{ur} = \begin{cases} abs\left(x_m^l - x_m^u\right) > abs\left(x_{\mathrm{m}}^{\mathrm{u}} - x_1^1\right) \\ abs\left(x_m^l - x_m^u\right) > 3 \\ \left(x_1^1 > x_2^n\right) \vee abs\left(x_2^1 - x_1^n\right) < w \end{cases} \tag{6}$$

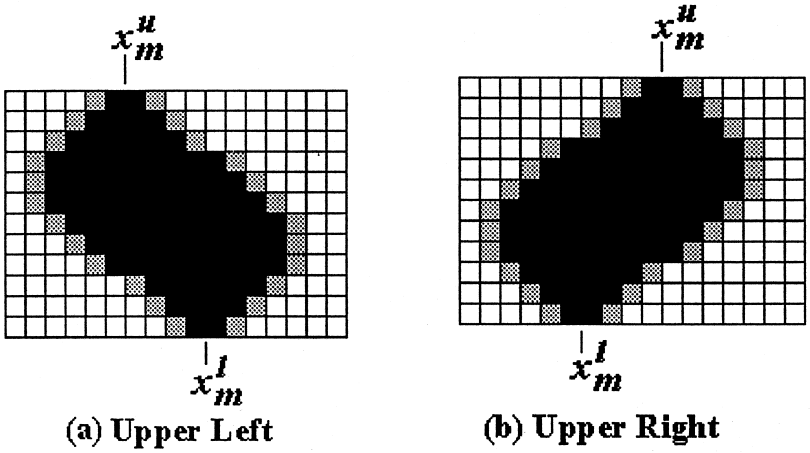


Fig. 6. Two kinds of marker overlapping.

where x_m^u and x_m^l are the mid-point co-ordinates of the top (first) and the bottom (nth) rows of the bounding rectangle ($brect$) of the identified group, and subscripts 1 and 2 refer to the boundary pixels of a row in the group.

If overlapping exists, two new groups are created with x_m^u and x_m^l as their mid-point x-co-ordinates. For an upper-left type of overlapping, the new groups have $2(x_m^u - brect \cdot x_1)$ and $2(brect \cdot x_2 - x_m^l)$ as their respective widths. In case of an upper-right type of overlapping, the groups have $2(brect \cdot x_2 - x_m^u)$ and $2(x_m^l - brect \cdot x_1)$ as their corresponding widths. The original group is deleted afterwards. Fig. 7 shows the envelopes of two markers that were separated with the above technique.

2.3.3. Long/wide test

The height-to-width ratio of a particular group g_k ($ratio = h_k/w_k$) is used to test if the group is long or wide:

$$\text{if } ratio > \max R \Longrightarrow g_k \text{ is Long}$$

$$\text{if } ratio < \min R \Longrightarrow g_k \text{ is Wide}$$

where max R and min R are the pre-defined upper and lower limit. Then these groups are broken down into as many groups as suitable such that each group has the ratio in the specified range. An example of a wide and a long marker is portrayed in Fig. 8. It should be noted that the long and wide markers represent a special configuration of overlapped markers, but the algorithm treats them separately and in turn.

2.3.4. Marker size

Contrary to the former scanning method, in the current algorithm marker

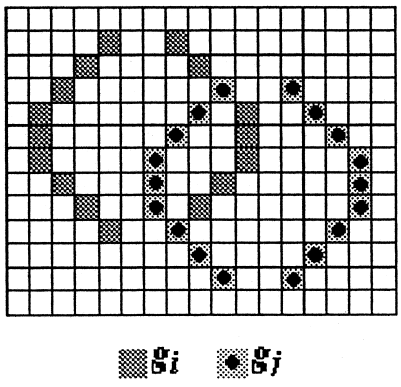


Fig. 7. Detected markers after overlap test.

groups are divided based on the height-to-width ratio rather than imperious marker size limit. Therefore, before imposing the size limit, it is assumed that the detected groups have their height to width ratio within the satisfactory range. Hence, if any group is found to be larger than a pre-specified marker size limit, instead of breaking it, that group is marked as an invalid marker. Large markers generally correspond to the images of LED light sources seen by facing cameras, and are thus eliminated. Very small sized groups are also marked as invalid. Invalid markers appear coloured on the display and are not exported to motion tracking algorithm.

3. Sample tests on marker detection

Pixel grouping and marker detection algorithm requires a set of thresholds to be specified, and its performance is highly dependent on a proper choice of these parameters. Exhaustive tests have been conducted to arrive at suitable threshold values, specifically for the equipment at hand and gait analysis conditions. The program has an added feature of user-specified parameters, enabling him/her to carry out experiments to find suitable values for different applications too.

Four test cases are chosen in order to demonstrate the effectiveness of the marker detection algorithm. The first is an artificially created image field demonstrating the functions performed by each test, especially for closely spaced markers and infrared strobe illumination. The remaining tests are taken from an actual gait analysis procedure.

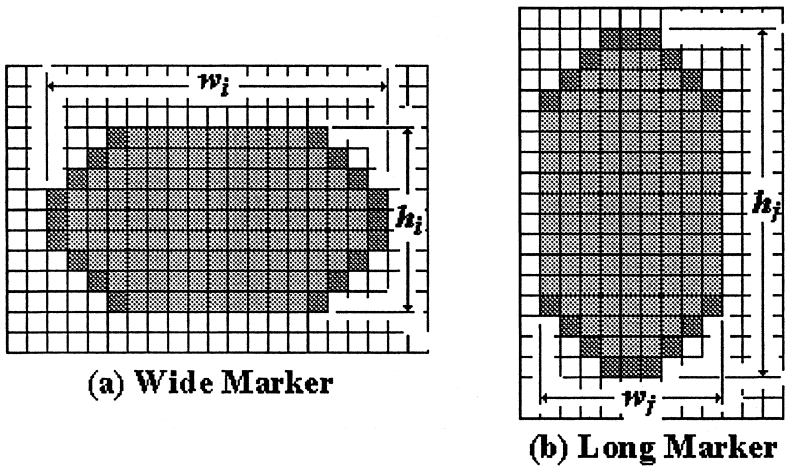


Fig. 8. A wide and a long marker.

