Stereoscopic Video Generation Based on Efficient Layered Structure and Motion Estimation From a Monoscopic Image Sequence

Konstantinos Moustakas, Dimitrios Tzovaras, and Michael G. Strintzis, *Fellow, IEEE*

***Abstract*—This paper presents a novel object-based method for the generation of a stereoscopic image sequence from a monoscopic video, using bidirectional two–dimensional motion estimation for the recovery of rigid motion and structure and a Bayesian framework to handle occlusions. The latter is based on extended Kalman filters and an efficient method for reliably tracking object masks. Experimental results show that the layered object scene representation, combined with the proposed algorithm for reliably tracking object masks throughout the sequence, yields very accurate results.**

***Index Terms*—Bayesian classification, object tracking, recursive filtering, stereoscopic video generation, structure reconstruction.**

I. Introduction

The GENERATION of a stereoscopic image sequence using only monoscopic video is a problem of considerable practical interest due to the large number of monoscopic videos existing in many databases, which when converted to stereoscopic can offer a more realistic sense of the scene to the viewer. The bottleneck in this conversion is in the estimation of the scene structure from the monoscopic sequence, a problem admitting an infinite number of solutions since true lengths in the scene are unknown. The resulting mathematically ill-posed problem, commonly called structure from motion (SfM) in the literature [1], has been under extensive research [2] and analysis [3] for the last decades. Besides the intermediate view generation using a stereoscopic or multiview sequence [4] and the use of SfM for specific problems like the modeling of a face from video data [5], only recently [6], have researchers tried to explicitly extend the SfM problem to stereoscopic video generation from a monoscopic image sequence. In the industry, most software providing companies simply exploit the "Pulfrich" [7] effect to generate a pseudo sense of depth, by using the same video as a second sequence with the difference of a small temporal delay. Others provide "two-dimensional (2-D) to three-dimensional (3-D) conversion," for still images, as a manually performed service.

Different approaches have been exploited in the past to estimate 2-D motion in monoscopic sequences, which may then be used for 3-D motion estimation and scene structure recovery. Very efficient approaches include those based on optical flow estimation, mathematical transformations and feature-point tracking. Although all these methods yield good results each for different types of sequences, the feature-point tracking based method appears to yield generally more robust results, because the motion between consecutive frames is, in the majority of sequences in practice, small enough to allow efficient feature tracking. Extended Kalman filters (EKF) have been successfully used [1], [8], [6] for the estimation of scene structure parameters, utilizing the 2-D motion estimation data. Improving on past results [9] Azarbayejani and Pentland proposed in [1] a very robust method for SfM, which in addition to the standard scene structure parameters was also able to estimate focal length. The recovered depths of the feature points may be used to create dense depth maps via an interpolation method using e.g., 2-D triangulation. Finally, the stereoscopic video sequence is produced, based on the generated depth information.

The present work extends the EKF method for application to objects; in fact EKF yields satisfactory results if the variations of the feature points' motion and depths are relatively small, which is a constraint that is usually met for objects. A novel algorithm for object tracking is also introduced, which utilizes the 2-D bidirectional motion estimation and an efficient prediction-correction procedure to track the objects' masks throughout the sequence. Finally, to handle occlusions, a Bayesian framework is presented, which registers occluded areas to the objects composing the scene.

II. Proposed Method

Initially, rigid objects are identified in the scene using the segmentation method described in [10], which utilizes a k-means approach with connectivity constraints for defragmentation. Reliable features are extracted for each object in the first frame, using texture sharpness criteria as described in [11], and tracked to the final one using an optimized Kanade–Lucas–Tomasi (KLT) feature tracker. Reliable object masks are extracted using the algorithm analyzed in the following Section II-A and motion and structure are subsequently estimated using the layered EKF-based algorithm. Then the probabilistic method to classify the occluded points to objects, described in Section II-B, is applied. Finally, the stereoscopic video is generated reassembling the processed frames. A schematic description of the proposed algorithm is shown in Fig. 1.

A. Reliable Object Tracking

This part of the proposed algorithm is focused upon reliable object tracking across the whole sequence and is based on bidirectional motion estimation in order to better handle occluded and emerging areas. Initially, N_A rigid objects $A_m, m = 1, \ldots, N_A$ are identified in the first and the last frame of each

Manuscript received February 24, 2004; revised November 20, 2004. This paper was recommended by Associate Editor H. Chen.

K. Moustakas and M. G. Strintzis are with the Electrical and Computer Engineering Department, Information Processing Laboratory, Aristotle University of Thessaloniki, Thessaloniki 541 24, Greece, and also with the Informatics and Telematics Institute/CERTH, Thermi-Thessaloniki 57001, Greece (e-mail: moustaka@iti.gr; strintzi@eng.auth.gr).

