Computers in Industry 70 (2015) 183–193

Contents lists available at ScienceDirect

Computers in Industry

journal homepage: www.elsevier.com/locate/compind

Dynamic 3D surface reconstruction and motion modeling from a pan–tilt–zoom camera

Salam Dhou, Yuichi Motai *

Department of Electrical and Computer Engineering, Virginia Commonwealth University, Richmond, VA 23284, USA

ARTICLE INFO

Article history:
Received 14 March 2014
Received in revised form 10 February 2015
Accepted 17 February 2015
Available online 20 March 2015

Keywords:
Pan–tilt–zoom camera
Video stream reconstruction
Structure-from-motion
Target tracking

ABSTRACT

3D surface reconstruction and motion modeling has been integrated in several industrial applications. Using a pan–tilt–zoom (PTZ) camera, we present an efficient method called dynamic 3D reconstruction (D3DR) for recovering the 3D motion and structure of a freely moving target. The proposed method estimates the PTZ measurements to keep the target in the center of the field of view (FoV) of the camera with the same size. Feature extraction and tracking approach are used in the imaging framework to estimate the target's translation, position, and distance. A selection strategy is used to select keyframes that show significant changes in target movement and directly update the recovered 3D information. The proposed D3DR method is designed to work in a real-time environment, not requiring all frames captured to be used to update the recovered 3D motion and structure of the target. Using fewer frames minimizes the time and space complexity required. Experimental results conducted on real-time video streams using different targets to prove the efficiency of the proposed method. The proposed D3DR has been compared to existing offline and online 3D reconstruction methods, showing that it uses less execution time than the offline method and uses an average of 49.6% of the total number of frames captured.

1. Introduction

3D surface reconstruction has been integrated in camera-mounted computers for industry applications in recent years. Thousands of companies manufacture surveillance cameras. However, the expected software that can facilitate the analysis of the huge flow of video information is virtually absent. A useful idea would be to use 3D information from one off-the-shelf camera to extract the target motion and structure from a sequence of 2D images. Dynamic 3D reconstruction of structure and motion of targets is an important topic in industry [1–3]. It emerged in many recent industrial applications including real-time 3D image visualization [4], face detection [5,6], and 3D surface and curve reconstruction [1,7].

The advantage comes when the 3D reconstruction method uses less time and space to reconstruct the 3D objects from fewer frames, hopefully without incorporating a second camera [8]. 3D reconstruction using a pan–tilt–zoom (PTZ) camera is very promising due to its active imaging function, such as changing focus, pan–tilt head moving in network camera applications [9,10]. The work presented here specifically features a PTZ configuration so that the 3D object modeling of the target can be extracted from one single actively moving camera. In dynamic environments, a big need arises to combine the 3D reconstruction with tracking in which the 3D structure and motion are sequentially updated considering tracking information [11–13]. An active PTZ system can be useful for modeling because it uses the estimated PTZ motion to keep the target in the center of the image and updating the 3D model of the target. We propose a new method, D3DR, to recover the 3D motion and structure of the target, incrementally considering information from a single PTZ camera tracking the target.

The contribution of this paper is twofold: (1) proposing a dynamic 3D reconstruction method (D3DR) that dynamically estimates PTZ measurements of a camera tracking a target, and (2) proposing a keyframe selection strategy to select the frames that have a significant amount of motion to be used in the reconstruction process. Those new approaches simply update the 3D structure and motion of the target incrementally with fewer frames. We have specialized the imaging device as a latest PTZ

* Corresponding author. Tel.: +1 804 828 1281.
E-mail addresses: dhous@vcu.edu (S. Dhou), ymotai@vcu.edu (Y. Motai).

http://dx.doi.org/10.1016/j.compind.2015.02.005
0166-3615/

camera, and compared our proposed D3DR to those existing salient methods (called offline and online methods using the orthographic projection) in the identical pan–tilt–zoom setting.

Our overall proposed flow works as follows: the user selects an area inside a target by dragging the mouse to form a rectangular area. Then, the camera tracks the target area to keep it in the FoV of the camera with the same size. The keyframe selection strategy is applied to select the frames that show significant motion of the target to be used in the reconstruction process. The algorithm updates the target's motion and structure after every keyframe selected. Thus, the proposed method uses the tracking information immediately without the need to store growing matrices in memory.

The remainder of this paper is organized as follows: Section 2 presents a related study about 3D reconstruction. Section 3 reviews the existing offline and online reconstruction methods. Section 4 presents the proposed D3DR. The proposed method's performance is tested and the result analysis is shown in Section 5. Section 6 concludes the paper.

2. 3D reconstruction background

We present a brief reference background of 3D reconstruction methods that are relevant to our research. Section 2.1 reviews the problem of target tracking. Section 2.2 reviews 3D motion and structure recovery.

2.1. Target tracking

Feature points extraction and tracking is one of the most important parts of the reconstruction process. Optical flow can be used as a method for feature tracking. Yilmaz et al. [14] reviewed and classified the state-of-the-art tracking methods into different categories based on the object and motion representations used. Feature-based methods are developed to work in long image sequences for structure-from-motion problems [15,16] or for dynamic environments when both the camera and the target are moving [17]. A tracker should be able to detect and track the good feature points [18]. Kanade–Lucas–Tomasi (KLT) tracker [19] assumes the affine projection model and uses the least sum of squared difference (SSD) of pixel intensity between frames in the feature neighborhood to estimate new feature location. We choose the KLT tracker for our implementation because it has been demonstrated to be robust and reliable. Active PTZ network cameras are used for tracking a target as in [20–22].

2.2. 3D motion and structure recovery

3D reconstruction methods take advantage of information provided by a long sequence of images [23–25]. In [26], the method constructs 3D shapes of geometrical entities based on the iterative closest point (ICP) algorithm. Bozdagi et al. [27] proposed a method that estimates 3D motion and adapts a generic wire-frame to a particular speaker simultaneously within an optical flow based framework. Feature-based structure-from-motion methods including [28–30] depend on feature points selection and tracking in an image stream. The recovery and triangulation of 3D trajectories has been studied in [31,32], respectively. Online 3D reconstruction has advantages over the offline 3D reconstruction methods when used in real systems [11–13]. Automatic reconstruction methods exist to recover motion and structure from monocular image sequences [33–35] or using feature-based structure-from-motion approaches [28,36]. Extended Kalman filter is used to recover motion and structure from an uncalibrated video [37]. Distributed and scalable volumetric architecture for reconstructing arbitrary structures in real time are used as in [38]. This architecture consists of acquisition nodes to reconstruct partial models from multiple views and a master node to merge those partial models.

3. Existing 3D reconstruction studies

We present existing 3D reconstruction methods directly relevant to our research. Section 3.1 shows the offline reconstruction using the factorization method. Section 3.2 shows the online reconstruction from a moving camera using the sequential factorization method.

3.1. Offline 3D reconstruction from a stationary camera using the factorization method

For the recovery of 3D structure and motion, we present Tomasi and Kanade's factorization method [29]. The input of this method is the feature points correspondences provided by the KLT tracker. The KLT tracker used in this paper rejects the outliers and chose the "good" features to track. We assume a sequence of F images taken for a moving target in front of a PTZ camera. The orthographic projection model assumes that projection rays are parallel to the camera's optical axis.

Fig. 1 shows the systems of reference for the image, target, and camera. For frame f, the camera orientation is described by the unit vectors i_f, j_f, and k_f. The distance between the camera origin and the fixed world coordinate is represented by the translation vector T. Each point $s_p = [X, Y, Z]^T$ in the fixed world coordinate is located by KLT at the image position $p_{f,p} = (u_{f,p}, v_{f,p})$ in image frame f and located at feature point position $p_{f-1,p} = (u_{f-1,p}, v_{f-1,p})$ in frame $f-1$.