3.1. Manually created image

The image shown in Fig. 9(a) was created manually to demonstrate the functions of different tests in the advanced analysis. When advanced analysis was not performed on this 12-marker image, the algorithm identified 10 markers as shown in Fig. 9(b). Note that a cluster of three markers was obtained for two overlapping markers in the original image. After enabling sub-marker test, this cluster reduced to a single marker and a total of eight markers were detected as shown in Fig. 9(c). When all the advanced tests were enabled, except marker size

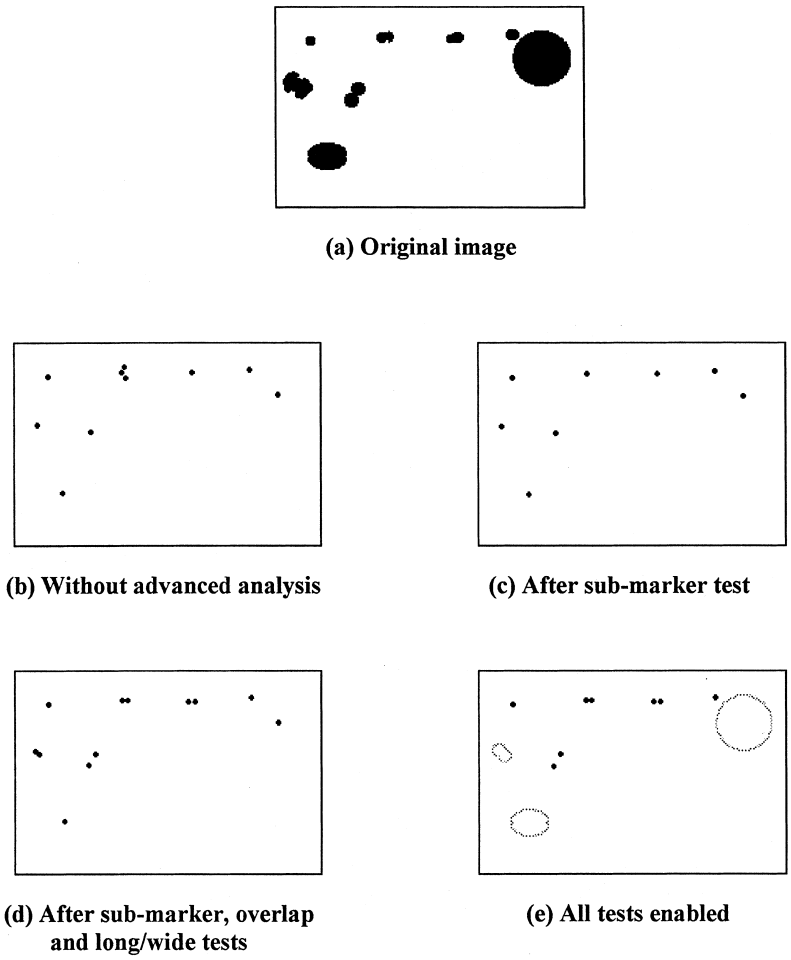
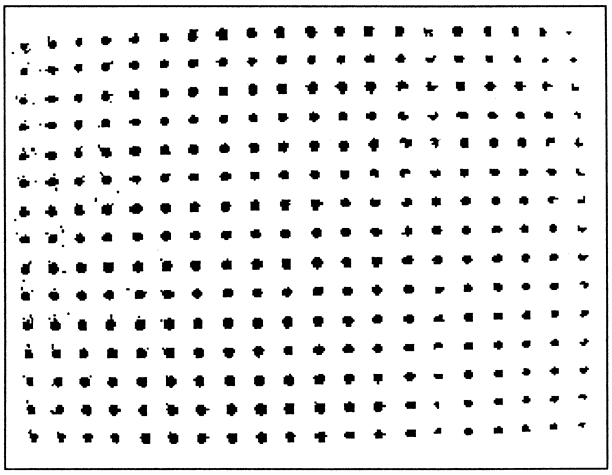
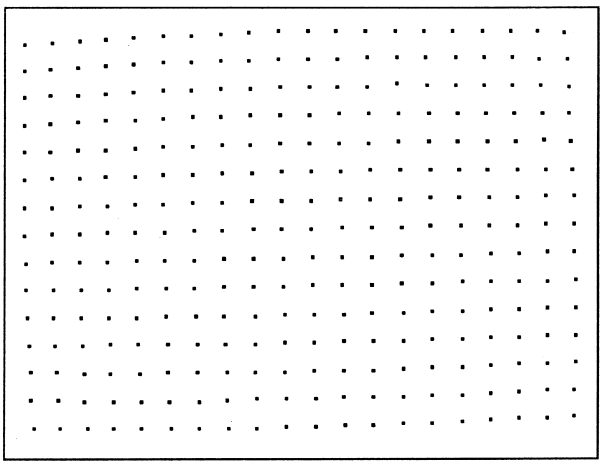


Fig. 9. Artificially created image with 12 markers.

limit, the algorithm identified the 12 markers as shown in Fig. 9(d). Fig. 9(e) resulted when the size limits were also imposed and four markers were detected as invalid markers. The algorithm could deal with this artificial situation which contains some awkward marker arrangements.



(a) Original image



(b) Identified 300 markers

Fig. 10. Detection of linearisation grid markers.

3.2. Linearisation grid image

Camera linearisation is a standard procedure in gait analysis used to correct the lens distortions. To accomplish this task, a planar grid containing 300 equally spaced retro-reflective points arranged in rows and columns (15×20) is viewed. This has to be a close-range shot, while keeping the same focal length adjusted for a very much larger field of view. Fig. 10(a) shows the original image of the grid by one of the cameras, which has a barrel type distortion due to the nonlinearity of the lens. The deviations from straightness of grid lines are utilised to calculate the parameters of a linearisation filter. All the 2D co-ordinates are then passed through this linearisation filter before generating the 3D co-ordinates. This requires the co-ordinates of grid points to be detected by each camera. The program identified exactly 300 marker centroids, as illustrated in Fig. 10(b), indicating the conformity of the identified markers with the original image despite the noise due to improper focal adjustment.

3.3. Calibration frame image

The 2D data is related with its 3D counterpart by camera calibration parameters. These parameters are calculated based on known 3D co-ordinates of reference markers in a calibration volume. The volume is enclosed by four hanging rods with six markers attached on each. The markers are placed in a unique configuration on each rod so that a particular rod can be automatically recognised. The measured co-ordinates of these markers in a laboratory reference frame and their images are used to find the calibration parameters, and the results of the motion analysis are expected to be valid only within the calibrated volume.

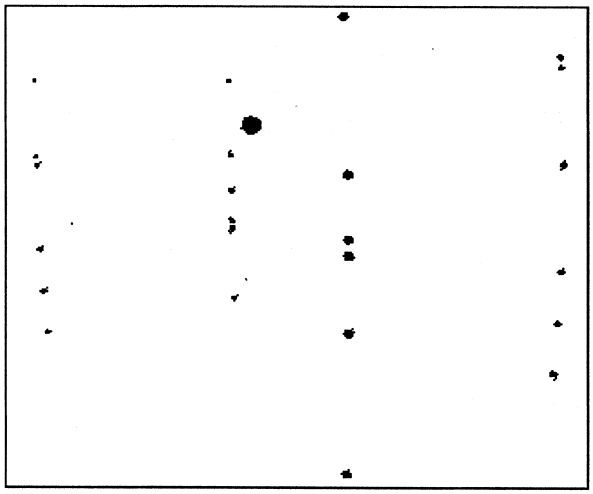
In Fig. 11(a), calibration volume as seen by one of the cameras is shown, and Fig. 11(b) is the result of marker identification algorithm. A total of 25 markers were detected, the large extra marker being the infrared strobe of the opposite camera shown in light grey colour. This was detected by the program as an invalid marker. The valid detected markers were therefore exactly 24.