D. Tzovaras is with Informatics and Telematics Institute/CERTH, Thermi-Thessaloniki 57001, Greece.

Digital Object Identifier 10.1109/TCSVT.2005.852401

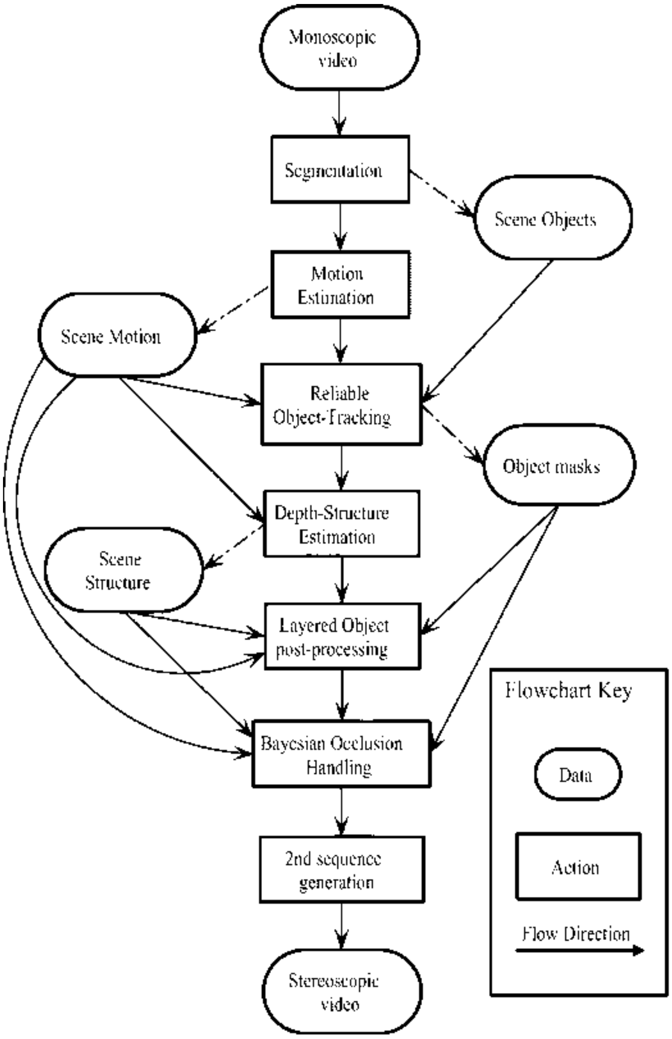


Fig. 1. System flowchart.

group of frames (GOF) of the sequence. The KLT feature tracker is then applied to each object and tracking feature (TF) points are extracted and tracked throughout the image sequence, thus producing the $2N_T \times N$ trajectory array T_{TF}, where N_T and N are the number of the TFs and frames, respectively. The TFs are used for the 2-D motion estimation procedure and are distributed on the visible 2-D surface of each object. It should be noted that if the tracking procedure for a specific TF fails in a specific frame, a new feature is selected, corresponding to the highest texture sharpness over the surface of the object and is tracked throughout the rest of the sequence. The TFs with truncated trajectories are used in the object tracking process, but not during structure estimation, which requires complete trajectories.

To generate an estimate of the motion of object A_m, l_m contour feature (CF) points are extracted, selected (usually on the contour of each object) so as to most accurately describe the shape of each object [12]. It should be noted that the TFs and the CFs are by definition different feature point sets. However, the set of CFs could contain TFs, especially those that lie on or close to the object's contour. The motion of each CF is computed as a weighted average of the motion of the k TFs nearest to it (typical value for k is 3). The contribution of each TF to the motion of the CF diminishes as the distance between them increases. Let $\mathbf{t}_j^i$, $i = 1, \ldots, N_T$, and $\mathbf{c}_j^i$, $i = 1, \ldots, l_m$, represent the tracking and contour features in frame j, respectively. Then

$$\mathbf{c}_{j+1}^i = \mathbf{c}_j^i + \sum_{m=1}^{k} v_m \left(\mathbf{t}_{j+1}^m - \mathbf{t}_j^m\right) \qquad (1)$$

where the weights v_i are computed using

$$\sum_{m=1}^{k} v_m = 1 \quad \text{and} \quad v_1 \left|\mathbf{c}_j^i - \mathbf{t}_j^1\right| = v_2 \left|\mathbf{c}_j^i - \mathbf{t}_j^2\right| = \cdots = v_L \left|\mathbf{c}_j^i - \mathbf{t}_j^k\right|.$$

The final result of this procedure is the generation of the contour features', $2l_m \times N$, trajectory array T_{CT}, which contains the information of the positions of the CFs of object A_m in each frame and thus also information about its mask M_i' for each frame i. The above procedure is carried out in both time directions, thus resulting in extra object masks M_i'' for each frame. These are merged into a final mask, which is simply their mathematical union

$$M_i = M_i' \cup M_i'' \qquad (2)$$

This final mask is refined using the novel prediction-correction algorithm described in the sequel.