Given a video stream, suppose that we have tracked P feature points over F frames. We then have the image coordinates: $p_{f,p} = \{(u_{f,p}, v_{f,p}) : f = 1, \ldots, F, p = 1, \ldots, P\}$. The corresponding feature points $p_{f,p} = (u_{f,p}, v_{f,p})$ are written in a measurement matrix $W : 2F \times P$:

$$W = \begin{bmatrix} u_{11} & \cdots & u_{1P} \\ \vdots & \ddots & \vdots \\ u_{F1} & \cdots & u_{FP} \\ v_{11} & \cdots & v_{1P} \\ \vdots & \ddots & \vdots \\ v_{F1} & \cdots & v_{FP} \end{bmatrix} = \begin{bmatrix} U \\ V \end{bmatrix}. \quad (1)$$

The registered measurement matrix W^* is computed by subtracting from W the mean $x_f = \sum_{p=1}^{P} u_{fp}/P$ and $y_f = \sum_{p=1}^{P} v_{fp}/P$ of all $u_{f,p}$ and $v_{f,p}$ at a frame f respectively:

$$W^* = \begin{bmatrix} U \\ V \end{bmatrix} - \begin{bmatrix} x_f \\ y_f \end{bmatrix} = MS \quad (2)$$

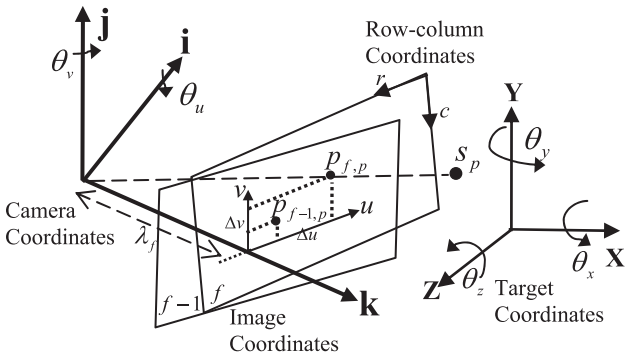


Fig. 1. The coordinate system showing the geometric relation between frame $f-1$ and frame f.

where $M = [i_1 \ \cdots \ i_F \ \ j_1 \ \cdots \ j_F]^T$ represents the $2F \times 3$ motion matrix and $S = [s_1 \ \ldots \ s_P]$ represents the $3 \times P$ shape matrix. The rows of M represent the orientations of the horizontal and vertical motion throughout the stream. The columns of S are the coordinates of the P feature points with respect to their centroid $c = 1/P \sum_{p=1}^{P} s_p$. Singular value decomposition (SVD) is applied on the registered matrix W^* to recover structure and motion matrices.

Although the offline method recovers the 3D structure and motion from video streams, it is difficult to be applied in real-time applications. It requires all the input images are given before the method runs. Thus the space complexity needed for measurement matrix will be huge and SVD operation will be expensive. So, the online reconstruction method is more appropriate in real-time applications.

3.2. Online 3D reconstruction from a PTZ camera using the sequential factorization method

The sequential factorization method [28] is applied to recover the 3D structure and motion sequentially of the moving target in front of a moving PTZ camera that tracks the target motion. The sequential factorization method consists of the following two steps:

- Sequential structure space computation
- Sequential metric transformation

The measurement matrix W_f grows for each frame $f = 1, 2, \ldots, F$ in the following manner:

$$W_1 = [u_{1p} \ \ v_{1p}]^T, \quad W_2 = [u_{1p} \ \ u_{2p} \ \ v_{1p} \ \ v_{2p}]^T, \quad W_f = [u_{1p} \ \cdots \ u_{Fp} \ \ v_{1p} \ \cdots \ v_{fp}]^T. \tag{3}$$

Let $Z_f = W_f^T W_f$, since $W_f = U_f D_f V_f^T$, then $Z_f = (U_f D_f V_f^T)^T U_f D_f V_f^T = V_f D_f^2 V_f^T$. The eigenvectors of Z_f are equivalent to the right singular vectors V_f of W_f. So, the structure space is obtained by computing the eigenvectors of Z_f. Given a matrix Q_0 with orthonormal columns, a sequence of matrices Q_f are generated for every frame $f = 1, 2, \ldots, F$, by adding the feature correspondences to the matrix Z_f: $Z_f = Z_{f-1} + u_f u_f^T + v_f v_f^T$ and taking the QR factorization of Y_f as in

$$Q_f R = Y_f, \tag{4}$$

where $Y_f = Z_f Q_{f-1}$. Y_f and R_f are intermediate variables used to calculate Q_f.

Because Q_f converges to the matrix V_f, let $H_f = Q_f Q_f^T$ be the projection matrix onto range Q_f. Given a matrix $\bar{Q}_0$ with orthonormal columns, a sequence of matrix $\bar{Q}_f : P \times 3$ providing the stationary basis for the shape space can be generated for every frame f. Two intermediate matrices H_f and Y_f are computed as $H_f = Q_f Q_f^T$ and $Y_f = H_f \bar{Q}_{f-1}$. Then the QR factorization of Y_f is taken as in

$$\bar{Q}_f R_f = Y_f. \tag{5}$$

The camera coordinates i_f and j_f are computed sequentially for every frame f as in $\hat{i}_f^T = u_f^T \bar{Q}_f$ and $\hat{j}_f^T = v_f^T \bar{Q}_f$. The affine transformation A_f is recovered by

$$A_f = D_f^{-1} E_f \tag{6}$$

where $E_f = E_{f-1} + g(\hat{i}_f, \hat{i}_f) + g(\hat{j}_f, \hat{j}_f)$, $D_f = D_{f-1} + g(\hat{i}_f, \hat{i}_f) g^T(\hat{i}_f, \hat{i}_f) + g(\hat{j}_f, \hat{j}_f) g^T(\hat{j}_f, \hat{j}_f) + g(\hat{i}_f, \hat{j}_f) g^T(\hat{i}_f, \hat{j}_f)$, and $g^T(i_f, j_f) = [i_{f1} \cdot j_{f1} \ \ i_{f1} \cdot j_{f2} + i_{f2} \cdot j_{f1} \ \ i_{f1} \cdot j_{f3} + i_{f3} \cdot j_{f1} \ \ i_{f2} \cdot j_{f2} \ \ i_{f2} \cdot j_{f3} + i_{f3} \cdot j_{f2} \ \ i_{f3} \cdot j_{f3}]$.

The camera coordinates are obtained as in $i_f^T = \hat{i}_f^T A_f$ and $j_f^T = \hat{j}_f^T A_f$. Thus, motion matrix M_0^f of the target is formed by i_f^T and j_f^T as discussed in Section 3.1:

$$M_0^f = [\ i_1^T \ \cdots \ i_F^T \ \ j_1^T \ \cdots \ j_F^T]^T \quad \text{and} \quad S_0^f = A_f^{-1} \bar{Q}_f. \tag{7}$$

Eq. (7) provides the sequentially updated motion matrix M_0^f and the structure matrix S_0^f using the online reconstruction method. Thus, using this method, the structure and motion are sequentially updated at each frame. The online method can be performed in real time as it significantly reduces the computational complexity.

4. Proposed dynamic 3D reconstruction (D3DR) method

We propose in this section the D3DR method for the recovery of the 3D motion and structure. This method consists of three parts. Section 4.1 discusses the frame selection strategy based on motion detection. Section 4.2 discusses estimating the initial Motion matrix based on affine projection matrix. Section 4.3 shows the updating method of the 3D motion matrix considering the estimation of PTZ. Section 4.4 discusses updating the 3D structure method based on the 3D motion matrix update.

4.1. Frame selection strategy based on motion detection

We aim to select the frames that have a significant amount of motion compared to the previous frames so that we can reduce the computation time [39–41]. To achieve this purpose, we use the projection matrix and the camera eigenvectors to determine the amount of the total motion detected.

From the QR factorization method applied in (5) in Section 3.2, we use the Projection matrix $H_f = Q_f Q_f^T$ as projection rays. The projection rays can be reconstructed from the 2D points in the projection matrix H_f. We then find the transformation that ensures the least distance between projection lines with 3D estimated Plucker lines [42]. Plucker lines are the implicit representation of 3D lines. A Plucker line $L = (l, e)$ is described by a unit vector l and a moment e. Using this line representation, we propose the following metric to conveniently determine the distance of a 3D point s_p to the line L:

$$d(s_p, L) = \|(S_0^f A_f) \times l - e\| \tag{8}$$

where $\times$ is the cross product and A_f is recovered sequentially in (6), Section 3.2.

The transformed 3D points must be on the projection rays reconstructed from their corresponding 2D points. Particularly, we look for a transformation τ_f applied to all 3D points s_p in (8) such that the total distance over all correspondences is minimized in the least squares sense. Instead of using the structure matrix S_0^f in (8) we use the basis of the structure matrix $\bar{Q}_f$ to determine the distance by substitution (7) in Section 3.2 in (8) we got: $\arg\_\min_T = \|\bar{Q}_f \times l - e\|$.

So, we determine the transformation τ_f using the basis of the structure matrix $\bar{Q}_f$ without the complete recovery of the 3D structure matrix S_0^f. We now examine the transformation matrix recovered τ_f to determine the amount of motion occurred in frame f. Eq. (2) in Section 3.1 shows the transformation process from 2D feature points to 3D structure using the motion matrix M and the translation vector T: $W = MS + T$ which can be written in the corresponding homogeneous matrix form: $W = [M \ \ T]S$. The transformation matrix recovered τ_f is compared to $[M \ \ T]_f$ to determine if selecting the current frame will add information to

the motion matrix. We recover the basis vectors of each transformation matrix and find the angle between the corresponding basis vectors.