Although all the markers are of the same size, different depth of view results in different size marker images. Furthermore, some reference markers appear very close to each other. The algorithm can successfully deal with these situations. Subsequent calibration, linearisation and 3D calculations regenerated the reference marker co-ordinates with no more than 0.6 mm mean error in a control volume of size 1.4 m $\times$ 2.4 m $\times$ 2 m, which reflects the marker detection accuracy under static conditions.

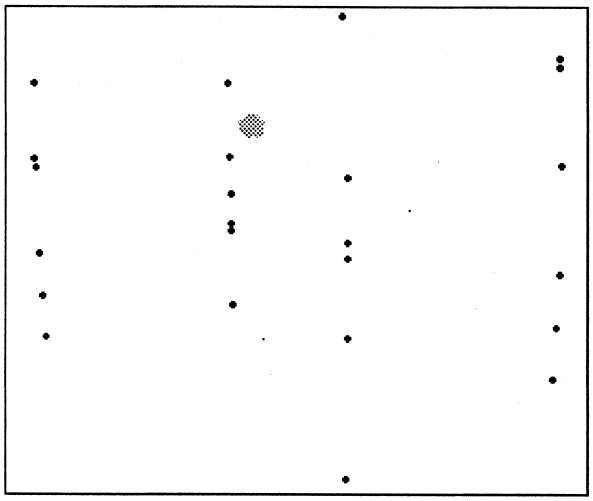
3.4. An Image during gait

Finally, pixel grouping and marker detection algorithm was tested while performing a sample gait trial of a subject with 15 markers placed. The test

duration was 2 s and images were sampled at 50 Hz. A total of 148 image fields were required to be scanned. The motion data (all 148 fields) was scanned in less than 2 s on a computer with a Pentium II processor. Five views obtained from one of the cameras are illustrated in Fig. 12, which shows only the region enclosing all the markers. The algorithm was able to detect all the markers in all the fields, free from noise and artefacts, including the two overlapping markers at



(a) Original calibration volume image



(b) Identified calibration rod markers

Fig. 11. Detection of calibration markers.

the bottom–left corner of 80th field. Such a situation arises frequently, and the effectiveness of tracking algorithm depends highly on identifying overlapping markers.

3.5. Discussion

The pixel grouping and marker detection algorithm presented in this paper worked satisfactorily and efficiently for the purpose of a clinical gait analysis. We have been conducting exhaustive tests on the kinematic data collection software of our gait analysis system. So far, one extra marker was detected in only four image fields, and these were because of invalid division of some strange shaped marker images which appeared due to improper camera positioning and poor illumination. In only one case, the number of detected markers was one less than the actual ones. This happened because two marker images were so completely overlapped that the program could not identify them as being two superimposed marker images. The marker tracking algorithm, however, could tolerate such a level of mis-detection due to redundant number of cameras used for 3D motion analysis. The methods employed may also be useful for other image analysis applications involving identification of some target points from a cluster of white-pixels.

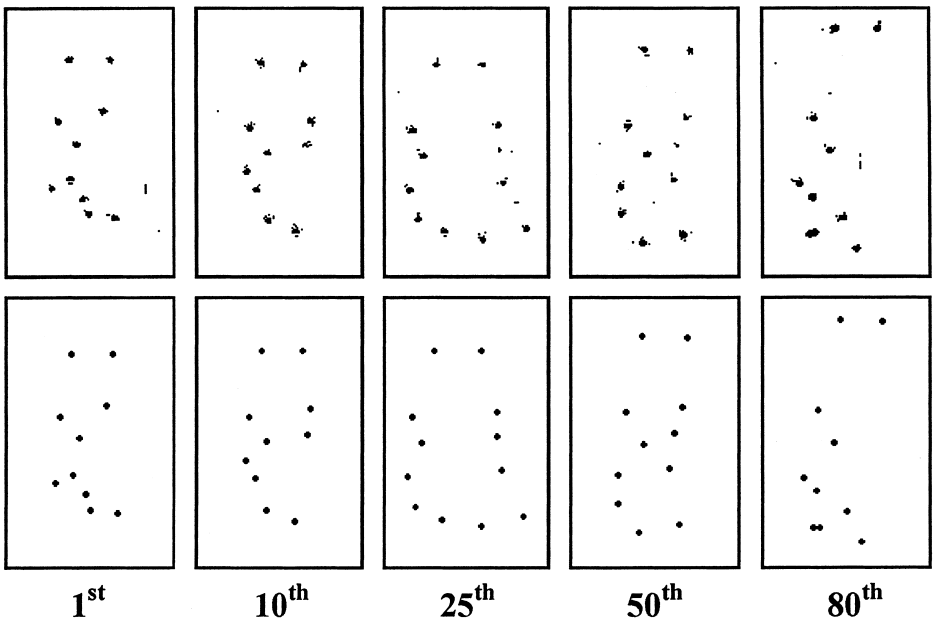


Fig. 12. Markers detected during gait on five fields of one of the cameras.

4. Motion tracking and trajectory generation

One of the main issues in trajectory generation is to accomplish matching of the marker centroid co-ordinates that are provided by the pixel grouping and marker detection algorithm described above. The available marker dots are not associated with specific physical markers, and two types of matching are required to determine the 3D paths followed by the markers during motion. Temporal, or field-to-field (FTF) matching is the identification of one marker's image in a camera in successive fields, whereas the spatial, or camera-to-camera (CTC) matching associates marker centroids among corresponding images in different cameras. The two kinds of matching are illustrated in Fig. 13, which depicts images of two cameras (r and s) in two successive fields (i and $i+1$). In temporal matching, the aim is to associate image points in field-i ($P^r_{i,1}$, $P^r_{i,2}$, $P^r_{i,3}$) to their corresponding images (if any) in field-$i+1$ ($P^r_{i+1,1}$, $P^r_{i+1,2}$, $P^r_{i+1,3}$), as shown by thin connecting lines. The spatial matching involves finding the associations (if any) among image points of camera-r ($P^r_{i,1}$, $P^r_{i,2}$, $P^r_{i,3}$) and camera-s ($P^s_{i,1}$, $P^s_{i,2}$, $P^s_{i,3}$) at a given field, which is shown by thick connecting lines. Both of these matching can be accomplished either in 2D image co-ordinates or in 3D world co-ordinates. There is also an option of applying both the matching simultaneously, or consecutively in any sequence.