A sequence consisting of N frames is illustrated in Fig. 2. The black dots represent the TFs, the small circles the CFs and the "×" marks the predictions of the CFs. Let $\mathbf{S}_{\text{CT}}^i$ be a vector composed of all, unknown, real positions of all CF points in frame i. It should be noted that $\mathbf{S}_{\text{CT}}^i$ is known only in the first and last frame of the GOF. Predictions of its value for the remaining frames are found from $\hat{\mathbf{S}}_{\text{CT}}^i = \mathbf{T}_{\text{CT}}^i$. Then the vector $\mathbf{f}_{\text{disp}}^i$ of the interframe displacement of the CFs between frames i and $i+1$ is calculated from

$$\mathbf{f}_{\text{disp}}^i = \hat{\mathbf{S}}_{\text{CT}}^{i+1} - \hat{\mathbf{S}}_{\text{CT}}^i. \qquad (3)$$

Thus

$$\hat{\mathbf{S}}_{\text{CT}}^N = \mathbf{S}_{\text{CT}}^1 + \sum_{i=1}^{N-1} \mathbf{f}_{\text{disp}}^i. \qquad (4)$$

The prediction for the last frame $\hat{\mathbf{S}}_{\text{CT}}^N$ can be assumed to be linked to the known original contour feature vector $\mathbf{S}_{\text{CT}}^N$ as follows:

$$\mathbf{S}_{\text{CT}}^N = \hat{\mathbf{S}}_{\text{CT}}^N + \mathbf{e} \qquad (5)$$

where a simple, additive error model is assumed

$$\mathbf{e} = \sum_{i=1}^{N-1} D\left(\mathbf{f}_{\text{disp}}^i\right) \cdot \mathbf{w}^i \qquad (6)$$

and $D(f)$ is the $2l_m \times 2l_m$ matrix ($2l_m$ is the size of the CF space) with elements

$$D_{kl}\left(\mathbf{f}_{\text{disp}}^i\right) = \begin{cases} f_{\text{disp},k}^i & k = l \\ 0 & k \neq l \end{cases}.$$

$$\mathbf{f}_{\text{disp}}^i = \left[f_{\text{disp},0}^i, f_{\text{disp},1}^i, \ldots, f_{\text{disp},2l_m-1}^i\right]^T$$

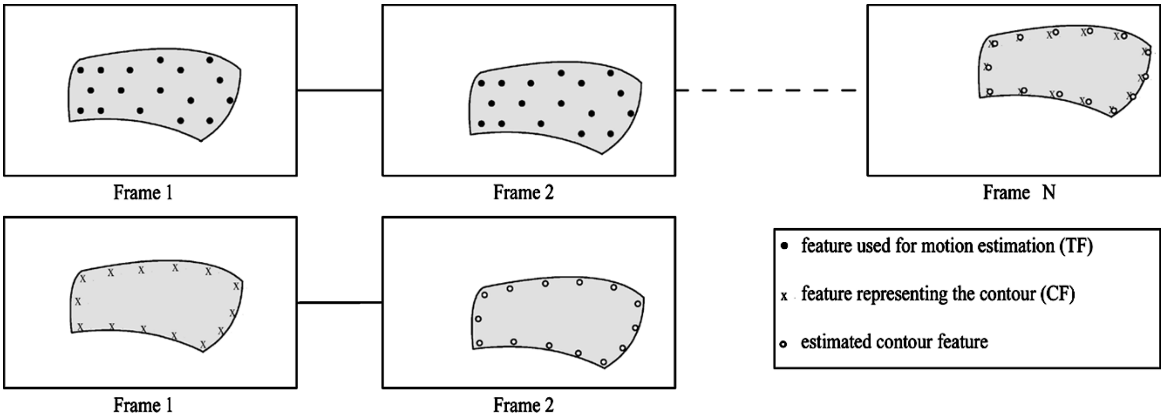

Frame 1
Frame 2
Frame N
Frame 1
Frame 2
• feature used for motion estimation (TF)
x feature representing the contour (CF)
o estimated contour feature

Fig. 2. Contour features' prediction and correction.

and $\mathbf{w}$ is a weight vector with dimension $2l_m \times 1. i = 1.....N-1$.

Equation (5) is applicable only if there exists an accurate mapping between the original contour features S_{CF}^{N} and the predicted features $\hat{S}_{\mathrm{CF}}^{N}$, i.e., vectors S_{CF}^{N} and $\hat{S}_{\mathrm{CF}}^{N}$ have to be of the same size and their corresponding elements should index to the same CF. In order to achieve such a mapping, back-tracking of the predicted CFs of the last frame is executed, using the TFs of the backward motion estimation. When reaching the first frame, the position of the original CFs is known since this is the frame where they were defined. The inverse procedure is executed for the masks defined in the last frame of the GOF.