Suppose that $\{a_1, b_1, c_1\}$ are the basis vectors of $[M \quad T]_f$ and $\{a_2, b_2, c_2\}$ are the basis vectors of τ_f. We calculate the angle between the corresponding basis vectors as follows:

$$\Theta_a = \cos^{-1}\left(\frac{a_1 \cdot a_2}{\|a_1\|\|a_2\|}\right), \quad \Theta_b = \cos^{-1}\left(\frac{b_1 \cdot b_2}{\|b_1\|\|b_2\|}\right), \quad \Theta_c = \cos^{-1}\left(\frac{c_1 \cdot c_2}{\|c_1\|\|c_2\|}\right)$$

Fig. 2 shows the angles between the corresponding basis vectors. Those angles show the difference between the subspaces. We select the frame for computation if the sum of the angles is considerably large. Thus, the selected keyframes are used in the next subsections for the reconstruction process.

4.2. Estimating the initial motion matrix based on affine projection matrix

We use the affine camera model. This model assumes that the object frame is located in the centroid of the target being observed. We estimate the projection matrix and compute the camera parameters in the following two steps.

Step 1: We show the relation between the 3D coordinates (X, Y, Z) and 2D coordinates $(u_{f,p}, v_{f,p})$ through the following affine camera model equation [43]:

$$[u' \quad v' \quad q']^T = \beta \cdot [X \quad Y \quad Z \quad G]^T \tag{9}$$

with $u_{f,p} = \frac{u'}{q'} = \frac{m_{11}X+m_{12}Y+m_{13}Z+m_{14}G}{m_{31}X+m_{32}Y+m_{33}Z+m_{34}G}$ and $v_{f,p} = \frac{v'}{q'} = \frac{m_{21}X+m_{22}Y+m_{23}Z+m_{24}G}{m_{31}X+m_{32}Y+m_{33}Z+m_{34}G}$ and the affine projection matrix is:

$$\beta = \begin{bmatrix} s_x \cdot \lambda \cdot r_1 & s_x \cdot \lambda \cdot t_x + c_0 \cdot t_z \\ s_y \cdot \lambda \cdot r_2 & s_y \cdot \lambda \cdot t_y + r_0 \cdot t_z \\ 0^{1\times 3} & t_z \end{bmatrix} \tag{10}$$

where s_x and s_y are scale factors (pixels/mm), r_1 and r_2 are the first two rows of the rotation matrix. $T : [t_x \quad t_y \quad t_z]$ is the 3D translation vector, c_0 and r_0 are the coordinates of the principle point in pixels relative to row-column frame and λ is the focal length. Under simplifying conditions we assume s_x and s_y equal 1. The rotation matrix and focal length are discussed in IV C and IV D.

Step 2: We determine the 8 degrees of freedom (independent entries) in the projection matrix through a homogeneous linear system formed by:

$$\Lambda m = 0 \tag{11}$$

where Λ has rank 8 and $m = [m_{11}, m_{12}, \ldots, m_{34}]^T$.

We need at least six world image point matches to solve the homogeneous linear system. The vector m can be found using the SVD of $\Lambda : \Lambda = UDV$. The solution m is the column of V corresponding to the smallest singular value of Λ.

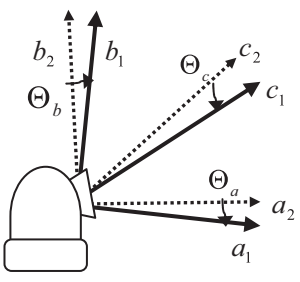


Fig. 2. The coordinate system showing the geometric relation between basis vectors $\{a_1, b_1, c_1\}$ and $\{a_2, b_2, c_2\}$.

Algorithm 1 shows the flow of estimation to update the initial motion matrix using the affine projection matrix. This subsection serves as the base of the updating method of the motion matrix as will be done in the following Section 4.3.

Algorithm 1. Step 1: We show the affine camera model equation which links the 3D coordinates (X, Y, Z) and 2D coordinates $(u_{f,p}, v_{f,p})$: $[u' \quad v' \quad q']^T = \beta \cdot [X \quad Y \quad Z \quad G]^T$ as in (9).

Step 2: We determine the 8 degrees of freedom in the projection matrix through a homogeneous linear system formed by: $\Lambda m = 0$, as in (11). This linear system is solved for m using SVD operation. The solution m is the column of V corresponding to the smallest singular value of Λ.

4.3. Updating motion matrix considering pan–tilt–zoom estimation

At each frame selected as in Section 4.1, the proposed method updates the motion matrix considering the estimated values of PTZ. Here, we use optical flow to measure the pan–tilt angles of the camera, and SIFT to measure the focal length. In Section 4.3.1, we present the motion update considering pan and tilt estimation while in Section 4.3.2, we show the motion update considering zoom estimation.

4.3.1. Motion matrix update considering pan–tilt estimation

To estimate the 3D motion, we assume the object is moving in front of the PTZ camera and the camera tracks the target's movement. At each image frame captured, we estimate the pan–tilt rotational angle and track the target to keep it in the center of the image as done by the following four steps.

Step 1: The pan–tilt rotational angles of the camera are estimated using optical flow. Fig. 1 shows the camera coordinates, the image coordinates, and the target coordinates. The angle vector θ is estimated for each frame in the sequence in order to keep the target centroid in the center of the camera frame. The angle vector θ consists of the horizontal movement and the vertical movement: $\theta = [\theta_u \quad \theta_v]^T$, where the rotational angle (θ_u) of the camera is defined by pan which is associated with the horizontal displacement Δu. Similarly, the tilt (θ_v) rotational angle of the camera is defined by tilt and associated with the vertical displacement Δv. We calculate the degree of these values as $\theta_u = \tan^{-1}(\Delta u/\lambda)$ and $\theta_v = \tan^{-1}(\Delta v/\lambda)$, where (λ) is the focal length of the camera. θ_u and θ_v are calculated using the optical flow induced by the target motion.

To estimate the pan–tilt motion, we assume also that some 3D point $s_p = (x, y, z)$ is projected onto location $P_{f-1,p}(u_{f-1,p}, v_{f-1,p})$ in the image plane of frame $f-1$. Suppose that $s_p = (x, y, z)$ is moved to a new location relative to the camera by rotating the object reference about the X-axis by angle θ_X and about the Y-axis by angle θ_Y. Under affine projection, the point $s_p = (x, y, z)$ in 3D space is projected onto a new location $P_{f,p}(u_{f,p}, v_{f,p})$ in the image plane of frame f. The rotational motion transformation of the 3D point can be viewed as moving an image point $P_{f-1,p}(u_{f-1,p}, v_{f-1,p})$ in frame $f-1$ to a corresponding image point $P_{f,p}(u_{f,p}, v_{f,p})$ in frame f based on a 2D image rotational mapping. Fig. 1 shows the projection of the 2D point $P_{f-1,p}(u_{f-1,p}, v_{f-1,p})$ in frame $f-1$ to the corresponding image point $P_{f,p}(u_{f,p}, v_{f,p})$ in frame f.

Step 2: We estimate the camera motion vector as shown in Fig. 1. Under the affine model, the projection of the new point $P_{f,p}(u_{f,p}, v_{f,p})$ in frame f makes a displacement in the image plane

denoted by Δu and Δv which causes a pan angle θ_v and/or tilt-directional angle θ_u, respectively. The camera is able to rotate only in pan angle θ_v and tilt angle θ_u direction with respect to the image plane u–v. So the motion vector of the camera is: $\theta_{ca} = [\theta_v \quad \theta_u \quad 0]^T$.

Step 3: The rotation matrix $R_f(\theta_z, \theta_y, \theta_x)$ for the arbitrary motion in the 3D space is determined. Suppose $(\theta_X, \theta_Y, \theta_Z)$ is a vector of the 3D rotation about the X, Y, and Z-axis of the target, respectively. A normal 3D rotation about an arbitrary axis of the target coordinates through the origin can be described by a successive rotation θ_X about its X-axis, followed by a rotation of θ_Y about its Y-axis, followed by a rotation of θ_Z about its Z-axis, respectively. The rotation matrix $R_f(\theta_z, \theta_y, \theta_x)$ for the arbitrary motion in the 3D space is recovered and given by:

$$R_f(\theta_Z,\theta_Y,\theta_X) = R(\theta_Z)\cdot R_f(\theta_Y)\cdot R_f(\theta_X) = \begin{bmatrix} \cos\theta_Z & -\sin\theta_Z & 0 \\ \sin\theta_Z & \cos\theta_Z & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} \cos\theta_Y & 0 & \sin\theta_Y \\ 0 & 1 & 0 \\ -\sin\theta_Y & 0 & \cos\theta_Y \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_x & -\sin\theta_x \\ 0 & \sin\theta_x & \cos\theta_x \end{bmatrix}$$
$$= \begin{bmatrix} \cos\theta_Z\cos\theta_Y & \cos\theta_Z\sin\theta_Y\sin\theta_x - \sin\theta_Z\cos\theta_x & \cos\theta_Z\sin\theta_Y\cos\theta_x + \sin\theta_Z\sin\theta_x \\ \sin\theta_Z\cos\theta_Y & \sin\theta_Z\sin\theta_Y\sin\theta_x + \cos\theta_Z\cos\theta_x & \sin\theta_Z\sin\theta_Y\cos\theta_x - \cos\theta_Z\sin\theta_x \\ -\sin\theta_Y & \cos\theta_Y\sin\theta_x & \cos\theta_Y\cos\theta_x \end{bmatrix}. \tag{12}$$

Hence, the rotational operations do not commute.