In open literature, there are only a few detailed descriptions of the methods employed for marker trajectory generation. As far as we are aware, no method has been reported which performs both the matchings simultaneously. This may

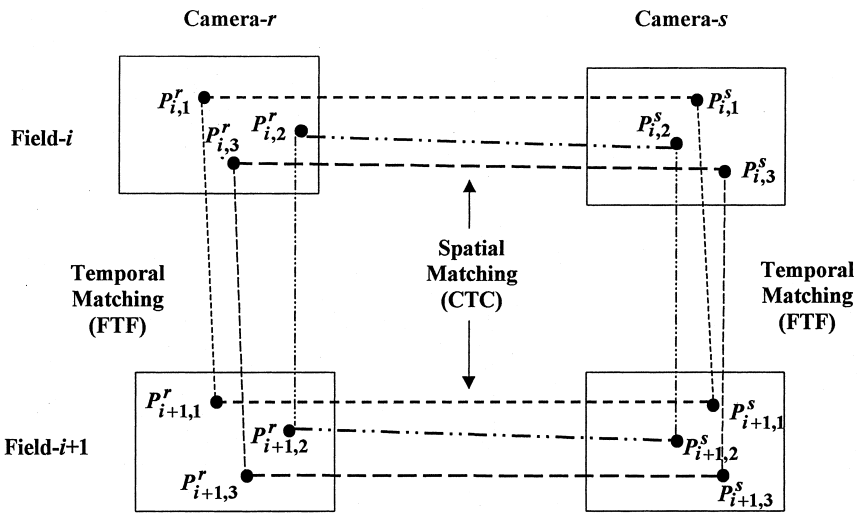


Fig. 13. Temporal (FTF) and spatial (CTC) matching between two cameras for two successive fields.

be attributed to the fact that the number of physical markers to be processed in a typical gait session is no less than 15,000, and the number of possible combinations would be outnumbered if both the matching are handled simultaneously. Sabel [5] reported on the method currently used for the PRIMAS system developed at the Delft University, The Netherlands, which performs 2D temporal matching followed by 3D spatial matching. In this approach, tracking is achieved in the 2D domain by treating cameras independently, and spatial matching is accomplished between 2D trajectories by the subsequent 3D reconstruction program. However, the program requires some manual intervention by the user to deal with unmatched trajectories left after the automatic matching, which is based on a residual table prepared for all the camera pairs and tracing of the matching history.

We had initially developed a similar trajectory generation program based on the 2D temporal matching followed by 3D spatial matching, for our two-camera and VTR system which was the predecessor of the *Kiss*. In a two-camera system, all the markers must be observed by both cameras, otherwise 3D co-ordinates can not be calculated. Therefore this system was restricted to unilateral study and to a specific camera configuration which ensured visibility of all markers.

The 2D temporal matching involved an initial search for two consecutive fields containing images of all the markers that had the minimum average movement. The process began from those two consecutive fields and proceeded first forward and then backward in time to cover all the fields. The nearest neighbours on the two overlaid images readily gave matching markers in the first two fields. For each matched marker pair in the initial two fields, a two-point, 2D linear extrapolation was performed to predict the location in the next field [3]. Subsequent predictions of the new positions of the markers were calculated using the tracked image points from the three previous fields, by using a modified version of the previous extrapolation method; three-point 3D linear least squares approximation. To associate markers with the predictions, an association window was placed around each prediction. If two or more markers did lie within an association window, the proximal marker was selected for association with the prediction. Alternatively, if no marker lied within an association window, it was assumed that the marker had been occluded for this field, possibly due to swinging arms. In that case, the extrapolation algorithm filled the gaps by the generated predictions. This 2D matching procedure was repeated for the second camera.

The 3D spatial matching utilised the following Direct-Linear-Transformation (DLT) equations introduced for camera calibration [6]:

$$(a_{11} - a_{41}u)X + (a_{12} - a_{42}u)Y + (a_{13} - a_{43}u)Z - (a_{14} - u) = 0$$

$$(a_{21} - a_{41}v)X + (a_{22} - a_{42}v)Y + (a_{23} - a_{43}v)Z - (a_{24} - v) = 0 \tag{7}$$

where a_{ij} represent calibration parameters of the particular camera, (u, v) are the

image co-ordinates and (X, Y, Z) are the world co-ordinates of a marker. Knowing the camera calibration parameters from the DLT calibration software, four DLT equations for a candidate matching marker pair of the two cameras can be written in terms of three unknowns (X, Y, Z). Spatial matching algorithm employed three of these equations to calculate the world co-ordinates for all matching combinations and a residual from the fourth equation. The combination which gave the minimum residual was selected as the matched pair. Theoretically, this matching is sufficient for only one field. However, the process was repeated for all the fields to verify the previous 2D matching as well as to resolve ambiguities, such as more than one or no matching was obtained. On rare occasions, this verification led to the modification of the 2D trajectories, which were wrongly branched at a field, where they became very close to each other.

This algorithm worked satisfactorily for the two-camera system without necessitating any user interference, which was set as our target. However, it was a complete failure when modified for the six-camera system. The main reason for the failure was the discontinuous 2D tracks due to inevitable occlusions in the bilateral study in which the number of cameras that see a marker vary drastically. No field of any camera contained all the physical markers for the initiation of the 2D temporal matching process previously applied for the two-camera system. It was therefore necessary to begin the process by assumed matching, and introduce refinements in a loop as more information is obtained in later fields. Moreover, when the 3D spatial matching method was modified for six cameras, the number of possible combinations increased by factorial, rendering it impractical. To summarise, the old version could work satisfactorily for well separated markers, which was achieved in unilateral study, but not so in a bilateral study involving more markers. It was therefore decided to adopt a new approach to trajectory generation and the algorithm presently used in the *Kiss*-DAQ has been developed.

The new algorithm employs 2D spatial matching, followed by 3D temporal matching. For 2D spatial matching the concept of epipolar lines is used. Image matching by the use of epipolar lines has been dealt with in photogrammetry [7]. Although photogrammetric applications deal with static poses, the same principles can readily be used, since recorded motion is simply a sequence of static images. Sabel [5] qualitatively discussed the use of epipolar lines for the 2D spatial matching of marker images, and concluded that its computational cost might be unnecessarily high. Instead, he suggested two new methods, one based on the epipolar line slopes and the other on the epipolar plane angles, but did not report on their implementation. According to Sabel [5], 2D matching with epipolar plane angles has been implemented in the 3D kernels developed by van Veenendaal [8], as a future alternative to the present method used in the PRIMAS system.