Step 4: The motion matrix can be estimated based on the equation: $W = MS + T$. We use the equations of the online reconstruction method to determine the rotation matrix.

Algorithm 2 shows the flow of estimation to update the motion matrix considering pan–tilt estimation of the camera. This subsection provides the updating method of the motion matrix M_{PT}^f given the pan and tilt operations which are estimated dynamically at each frame.

Algorithm 2. Step 1: Estimate pan $\theta_v = \tan^{-1}(\Delta v/\lambda)$ and tilt $\theta_u = \tan^{-1}(\Delta u/\lambda)$ using optical.

Step 2: The camera motion vector is estimated as the following: $\theta_{ca} = [\theta_v \quad \theta_u \quad 0]^T$.

Step 3: The rotation matrix $R_f(\theta_z, \theta_y, \theta_x)$ for the arbitrary motion in the 3D space is recovered as in (12).

Step 4: The motion matrix can be estimated based on the equation: $W = MS + T$. We use the equations of the online reconstruction method (4) and (5) to determine the rotation matrix.

4.3.2. Motion matrix update considering zoom estimation

We implemented an automatic zoom control to keep the target the same size in the FoV of the camera. We discuss in the following three steps how we control the focal length of the camera λ_f at each fth frame.

Step 1: Recover the depth of the feature points in each of the consequent frames. To do this, we applied the scale invariant feature transform (SIFT) algorithm [44] to extract features from images. Those keypoints are highly distinctive and invariant to image scaling, rotation, change in illumination, and 3D camera viewpoint. Let the scale of a keypoint p in frame $f-1$ be $\sigma_{f-1,p}$.

Let Γ_{f-1} be a vector of the scales of keypoints in frame $f-1$ in the sequence: $\Gamma_{f-1} = [\sigma_{f-1,1} \quad \sigma_{f-1,2} \quad \cdots \quad \sigma_{f-1,P}]$, when capturing new frame f, the scales of the keypoints of frame f are saved in $\Gamma_f = [\sigma_{f,1} \quad \sigma_{f,2} \quad \cdots \quad \sigma_{f,P}]$.

The keypoints in frame $f-1$ are matched to their correspondences in frame f using their descriptors. For the matched keypoints, we compare the scales of the keypoints in both frames by taking the average of the ratio of scales for the matched keypoints between frame $f-1$ and frame f:

$$\bar{\Gamma} = \frac{\sum_{p=1}^{P} \sigma_{f,p}/\sigma_{f-1,p}}{P}, \tag{13}$$

where $\bar{\Gamma}$ is the average of the ratio of scales between frame $f-1$ and frame f and P is the total number of keypoints in the both frames.

Step 2: Adjust the focal length according to the ratio of scales recovered in Step 1. When $\bar{\Gamma} > 1$, this means that the target is moving closer to the camera. Similarly, $\bar{\Gamma} < 1$ means that the target is moving further away from the camera. In both cases, the focal length is changed accordingly to keep the object look with the same size as follows:

$$\lambda_f = \bar{\Gamma}\cdot\lambda_{f-1}, \tag{14}$$

where λ_f is the focal length of the camera in frame f. Thus, (14) finds the focal length of the camera at frame f based on the keypoints scales and the focal length of the camera at frame $f-1$.

Fig. 1 shows the projection at two different frames (at frame $f-1$ and frame f). From Fig. 1, d_{f-1} is the depth of the center of object mass in frame $f-1$. The target moved toward the camera center at frame f, which results in $d_f < d_{f-1}$ and a change in the position of the point s_p relative to the camera frame. Thus, the point s_p is projected to p in a new location in frame f making the target size projected at frame f look larger. The zoom function in the camera is implemented so that the ratio between the depth d_{f-1} of the middle mass of the object and focal length λ_{f-1} in frame $f-1$: λ_{f-1}/d_{f-1} should be constant through all frames. Let this constant ratio C be a constant image size of the target:

$$C = \frac{\lambda_{f-1}}{d_{f-1}}. \tag{15}$$

Thus, we recover the value of the focal length λ_f at frame f by multiplying the average scales ratio of the keypoint $\bar{\Gamma}$ by the focal length value of the previous frame $f-1$: λ_{f-1}.

Step 3: The projection matrix is updated based considering new value of the focal length estimated. By substituting (14) into (10), we get the projection matrix at frame f: β_f

$$\beta_f = \begin{bmatrix} s_x\cdot\bar{\Gamma}\lambda_{f-1}\cdot r_1 & s_x\cdot\bar{\Gamma}\lambda_{f-1}\cdot t_x + c_0\cdot t_z \\ s_y\cdot\bar{\Gamma}\lambda_{f-1}\cdot r_2 & s_y\cdot\bar{\Gamma}\lambda_{f-1}\cdot t_y + r_0\cdot t_z \\ 0 & t_z \end{bmatrix} \tag{16}$$

The first and second rows r_1 and r_2 of the rotation matrix are estimated. So the motion matrix M_Z^f at frame f can be updated.

Algorithm 3 shows the flow of estimation to update the motion matrix based on zoom estimation of the camera at every frame f. This subsection provides the updating method of the motion

matrix M_Z^f given the zoom operation which is done dynamically at each frame.

Algorithm 3. Step 1: Recover the depth of the feature points in every consequent frames $\bar{\Gamma}$ by comparing the vector of scales $\Gamma_{f-1} = [\sigma_{f-1,1} \quad \sigma_{f-1,2} \quad \cdots \quad \sigma_{f-1,P}]$ in frame $f-1$ and the vector of scales $\Gamma_f = [\sigma_{f,1} \quad \sigma_{f,2} \quad \cdots \quad \sigma_{f,P}]$ in frame f using the formula: $\bar{\Gamma} = \left(\sum_{p=1}^{P} \sigma_{f,p}/\sigma_{f-1,p}\right)/P$, as in (13).

Step 2: Adjust focal length λ_f at frame f according to the ratio of scales $\bar{\Gamma}$ recovered in Step1, using: $\lambda_f = \bar{\Gamma} \cdot \lambda_{f-1}$. Keep a constant ratio C as a constant size of the target: $C = \lambda_{f-1}/d_{f-1}$ as in (14) and (15).

Step 3: The projection matrix is updated considering the new value of the focal length estimated λ_f at frame f. This is done as in (16) to get the projection matrix at frame f: β_f.

4.4. Updating structure matrix considering pan–tilt and zoom estimation

We aim to update the 3D structure S_{PTZ}^f of the target at every frame given the feature points coordinates in frame f and the recovered motion M_{PTZ}^f based on the pan, tilt, and zoom operations of the camera. We assume the following relation holds:

$$W_f = M_{PTZ}^f S_{PTZ}^f + T_f \tag{17}$$

where W_f is the measurement matrix discussed in (3) in Section 3.2 and T_f is the translation vector. The 3D structure of the object can be computed at each frame given the new camera coordinates and the horizontal and vertical translation. The projection from the 3D point s_p to the 2D point $P_{f,p}(u_{f,p}, v_{f,p})$ in frame f is given by in Fig. 1. Using (17) the structure of the target can be estimated using the least squares solution.

$$S_{PTZ}^f = M_{PTZ}^{f\ +}(W_f - T_f) \tag{18}$$

where $(.)^+$ denote the matrix pseudo inverse and S_Z^f is the updated structure matrix using zoom operation.

Given that the registered matrix in (2) in Section 3.1 is the measurement matrix after subtracting the mean of all elements in the same row, (18) can be written in the following form: $W_f^* = W_f - T_f$, where W_f^* is the registered matrix. So, the 3D structure of the current frame can be written in the following form: $S_{PTZ}^f = M_{PTZ}^{f\ +} W_f^*$.

This section provides the updating method of structure matrix S_{PTZ}^f, given the motion matrix M_{PTZ}^f and coordinates of the points in the image plane.

5. Experimental results

In this section, we discuss the experimental results of the proposed method. Section 5.1 presents the datasets specifications. Section 5.2 shows the experimental results for the offline method (described in Section 3.1). Section 5.3 shows the experimental results for the online method (described in Section 3.2). Section 5.4 shows the experimental results for the proposed D3DR method (described in Section 4). The quantitative comparison between the three methods is presented in Section 5.5. A comparative evaluation to other feature extraction and tracking methods is conducted in Section 5.6.

5.1. Datasets conditions

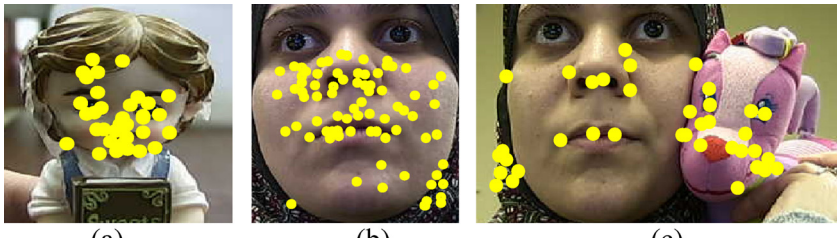
Offline, online, and D3DR methods were tested using real image datasets of a toy face, human face, a combination of a toy and human face [49]. The same datasets were used to test the three methods. All video streams contain motion and translation in 3D space and were taken with a frame frequency 33.33 Hz. The size of the frames was fixed (320 × 240) for all video streams. We used Visual Studio with OpenCV library to implement the code. We used the built in KLT tracker in the OpenCV library. We also use the SIFT implementation by Hess [45].

In Fig. 3, the extracted feature points from the first frame of each dataset are shown. The number of feature points varies using different datasets. It is affected by the size of the region selected and the texture in this selected region. The toy face (a) was a comparison of the real human face (b) in Fig. 3, represented by a rigid object (a), a deformable object (b), and a combination of the two (c). The face target was widely used for common engineering applications. Thus, this study demonstrated the feasibility and applicability of D3DR.

Table 1 shows the video streams properties in terms of the number of frames and the number of points extracted for each video stream. As noticed in Table 1, the three video streams consist of frames in the range 200–300. The experimental settings include a Canon VB-C60 network pan–tilt–zoom camera interfaced to a PC running Microsoft Visual Studio 2003 to create all the datasets in Table 1.

5.2. Offline reconstruction method

We present the experimental results for the offline reconstruction method. Fig. 4 shows the 3D structures of the objects in the three video streams. The upper image in (a)–(c) shows the recovered 3D structure of the object after mesh gridding the recovered 3D points of the object. Mesh grid creates wireframe parametric surfaces from 3D points. The bottom image in (a)–(c) corresponds to the 3D model of the object after mapping the texture and color information.



(a) (b) (c)

Fig. 3. The extracted output of the KLT feature tracker for the datasets: (a) toy face, (b) human face and (c) toy and human face.

Table 1
Video streams properties.

Video stream	Frames number	Feature points number
Toy face	236	48
Human face	232	108
Toy and human face	260	47

Table 2
Average error of 3D structure and motion using offline 3D reconstruction method.

Video stream	Structure RMS (19) (mm)	Yaw RMS (20) (°)	Roll RMS (20) (°)	Pitch RMS (20) (°)
Toy face	3.254	0.9546	4.9023	6.4686
Human face	4.012	0.7371	4.9339	3.5994
Toy and human face	4.901	4.6665	6.4554	5.4391

Fig. 5 shows the recovered 3D motion of the three objects using offline method compared to the measured motion recovered by Polhemus Liberty motion estimator [46]. As shown in Fig. 5, the recovered 3D motion is synchronized with the measured motion. The difference between the computed angle value and the measured value increases while more frames are captured and more motion is accomplished. You can also notice that the error is high in comparison to the error estimated in the original papers [29]. This is due to several reasons: the free and uncontrolled motion of the target applied in this paper, the camera movement to track the target, and the delay associated with the camera movement and the Polhemus tracker error.

Table 2 shows the quantitative errors in motion and structure compared to the measured value. To evaluate the structure's quality quantitatively, we measured some distances on the actual objects with a ruler and compared them to the distances computed from the recovered 3D points using the average root mean square (RMS):

$$\mathrm{RMS}(S) = \sqrt{\frac{\sum (s_{p,comp} - s_{p,meas})^2}{P}} \tag{19}$$

where $s_{p,comp}$ corresponds to the computed 3D points and $s_{p,meas}$ indicates the measured 3D points.

We compared also the error in the recovered 3D motion using offline reconstruction method and the measured motion using Polhemus Liberty motion estimator. The same RMS measurement criteria were taken to compare between the computed and the measured motion. Eq. (20) calculates the average RMS measurements for the difference in 3D motion angles.

$$\mathrm{RMS}(\Theta) = \sqrt{\frac{\sum (\Theta_{comp} - \Theta_{meas})^2}{P}} \tag{20}$$

where Θ_{comp} corresponds to the 3D angle computed using the offline method and Θ_{meas} corresponds to the measured angle.

As shown in Table 2, the recovered average motion error in all the cases is less than 5.5° except the roll angle of the toy and human face object, which is considered the hardest reconstructed object. The structure error was always less than 5 mm for all datasets.

We applied the offline method identical to the original paper [29] to evaluate the error measurements. This error is due to

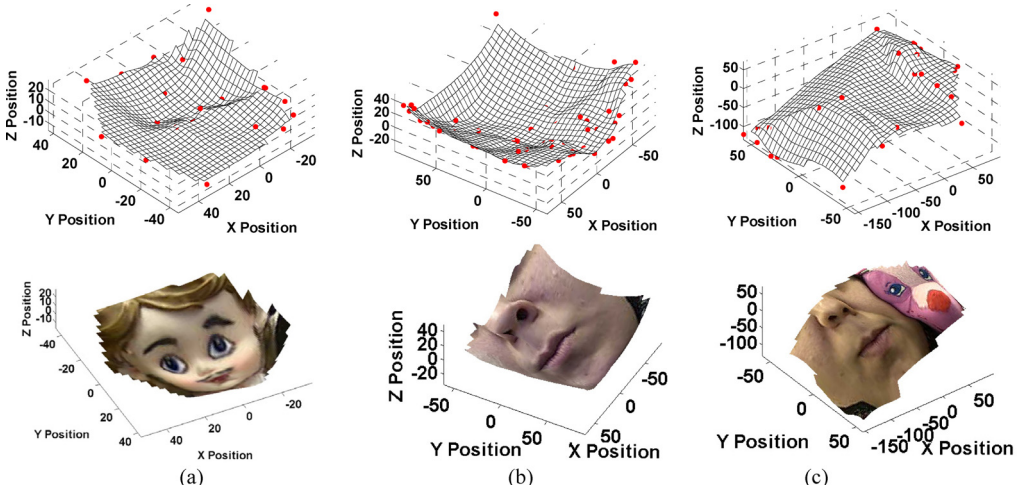


Fig. 4. 3D surface structure and 3D structure models with texture mapping using the offline reconstruction method for (a) toy face, (b) human face and (c) toy and human face.

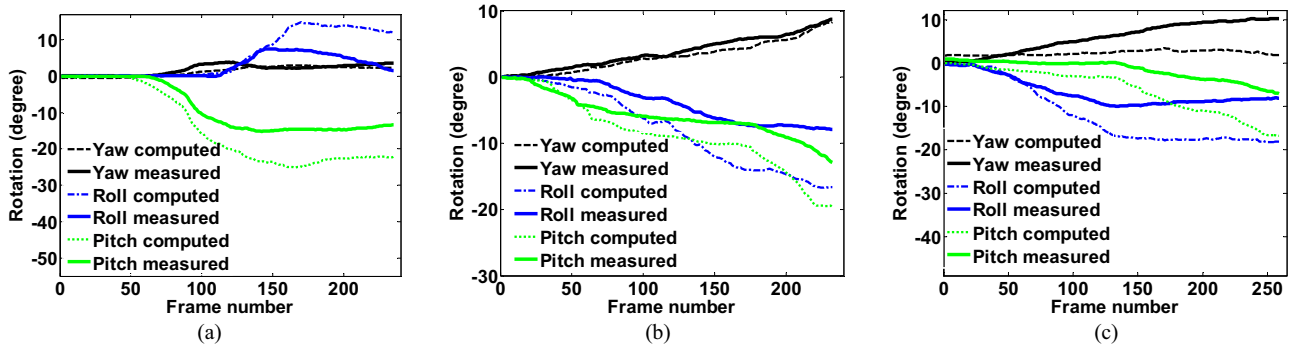


Fig. 5. Offline reconstruction method for 3D motion in terms of yaw, roll, and pitch using (a) toy face, (b) human face and (c) toy and human face.