The following sections outline the methods used for the 2D spatial matching by epipolar lines and the subsequent 3D temporal matching which is based on the previous 2D extrapolation method extended to 3D. Spatial matching of 2D image co-ordinates is achieved between the corresponding images of different cameras. Matched image points are then used to determine the 3D co-ordinates of markers

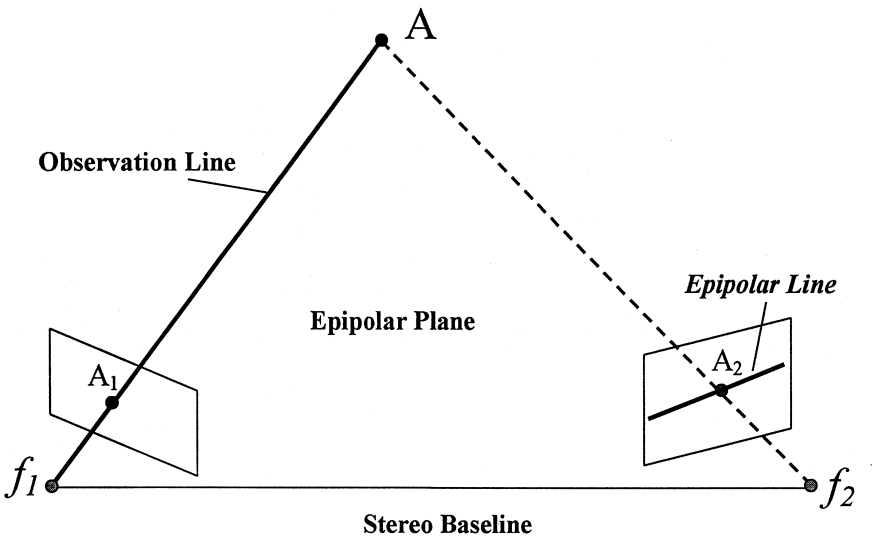


Fig. 14. Epipolar line.

at each instant. Finally, these 3D marker co-ordinates in successive fields are associated to each other via extrapolation, thus forming the marker trajectories.

4.1. Spatial matching using epipolar lines

An epipolar line on the image plane of camera-2 is the projection of an observation line of camera-1, which is the line in 3D space joining a target

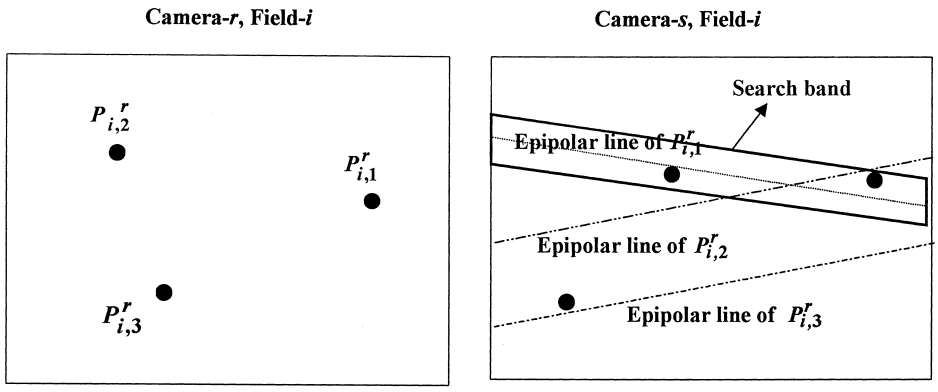


Fig. 15. Matching of two camera images with epipolar lines.

to its image through the focal point of camera-1 (Fig. 14). It follows from this definition that if the target is seen by the second camera as well, its image has to be on the epipolar plane formed by the two images and the physical point. Given two calibrated cameras and the image co-ordinates of a target point in one camera, the equation of the corresponding epipolar line can be written on the image plane of the second camera [9]. This equation takes the following form in terms of DLT parameters used in our system for the camera calibration [4]:

$$(u_{A1}m_1 + v_{A1}m_2 + m_3)u_2 + (u_{A1}m_4 + v_{A1}m_5 + m_6)v_2 + u_{A1}m_7 + v_{A1}m_8 + m_9 = 0 \quad (8)$$

where u_{A1} and v_{A1} are the image co-ordinates of the target point A on camera-1, m_i are the constants defined in terms of DLT parameters a_{ij} of Eq. (7) for both cameras, and u_2 and v_2 are the coordinate axes of the image plane of camera-2.

Corresponding to an image-point $P^r_{i,1}$ in field-i of camera-r, all potentially matching points in the synchronous field of camera-s are determined as the points in the vicinity of the corresponding epipolar line (Fig. 15). Depending upon the placement of markers, camera configuration and involvement of digitisation errors, some cameras may have more than one candidates. This is illustrated in Fig. 15 where there are two candidates corresponding to marker image $P^r_{i,1}$. There may also be cases where some cameras have no candidate image points. A severe case would be when only one camera is showing an image, in which case there would be no candidate image point on any other camera. Consequently, there exists C_c ($\leq N_c$: no. of cameras) "candidate-cameras", which have candidate image points for a particular image point in one camera.

A threshold corresponding to the search band in Fig. 15 is used to select the candidates. From each camera ($s \neq r$), candidates that lie near to the specific epipolar line are selected and they are sorted in ascending order using their distances from the epipolar line. Maximum of five candidate image points are considered, and if more than five image points are found to lie near an epipolar line, only the closest five are considered as match-candidates. This limit was decided based on several experiments which showed that more than three candidates rarely occur; therefore, the probability of leaving out a true match is quite low.

The selected cameras (C_c) are also ordered such that, the camera image for which distance from the epipolar line to a candidate image point is minimum compared to all other camera images, is given the first priority. The one with next closest image point to the epipolar line is given the second priority. In this way, all C_c's are ordered.

Central strategy for the search of matched image points is illustrated in Fig. 16, where a base camera c_1, an image point on it ($P^1_{i,1}$ in Fig. 16), and the corresponding first and second priority cameras, c_r and c_s, are considered. Epipolar lines corresponding to $P^1_{i,1}$ are constructed in c_r and c_s. Then the best