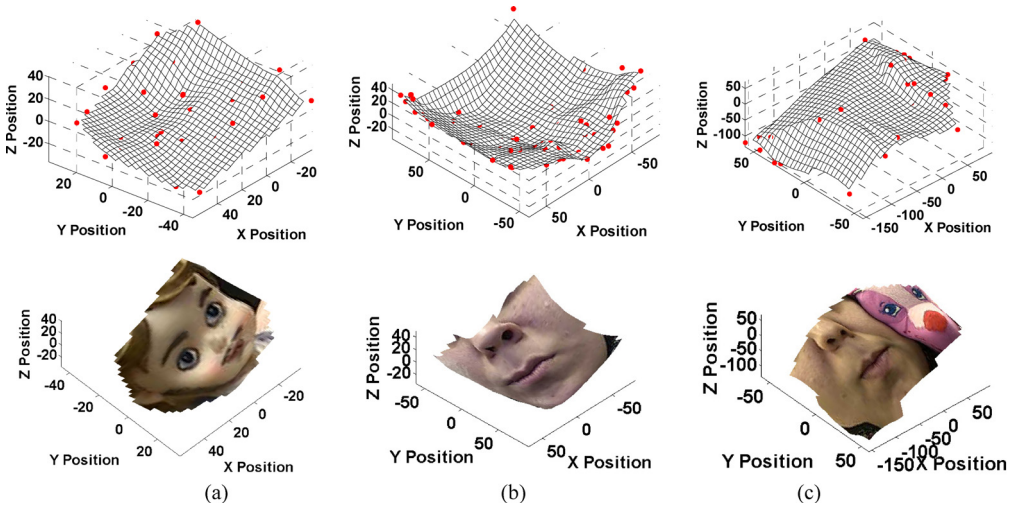


Fig. 6. 3D surface structure and 3D structure models with texture mapping using the online reconstruction method for (a) toy face, (b) human face and (c) toy and human face.

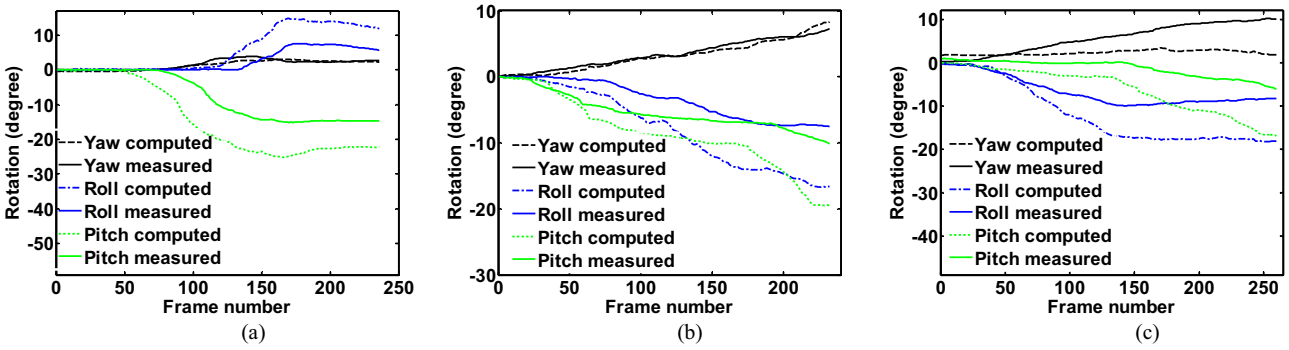


Fig. 7. Online reconstruction method for 3D motion in terms of yaw, roll, and pitch using (a) toy face, (b) human face and (c) toy and human face.

several reasons: the free and uncontrolled motion of the target applied in this paper which may include the off-set from the center of FoV, the camera movement to track the target, and the Polhemus tracker error.

5.3. Online reconstruction method

We present the results of the online method when the target is moving in front of the PTZ camera. The PTZ camera is moving to track the target in order to keep it in its FoV. Fig. 6 shows 3D structures of the objects in the three video streams. The upper image in (a)–(c) corresponds to the recovered 3D model of the object generated by mesh gridding the recovered 3D points of the object. The bottom image in (a)–(c) corresponds to the 3D model of the object after mapping the texture and color information.

Fig. 7 shows the recovered 3D motion of the three objects using the online method compared to the ground truth motion measured using Polhemus Liberty motion estimator. The recovered 3D motion is close to the measured 3D motion except for the yaw angle in some of the datasets. The pitch and roll angles are closer to the measured motion.

Table 3 shows the quantitative errors in motion and structure using the online method compared to the ground truth. We compared the error in the recovered 3D motion and structure between the online and the ground truth using the average root mean square (ARMS) as in (19) and (20).

As shown in Table 3, the recovered average motion error in all the cases does not exceed 6.5° except for the pitch angel of the toy object. This was due to the hard motion that occurred at that direction. The structure error was always less than 5.5 mm for all datasets.

5.4. D3DR method using pan–tilt and zoom estimation

We present the results of the proposed D3DR method in which the camera is tracking the target using PTZ operations to keep it in its FoV. We assume the pan, tilt, and zoom information estimated by the tracking camera to be used in the structure and motion reconstruction.

Fig. 8 shows the 3D motion recovered for all video streams using D3DR method. It shows also 3D structures of the objects used with texture mapping. The 3D structures of the objects are incrementally updated after each frame capture. Moreover, the keyframe selection strategy is applied here. The red line at the bottom of each graph shows the frames selected to be used for the reconstruction procedure.

As shown in Fig. 8, the D3DR method selects keyframes to be used in the reconstruction process. In (a), the number of frames selected was 123 out of 260. In (b), it was 119 out of 232. In (c), it was 121 out of 236. The average of the number of frames selected is

Table 3
Average error of 3D structure and motion using online 3D reconstruction method.

Video stream	Structure RMS (19) (mm)	Yaw RMS (20) (°)	Roll RMS (20) (°)	Pitch RMS (20) (°)
Toy face	4.0135	0.6150	4.4894	7.6209
Human face	4.8920	0.5288	5.3000	4.3084
Toy and human face	5.3456	4.4459	6.4513	5.9056

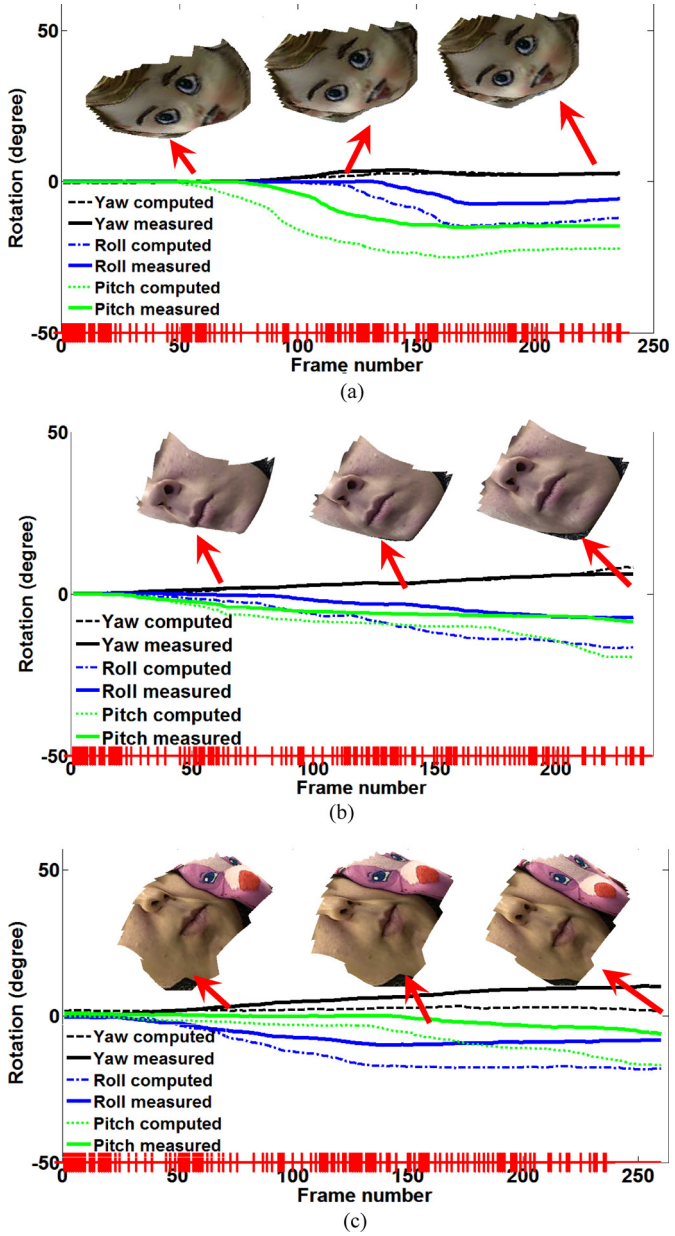


Fig. 8. D3DR measurement from PTZ estimation for 3D motion and structure per frame for (a) toy face, (b) human face and (c) toy and human face.

49.6% of the total number of frames. Regarding the recovered 3D structure, you can see that the accuracy of the 3D structure increases over time.