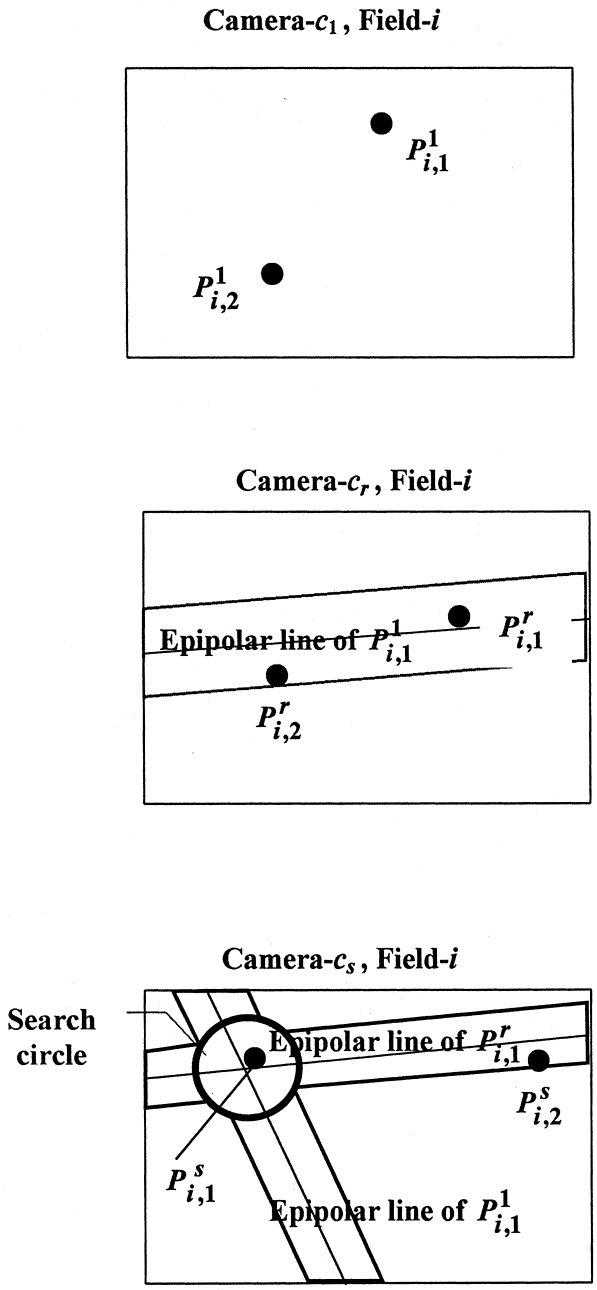


Fig. 16. Central strategy for matching of an image point for a camera triplet.

candidate (evaluated by distance from epipolar line) is determined in c_r (which is $P^r_{i,1}$ in Fig. 16), and its corresponding epipolar line in c_s is constructed. The image plane of c_s has already an epipolar line from c_1, which produces an intersection. If one of the candidate image points in c_s (which is $P^s_{i,1}$ in Fig. 16) is within the search radius around the intersection point, these three image points in three cameras form a potential match. For each potential match detected, epipolar lines corresponding to $P^r_{i,1}$ in c_1 and corresponding $P^s_{i,1}$ in c_1 and c_r are constructed, forming intersections and search radii in c_r and c_1 as well. If the match is verified by these intersections, then the three image points are grouped as the images of a marker. Otherwise, the process is repeated for the same camera triplet considering next best candidate image points in c_r and then in c_s, again for the same image point in c_1. After all five candidate image points are tried for both candidate cameras without a match, the second priority camera is replaced by the third and the process is repeated.

Once a marker is identified in terms of its image co-ordinates in three cameras, a search is conducted in the remaining cameras again by using the intersection of epipolar lines for that particular marker. In this way, a marker may be associated to several images in several cameras. Image points associated to a marker are excluded from further search.

The whole process is repeated for other image points in the base camera. When all the image points in the base camera are handled similarly, another camera is chosen as the base camera and the same is repeated with the remaining unmatched image points. Experiments on this algorithm showed that most of the image points are matched for the first camera triplet, and therefore, the search effort is much less for subsequent camera triplets.

The approach is different if the second priority camera is not available ($C_c < 2$). In this case, for each candidate image point in c_r, all the potential marker matching are created. This introduces some ghost markers that actually do not exist but are produced as a result of grouping each candidate in c_r and the base image point from c_1. These ghost markers do not cause much trouble while trajectory generation, because 3D co-ordinates of these ghost markers are generally way out of real marker trajectories and sometimes even out of the calibration volume. Moreover, a ghost marker disappears after a few fields, and therefore, is easily detectable. We have also provided an option in the software, to generate all possible dual combinations of potentially matching image points when a match cannot be verified by the selected camera triplet. This option, however, creates very large number of ghost markers causing problems in 3D trajectory generation. Normally, there is no need to use this option unless some awkward test conditions arise, in which case user interference is generally required for manual elimination of ghosts.

4.2. Marker co-ordinates in 3D

Once the correspondences between the camera images have been established at

every field, combined image co-ordinates of the same marker are used to reconstruct its 3D co-ordinates. A marker's 3D co-ordinates can be reconstructed if its image co-ordinates are available in at least two calibrated cameras. Fig. 17 illustrates a marker with world co-ordinates (X, Y, Z) and its image co-ordinates (u_r, v_r) and (u_s, v_s) in camera-r and camera-s, respectively. Knowing the DLT calibration parameters, a_{ij}^r and a_{ij}^s, of both cameras, Eq. (7) can be written twice, which provide four equations in terms of three unknowns (X, Y, Z) as before. Each equation in the set represents one of the four planes (P_1, P_2, P_3, P_4) shown in Fig. 17, and the target marker redundantly satisfies all the four plane equations. However, due to the errors involved, any three of the four equations will give a different solution. Therefore, the redundancy can be utilised by minimising the error in the least squares sense. Degree of redundancy increases when the marker is seen by more than two cameras, hence providing more reliable solution. For a marker seen by n-number of cameras, Eq. (7) written n-times will then give the following set of linear equations:

$$\hat{B}_{2n\times 3}\hat{X}_{3\times 1} = \hat{D}_{2n\times 1} \tag{9}$$

Where; $\hat{X} = [X,\ Y,\ Z]^{\mathrm{T}}$, and the elements of $\hat{B}$ and $\hat{D}$ consist of image co-ordinates and calibration parameters. Solution of Eq. (9) by the pseudo-inverse corresponds to the least squares solution:

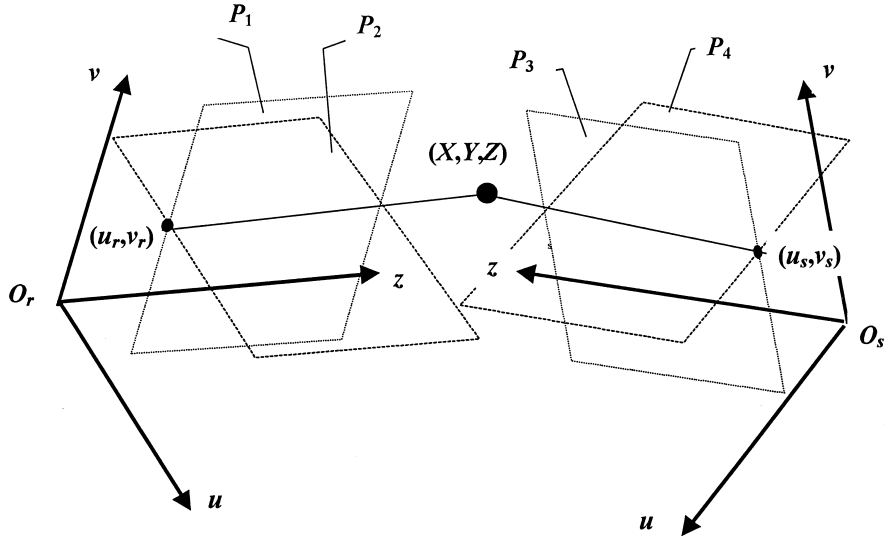


Fig. 17. Reconstruction of 3D co-ordinates of a marker from two camera images.