Table 4 shows the quantitative errors of the recovered structure and motion using the D3DR method. The structure error was around 5 mm for all datasets and the motion error was always an average of 5° for all angles.

Here we compare Table 4 to Tables 2 and 3. Comparing the motion error of D3DR method to the offline method, we found that

Table 4
Average error of 3D structure and motion using dynamic 3D reconstruction method.

Video stream	Structure RMS (19) (mm)	Yaw RMS (20) (°)	Roll RMS (20) (°)	Pitch RMS (20) (°)
Toy face	4.2050	0.5861	4.2541	7.4210
Human face	4.8492	0.5103	5.3278	4.2137
Toy and human face	5.2472	4.6120	6.7629	6.2301

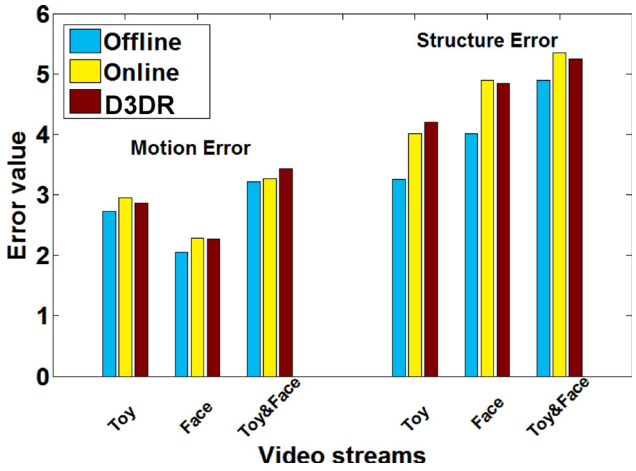


Fig. 9. Motion and structure error comparison between the three methods (offline, online, and D3DR) using the three video streams (toy, face, and toy and face).

they are around the same value. We compared the structure error of the D3DR method with the structure error of the offline and the online methods. The structure error using the D3DR is around the structure error of the online method. However, the structure error of the offline method is always less than both the online and D3DR methods.

5.5. Quantitative evaluation of offline, online, and d3dr methods

We compare the three reconstruction methods: offline, online, and D3DR in terms of motion and structure error as shown in Fig. 9. The motion error is evaluated as a combination of yaw, roll and pitch errors as in the following equation:

$$M_{err} = \frac{\sqrt{\mathrm{RMS}(\theta_x)^2 + \mathrm{RMS}(\theta_y)^2 + \mathrm{RMS}(\theta_z)^2}}{3} \quad (21)$$

where $\mathrm{RMS}(\theta_x)$, $\mathrm{RMS}(\theta_y)$ and $\mathrm{RMS}(\theta_z)$ corresponds to yaw, pitch and roll error, respectively calculated in (20).

As you can see the motion and structure error shown in Fig. 9, the D3DR method has less or almost the same error value as the online method. The offline method uses a different approach that uses batch processing which produces more accurate results than sequential methods such as online and D3DR methods.

Average execution time in seconds and number of frames are compared in Table 5. We used a Dell PC machine with an Intel Core 2 Duo CPU 2.33 GHz and 2 GB RAM. The three methods were tested with the same three video streams described in Section 4.1. Table 5 shows the quantitative comparison of the performance of the three reconstruction methods in terms of time and space complexity.

As shown in Table 5, the D3DR and online methods use less time than the offline. D3DR uses an average of 49.6% of the total number of frames captured to reconstruct the 3D motion and structure, while the offline and online methods use 100% of frames captured. The average execution time per frame of the D3DR method is 8% less than the execution time of the offline method. Therefore, D3DR is more efficient than the offline method in terms of time, and more efficient than both the offline and online methods in terms of space. The proposed method is designed to work in real time and

Table 5
Quantitative comparison of the three methods.

Metric	Offline	Online	D3DR
Time average (s)	0.421	0.0117	0.0372
Frames used (%)	100	100	49.6

Table 6
Average error of pan, tilt, and zoom using OF/SIFT, OF/SURF and histogram.

Video stream	Pan (°)	Tilt (°)	Zoom (mm)
OF-SIFT	0.425	0.232	2.323
OF-SURF	0.442	0.211	2.112
H-PMHT	0.891	0.573	N/A

minimize the number of frames used to minimize space. As a total, D3DR was the optimal method for the reconstruction with both small error and time, and improved on the weakness of both offline and online.

5.6. Comparative evaluation to other feature extraction and tracking methods

We further compare the feature extraction and tracking module of the D3DR method to other alternative approaches: SURF [47], and H-PMHT. [48]. SURF is a scale- and rotation-invariant detector by building on the strengths of the leading existing detectors and descriptors using a Hessian matrix-based measure for the detector, and a distribution-based descriptor. H-PMHT is a method applied to derive a stable tracking algorithm that uses the entire image as its input data avoiding peak picking and other data compression steps required to produce traditional point measurements by linking a histogram interpretation of the intensity data with the tracking method of probabilistic multihypothesis tracking (PMHT).

Table 6 shows the quantitative comparison between the three tracking methods: (1) OF-SIFT used in our proposed method, (2) OF-SURF, and (3) H-PMHT. We compare the three tracking methods in terms of the pan, tilt, and zoom errors. As you can see in Table 6, feature-based trackers using OF-SIFT and OF-SURT outperform H-PMHT tracking OF-SURF has almost the same results as OF-SIFT for the pan and tilt operations, because both of the algorithms use optical flow to estimate pan–tilt motion. For the zoom operation, OF-SIFT and OF-SURF use similar approaches, so their zoom errors are also very similar.

6. Conclusion

We presented in this paper the D3DR method using a PTZ camera. The PTZ measurements are estimated by the proposed method to track the target and keep it in the center of the FoV. The 3D structure and motion of the target were iteratively updated under the affine model considering the PTZ measurements. At every frame captured, we select the keyframe to be used in the reconstruction process. For every keyframe selected, the pan and tilt angles are measured and used to update the 3D motion of the target. The focal length of the camera is also computed at every keyframe selected and used also to update the 3D motion of the target. The structure matrix is updated iteratively based on the motion matrix.

The experimental results showed that the proposed D3DR method is in most cases as accurate as the online method. Using the selection strategy, D3DR uses an average of 49.6% of the total number of frames captured to reconstruct the 3D motion and structure while the offline and online methods use 100% of frames captured. Regarding the time complexity, the proposed method takes less time than the offline method. The space and time complexity of the proposed method allows it to be used in real-time applications.

Acknowledgements

This study was supported in part by the School of Engineering at Virginia Commonwealth University (VCU), Higher Education Equipment Trust Fund from the State Council of Higher Education for Virginia, VCU's Presidential Research Incentive Program, and NSF CAREER Award 1054333.

References

[1] J. Molleda, R. Usamentiaga, D.F. García, F.G. Bulnes, A. Espina, B. Dieye, L.N. Smith, An improved 3D imaging system for dimensional quality inspection of rolled products in the metal industry, Computers in Industry 64 (December (9)) (2013) 1186–1200.
[2] S. Kahn, U. Bockholt, A. Kuijper, D.W. Fellner, Towards precise real-time 3D difference detection for industrial applications, Computers in Industry 64 (December (9)) (2013) 1115–1128.
[3] F. Bianconi, L. Ceccarelli, A. Fernández, S.A. Saetta, A sequential machine vision procedure for assessing paper impurities, Computers in Industry 65 (February (2)) (2014) 325–332.
[4] W.M. Chiew, F. Lin, K. Qian, H.S. Seah, A heterogeneous computing system for coupling 3D endomicroscopy with volume rendering in real-time image visualization, Computers in Industry 65 (February (2)) (2014) 367–381.
[5] K. Emrith, L. Broadbent, L.N. Smith, M.L. Smith, J. Molleda, Real-time recovery of moving 3D faces for emerging applications, Computers in Industry 64 (December (9)) (2013) 1390–1398.
[6] R.F.V. Saracchini, J. Stolfi, H.C.G. Leitão, G.A. Atkinson, M.L. Smith, Robust 3D face capture using example-based photometric stereo, Computers in Industry 64 (December (9)) (2013) 1399–1410.
[7] Z. Jian-dong, Z. Li-yan, D. Xiao-yu, D. Zhi-an, 3D curve structure reconstruction from a sparse set of unordered images, Computers in Industry 60 (February (2)) (2009) 126–134.
[8] X. Zhang, W.-M. Tsang, M. Mori, K. Yamazaki, Automatic 3D model reconstruction of cutting tools from a single camera, Computers in Industry 61 (September (7)) (2010) 711–726.
[9] P.D. Varcheie, G.-A. Bilodeau, Human tracking by IP PTZ camera control in the context of video surveillance, Proceedings of the Sixth International Conference Image Analysis and Recognition. Lecture Notes in Computer Science, vol. 5627, Springer-Verlag, Berlin, 2009, pp. 657–667.
[10] A. Gardel, J.L. Lazaro, J.M. Lavest, J.F. Vazquez, Detection and tracking vehicles using a zoom camera over a pan-and-tilt unit, in: Proceedings of the IEEE Intelligent Vehicle Symposium, 2002, June, pp. 215–220.
[11] E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser, P. Sayd, Generic and real-time structure from motion using local bundle adjustment, Image and Vision Computing 27 (2009) 1178–1193.
[12] D. Nistér, Preemptive RANSAC for live structure and motion estimation, Machine Vision and Applications 16 (December (5)) (2005) 321–329.
[13] E. Imre, S. Knorr, A.A. Alatan, T. Sikora, Prioritized sequential 3D reconstruction in video sequences with multiple motions, in: ICIP, 2006, 2969–2972.
[14] A. Yilmaz, O. Javed, M. Shah, Object tracking: a survey, ACM Computing Surveys 38 (December (4)) (2006) 1–45.
[15] J. Park, J. Yoon, C. Kim, Stable 2D feature tracking for long video sequences, International Journal of Signal Processing, Image Processing and Pattern Recognition 1 (December (1)) (2008) 39–46.
[16] T. Cooke, R. Whatmough, Detection and Tracking of Corner Points for Structure from Motion. Technical Report DSTO-TR-1759, AR Number: AR-013-476, Australian Government, Defence Science and Technology Organisation, Australia, 2005, August.
[17] N.P. Papanikolopoulos, P.K. Khosla, Adaptive robotic visual tracking: theory and experiments, IEEE Transactions on Automatic Control 38 (March (3)) (1993) 429–445.
[18] J. Shi, C. Tomasi, Good features to track, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'94), 1994, June, pp. 593–600.
[19] C. Tomasi, T. Kanade, Detection and Tracking of Point Features. Technical Report CMU-CS-91-132, Carnegie Mellon University, Pittsburgh, PA, 1991, April.
[20] Y. Yao, B. Abidi, M. Abidi, 3D target scale estimation and target feature separation for size preserving tracking in PTZ video source, International Journal of Computer Vision 82 (May (3)) (2009) 244–263.
[21] S. Kang, J. Paik, A. Koschan, B. Abidi, M.A. Abidi, Real-time video tracking using PTZ cameras, Proceedings of the Sixth International Conference on Quality Control by Artificial Vision (QCAV'03), vol. 5132, 2003, May, pp. 103–111.
[22] Y. Ye, J. Tsotsos, E. Harley, K. Bennet, Tracking a person with pre-recorded image database and a pan, tilt, and zoom camera, Machine Vision and Applications 12 (July (1)) (2000) 32–43.
[23] M. Lhuillier, L. Quan, A quasi-dense approach to surface reconstruction from uncalibrated images, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (March (3)) (2005) 418–433.
[24] Y. Motai, A.C. Kak, An interactive framework for acquiring vision models of 3D objects from 2D images, IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 34 (1) (2004) 566–578.
[25] E. Imre, S. Knorr, B. Özkalayci, U. Topay, A.A. Alatan, T. Sikora, Towards 3-D scene reconstruction from broadcast video, Signal Processing: Image Communication 22 (2) (2007) 108–126.
[26] P.J. Besl, N.D. McKay, A method for registration of 3-D shapes, IEEE Transactions on Pattern Analysis and Machine Intelligence 14 (February (2)) (1992) 239–256.
[27] C. Bozdagi, A.M. Tekalp, L. Onural, 3-D motion estimation and wireframe model adaptation including photometric effects for model-based coding of facial image sequences, IEEE Transactions on Circuits and Systems for Video Technology 4 (June (3)) (1994) 246–256.

[28] T. Morita, T. Kanade, A sequential factorization method for recovering shape and motion from image streams, IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (August (8)) (1997) 858–867.
[29] C. Tomasi, T. Kanade, Shape and motion from image streams under orthography: a factorization method, International Journal of Computer Vision 9 (November (2)) (1992) 137–154.
[30] C.J. Poelman, T. Kanade, A paraperspective factorization method for shape and motion recovery, IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (March (3)) (1997) 206–218.
[31] A.J. Davison, I.D. Reid, N.D. Molton, O. Stasse, MonoSLAM: real-time single camera SLAM, IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (June (6)) (2007) 1052–1067.
[32] M. Chen, G. AlRegib, B.-H. Juang, Trajectory triangulation: 3D motion reconstruction with L1 optimization, in: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2011), Prague, Czech, 2011, May.
[33] S. Soatto, P. Perona, Reducing structure from motion: a general framework for dynamic vision part 1: modelling, IEEE Transactions on Pattern Analysis and Machine Intelligence 20 (September (9)) (1998) 933–942.
[34] A. Chiuso, P. Favaro, H. Jin, S. Soatto, Structure from motion causally integrated over time, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (April (4)) (2002) 523–535.
[35] S. Soatto, P. Perona, Recursive 3D visual motion estimation using subspace constraints, International Journal of Computer Vision 22 (March–April (3)) (1997) 235–259.
[36] A. Azarbayejani, A. Pentland, Recursive estimation of motion, structure and focal length, IEEE Transactions on Pattern Analysis and Machine Intelligence 17 (June (6)) (1995) 562–575.
[37] D.W. Murray, L.S. Shapiro, Dynamic updating of planar structure and motion: the case of constant motion, Computer Vision and Image Understanding 63 (January (1)) (1996) 169–181.
[38] D. Ruiz D, B. Macq, Exploitation of interframe redundancy for real-time volumetric reconstruction of arbitrary shapes, IEEE Journal of Selected Topics Signal Processing 2 (August (4)) (2008) 556–567.
[39] Y. Seo, S. Kim, K. Doo, J. Choi, Optimal keyframe selection algorithm for three-dimensional reconstruction in uncalibrated multiple images, Optical Engineering 47 (May (5)) (2008).
[40] Y. Hwang, J.K. Seo, H.K. Hong, Key-frame selection and an LMedS-based approach to structure and motion recovery, IEICE Transactions on Information and Systems E91D (January (1)) (2008) 114–123.
[41] J.K. Seo, Y.H. Hwang, H.K. Hong, Structure and motion recovery using two step sampling for 3D match move, 3rd Mexican International Conference on Artificial Intelligence (MICAI 2004): Advances Artificial Intelligence. Lecture Notes in Computer Science, vol. 2972, 2004, pp. 652–661.
[42] L.P. Kuptsov, Plücker coordinates, in: M. Hazewinkel (Ed.), Encyclopaedia of Mathematics, Springer, 2001.
[43] E. Trucco, A. Verri, Introductory Techniques for 3-D Computer Vision, Prentice Hall, Upper Saddle River, NJ, 1998, pp. 34–40, 132–136.
[44] D.G. Lowe, Distinctive image features from scale-invariant keypoints, International Journal of Computer Vision 60 (November (2)) (2004) 91–110.
[45] R. Hess, An open-source SIFT library, in: Proceedings of the International Conference on Multimedia, 2010, October.
[46] Polhemus Liberty Documentation, 2010 http://www.polhemus.com/?page=Motion_Liberty.
[47] H. Bay, T. Tuytelaars, L.V. Gool, SURF: speeded up robust features, Proceedings of the 9th European Conference Computer Vision (ECCV). Lecture Notes in Computer Science, vol. 3951, 2006, May, pp. 404–417.
[48] R.L. Streit, M.L. Graham, M.J. Walsh, Multitarget tracking of distributed targets using histogram-PMHT, Digital Signal Processing 12 (May) (2002) 394–404.
[49] Representative Codes, 2015 http://www.people.vcu.edu/~ymotai/Modeling.zip.

Salam Dhou received her B.S. and M.S. degrees in computer science from Jordan University of Science and Technology (JUST), Irbid, Jordan, in 2004 and 2007, respectively. She worked as an instructor in the Computer Science Department at JUST, in 2007–2008. She received her Ph.D. degree in electrical and computer engineering, Virginia Commonwealth University, Richmond, VA, USA in 2013. She is currently a post-doctoral research fellow at Harvard University, Boston, MA, USA. Her research interests include data mining, machine learning, and medical imaging.

Yuichi Motai received the B.Eng. degree in instrumentation engineering from Keio University, Tokyo, Japan, in 1991, the M.Eng. degree in applied systems science from Kyoto University, Kyoto, Japan, in 1993, and the Ph.D. degree in electrical and computer engineering from Purdue University, West Lafayette, IN, USA, in 2002. He is currently an associate professor of electrical and computer engineering at Virginia Commonwealth University, Richmond, VA, USA. His research interests include the broad area of sensory intelligence; particularly in medical imaging, pattern recognition, computer vision, and sensory-based robotics.