$$\hat{X} = \left(\hat{B}^{\mathrm{T}}\hat{B}\right)^{-1}\hat{B}^{T}\hat{D} \tag{10}$$

4.3. Trajectory generation: 3D temporal matching

This final step involves generation of marker trajectories by tracking the 3D markers in successive fields. The method used for this purpose is similar to the 2D tracking method previously described for the old two-camera system. When a track is formed up to field-i, its extension to field-$i+1$ is predicted by an extrapolation algorithm. Among the markers of field-$i+1$ within the search sphere around the prediction, the one closest to the prediction is assigned to the same track. In this way, tracks are extended in successive fields. If there exists no marker within the search sphere, the track is ended at field-i and those markers in field-$i+1$ that are not assigned to any track start new tracks. The track pieces generated in this way are handled at a later stage for possible blending.

The process begins from the first field, and all the 3D markers in this field are assumed to start an individual track. Available marker co-ordinates in the second field are then compared with those of the first field, and the tracks are extended by the nearest neighbours within a distance limit. For a track having two markers, a two-point linear extrapolation is used to locate the prediction at the next field. When a track has three or four markers, last three markers are used in a three-point linear least square extrapolation to locate the prediction. As the tracks build up with at least five markers, the next field location is predicted by a least square parabolic fit to the last five marker co-ordinates and then extrapolated on the space parabola [4].

The above process is applied continually, extending tracks if suitable marker match is found and creating new tracks with unassigned markers, until all the fields are handled. The tracks generated are then checked for possible blending. The preliminary blending is carried out automatically, and tracks ending and starting within at most two fields are considered as connectable, and depending upon the number of markers on each, one of the above extrapolation methods is used to verify the continuity. If it is ratified, the two tracks are connected to form a single track. Finally, a cleaning-up is performed to remove tracks that are short, i.e. containing markers less than a specified limit for being a valid track. Most of the ghost tracks are short, and therefore, eliminated at this stage.

A further track editing is necessary interactively by the user. At this stage, the user observes the tracks from different views, deletes remaining invalid tracks, may delete some markers and, if necessary, replaces them with interpolated co-ordinates in order to avoid unrealistic ripples, etc. Also, the user labels the tracks by choosing appropriate names from a list of physical markers used during the experiment. In doing so, the user may give the same label to two tracks that are not blended automatically, in which case the program prompts to verify that they are connectable and blending can also be realised at this stage. Finally, all the unlabelled tracks are deleted, the labelled tracks are trimmed at both ends so that

all have equal lengths and saved as marker trajectories. Most of the features in track editing program have been developed for problematic situations, and in a typical gait session the amount of user intervention is generally limited to labelling, trimming and saving.

5. Performance tests

5.1. Reconstruction of 3D co-ordinates

A relevant test for the 3D reconstruction performance of the system under dynamic conditions is to record the motion of a stick with two markers attached, while it is being moved in varying orientations covering the whole calibrated volume. The purpose is to compare the distance between the reconstructed markers with the true distance, so that a measure of system's 3D reconstruction accuracy and precision can be obtained. The stick used in this test had two markers 502.5 mm apart, and was moved for 4 s within the control volume [10]. Based on 200 samples taken, the mean absolute error of the stick length was 1.1 mm, with a standard deviation of 1.5 mm. The system resolution calculated from the same test results was 0.3 mm, based on Student's t-test at the 99% confidence level, 0.001 mm round-off error and ± 0.001 mm length measurement error.

Commonly used measures of accuracy and precision of marker based motion measurement systems are defined as:

$$Accuracy = \text{Error/Filed of View}$$

$$Precision = \text{Standard Deviation/Field of View}$$

When the field of view is taken as the largest dimension of the control volume, a 3D reconstruction accuracy of around 1 part in 2000, and a precision of 1 part in 1600 were calculated. These performance measures are deemed satisfactory for gait analysis measurements.

Throwing a marker from one side of the calibration volume to the other side and recording its projectile motion is yet another accepted test for accuracy validation. This test was performed in our laboratory, and 3D co-ordinates of the marker were determined at every field with 1/50 s intervals [10]. Parabolic time functions were then fit to X, Y and Z co-ordinates, from which acceleration components were calculated through double differentiation. The average of three tests gave vertical acceleration component of 9.724 m/s^2, and horizontal acceleration components of 0.109 and -0.055 m/s^2 for the projectile motion. Compared with the 9.806 m/s^2 gravitational acceleration in Ankara and considering the other factors involved (e.g. curve fitting and air resistance), less

than 1% error is yet another validation of 3D reconstruction accuracy of the system.

5.2. Trajectory generation

Ability of the motion tracking software to generate marker trajectories was tested under the actual conditions of gait analysis. So far, several tests have been performed, and marker trajectories could be obtained with very little user intervention. Here we report on the results of a typical gait session on a normal subject. In this test, 15 markers were placed on both legs and the pelvis, and upon trimming, the length of the tracks were 80 fields. The software detected 1329 markers, while the total number of physical markers were $15 \times 80 = 1200$. Of these 129 ghost markers, 33 were outside the control volume, and therefore, 96 extraneous markers got involved with the trajectory generation. The number of constructed tracks was 21, and 5 of these were identified as ghost tracks due to their short lengths and deleted automatically by the software. Most of the blending was done again automatically by the software, and therefore, the user had to derive 15 trajectories out of the 16 available. One of the foot markers came in two pieces and upon appropriate labelling and blending by the user interactively, the 15 valid trajectories shown in Fig. 18 were obtained.

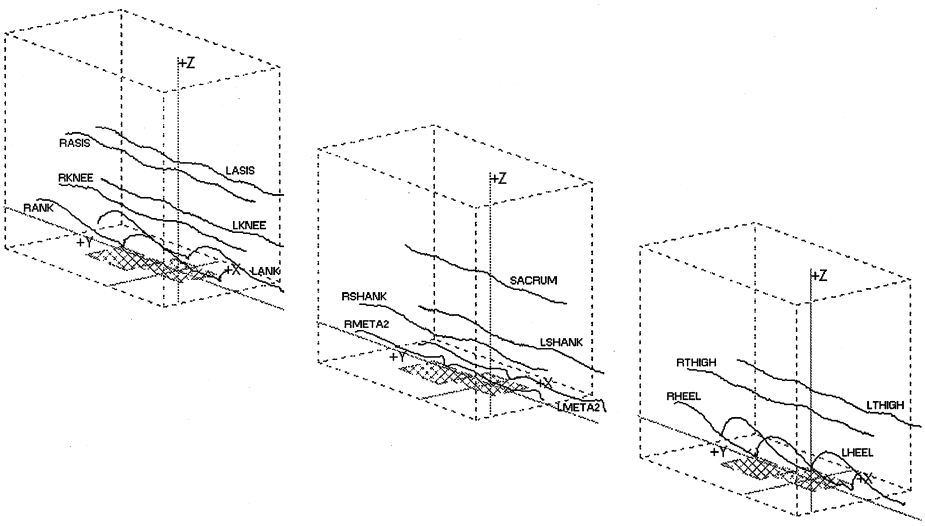


Fig. 18. Trajectories for physical markers generated during a gait session. Dotted lines represent the boundaries of the control volume.

6. Discussion and conclusion

A marker detection and trajectory generation algorithm has been developed for a multicamera based motion analysis system, typically used in gait analysis. The developed software detects markers from a cluster of white-pixels at each field of each camera, and markers are represented by their centroidal co-ordinates. Using the concept of epipolar lines, these marker dots are matched between cameras, independently at each field. Then, 3D co-ordinates of markers are reconstructed from multiple camera images. Finally, tracking of 3D markers at successive fields is accomplished by an extrapolation based prediction and matching method.

Several tests have been made to verify the reliability of the developed software in conjunction with the hardware of the locally developed gait analysis system, *Kiss*. The results revealed that the software performs the described tasks successfully for the purpose of gait analysis. The 3D reconstruction accuracy and precision of the system were found to be comparable with the performance of existing systems.

Performance is highly dependent on the proper choice of several control parameters used in the developed algorithms. Based on extensive experiments, a set of parameter values has been determined for the present laboratory set-up arranged specifically for gait analysis. These are used as default values in the program, but the user specified parameters option is also available for different test conditions. Besides the choice of control parameters, the number and configuration of cameras have also been found to be very effective on system performance. At present, a systematic analysis of camera configuration has not yet been done, but the tests showed that at least four cameras must be used to get acceptable marker trajectories during gait. Also, correction for lens distortion (camera linearisation) is essential for proper functioning of the developed algorithms.

On certain occasions, the program had some difficulty in tracking closely spaced markers, such as those on the feet of very young children, in which case the user interaction is needed for track editing. Also tracking of the crossing markers creates problems in the present software. During gait, crossing 3D trajectories are not normally encountered, but happens at 2D image planes. This is one of the reasons why the first approach based on 2D temporal matching followed by 3D spatial matching did not work for the multicamera system. Although it is not a problem for gait analysis, yet dealing with crossing 3D trajectories may be needed in other applications of marker based motion analysis. This can be achieved by incorporating prior knowledge about markers' motion, such as their expected velocity and accelerations, in the prediction method used for 3D marker tracking. Pattern recognition techniques may also find application in the process, if the motion of 3D markers can be shown to satisfy a certain structure. For example, the markers could be forced to satisfy a moving human body contour in gait analysis application. This, however, would restrict the use of software to a given application.

The main objective of this work was to develop marker trajectory generation software specifically for gait analysis system developed at the METU, and this aim was achieved. Several factors affect the performance of a gait analysis system. These include: (i) performance of the instrumentation and software with which 3D marker co-ordinates are reconstructed and trajectories are generated; (ii) errors in smoothing and differentiation of the trajectories for marker velocities and accelerations; (iii) errors due to soft tissue movements during gait; (vi) errors in determination of anatomical landmarks (e.g. joint centres) from marker positions; (v) errors due to the assumptions involved in biomechanical models. The effect of the last factor is inherent in the standard gait analysis methodology, and therefore, is inevitable unless a completely new approach to gait analysis emerges. In this paper, *Kiss* system has been evaluated based on the first factor, yielding satisfactory performance measures. However, the system has not yet been fully evaluated based on the remaining factors, and therefore, no confidence can be claimed on the quantitative gait data obtained.

In addition to gait analysis, *Kiss* system and its associated software has so far been used in two other applications. The first was again a biomechanical study involving wheelchair propulsion. The second application was on the measurement of movement of lips, in order to develop a pattern recognition algorithm for lip reading the numbers 0–9 in Turkish. Other applications of a similar multicamera marker based motion analysis system have been reported by Sabel [5], where the motion of beer bottles on a conveyor belt and the rotor blade movements in wind turbines were measured. Motion capture of limited number of markers is also finding application in 3D computer animation for the purpose of making movie scenes, commercials and computer games. At present, *Kiss* system has a limited recording time, which prohibits its use for computer animation studies.

Acknowledgements

The authors would like to acknowledge the financial support provided by the following organisations for the development of the gait analysis system described in this paper: Turkish Government State Planning Organisation (DPT) for the development of the current six-camera system; Turkish Scientific and Technical Research Council (TÜBİTAK) and The Middle East Technical University Research Fund (METU-AF) for the development of the former two-camera system.

References

[1] Capozzo A. Gait analysis methodology. Human Movement Science 1984;3:27–50.
[2] Harris GF, Jacqueline PE, Wertsch J. Procedures for gait analysis. Archives of Physical Medicine and Rehabilitation 1994;75:216–25.

[3] Taylor KD, Mottier FM, Simmons DW, Cohen W, Pavlac R, Cornell DP, Hankins GB. An automated motion measurement system for clinical gait analysis. Journal of Biomechanics 1982;15(7):505–16.
[4] Shafiq MS. Motion tracking in gait analysis. M.Sc. Thesis, Faculty of Engineering, The Middle East Technical University, Turkey, Sep. 1998.
[5] Sabel JC. Calibration and 3D reconstruction for multicamera marker based motion measurement. Ph.D. thesis, Faculty of Applied Physics, Delft University of Technology, The Netherlands, Feb. 1999.
[6] Abdel-Aziz YI, Karara HM. Direct linear transformation from comparator co-ordinates into object-space co-ordinates. In: Proceedings ASP Symposium on Close-Range Photogrammetry. Falls Church: American Society of Photogrammetry, 1971. p. 1–18.
[7] Baltsavias EP, Stallmann D. Trinocular vision for automatic and robust three-dimensional determination of the trajectories of moving objects. Photogrammetric Engineering and Remote Sensing 1991;57(8):1079–86.
[8] van Veenendaal HLJ. Recommendations for the next generation PRIMAS system. M.Sc. Thesis, Faculty of Applied Physics, Delft University of Technology, The Netherlands, May 1992.
[9] Schalkoff RJ. Digital image processing and computer vision. Singapore: Wiley, 1989.
[10] Karpat Y. Development and testing of kinematic data acquisition tools for a gait analysis system. M.Sc. Thesis, The Middle East Technical University, Turkey, Jan. 2000.