In (6) it is assumed that the accumulative error $\mathbf{e}$ of the CF motion is a weighted summation of the measurements $\hat{\mathbf{S}}_{\mathrm{CF}}^{N}$. This assumption stipulates that the estimation error is proportional to the projection of the TF space to the CF space. On combining (4), (5), and (6)

$$\hat{\mathbf{S}}_{\mathrm{CF}}^{N} = \mathbf{S}_{\mathrm{CF}}^{1} + \sum_{i=1}^{N-1} \mathbf{f}_{\mathrm{disp}}^{i} + \sum_{i=1}^{N-1} D\left(\mathbf{f}_{\mathrm{disp}}^{i}\right) \cdot \mathbf{w}^{i}. \tag{7}$$

The partial derivatives of $\mathbf{e}$ with respect to $\mathbf{w}^i$ are now computed, so as to employ Newton–Raphson techniques for the iterative minimization of the resulting error. Let $\mathbf{w}^i(j)$ and $\mathbf{e}(j)$ denote the weight and error vectors, respectively, at iteration step j. Then

$$\begin{aligned}\frac{\partial \mathbf{e}}{\partial \mathbf{w}^i} &= \frac{\partial\left(\sum_{k=1}^{N-1} D\left(\mathbf{f}_{\mathrm{disp}}^{k}\right) \cdot \mathbf{w}^{k}(j)\right)}{\partial \mathbf{w}^{i}(j)} = \frac{\partial\left(D\left(\mathbf{f}_{\mathrm{disp}}^{i}\right) \cdot \mathbf{w}^{i}(j)\right)}{\partial \mathbf{w}^{i}(j)} \\ &= \frac{\partial \mathbf{c}^{i}(j)}{\partial \mathbf{w}^{i}(j)} = J\left(\frac{\partial \mathbf{c}^{i}(j)}{\partial \mathbf{w}^{i}(j)}\right)\end{aligned} \tag{8}$$

where

$$\mathbf{c}^{i}(j) = D\left(\mathbf{f}_{\mathrm{disp}}^{i}\right) \cdot \mathbf{w}^{i}(j). \tag{9}$$

The updated $\mathbf{w}^i(j+1)$ vectors are then given from the following equation:

$$\mathbf{w}^{i}(j+1) = \mathbf{w}^{i}(j) - J^{-1}\left(\frac{\partial \mathbf{c}^{i}(j)}{\partial \mathbf{w}^{i}(j)}\right) \cdot \mathbf{e}. \tag{10}$$

Equation (9) implies that each element $c_k^i(j). k = 0.....2l_m - 1$, of vector $\mathbf{c}^i(j)$ is independent of all elements of vector $\mathbf{w}^i(j)$ apart from $w_k^i(j)$. Thus, the above Jacobian matrix can be simplified as follows:

$$J\left(\frac{\partial \mathbf{c}^{i}}{\partial \mathbf{w}^{i}}\right) = D\left(\mathbf{f}_{\mathrm{disp}}^{i}\right) \tag{11}$$

Thus

$$\mathbf{w}^{i}(j+1) = \mathbf{w}^{i}(j) - D^{-1}\left(\mathbf{f}_{\mathrm{disp}}^{i}\right) \cdot \mathbf{e}. \tag{12}$$

Summarizing, the recursive reliable object tracking algorithm is applied as follows.

- The f_{disp}^{i} vectors are computed using (3).
- The error $\mathbf{e}$ is computed (5) and its partial derivatives with respect to $\mathbf{w}$ are calculated (8).
- The weight vectors $\mathbf{w}$ are updated (12).
- This procedure is repeated iteratively until $|\mathbf{e}| \leq \delta$, where δ is an error threshold.

As seen by the experimental results, this method produces negligible error after a number of iteration steps if the motion variance between consecutive frames is not very high and if the additive error model between consecutive frames remains valid. In practice, the interframe translation can be assumed to be small and, assuming that the feature points TF are tracked correctly, the above error is small and additive, so that the above constraints are met.

Obviously, not all of the TFs contain the same information about the translation and deformation of the object's contour. Thus, the TFs which are the closest to at least one point of the CF, are used for the retrieval of the CFs motion throughout the GOF.

B. *Noncausal Bayesian Classification of Occluded Points*

When trying to generate a stereoscopic video from a monoscopic one, even if the scene structure is estimated correctly and reliable object masks are extracted, problems still exist concerning the handling of occluded areas [13]. Information about occluded objects is not directly available and therefore as the 3-D scene is projected to a new virtual plane, the existence of areas, which are not assigned an intensity value and are not classified to any object, is inevitable. Fig. 3(b) illustrates a scene

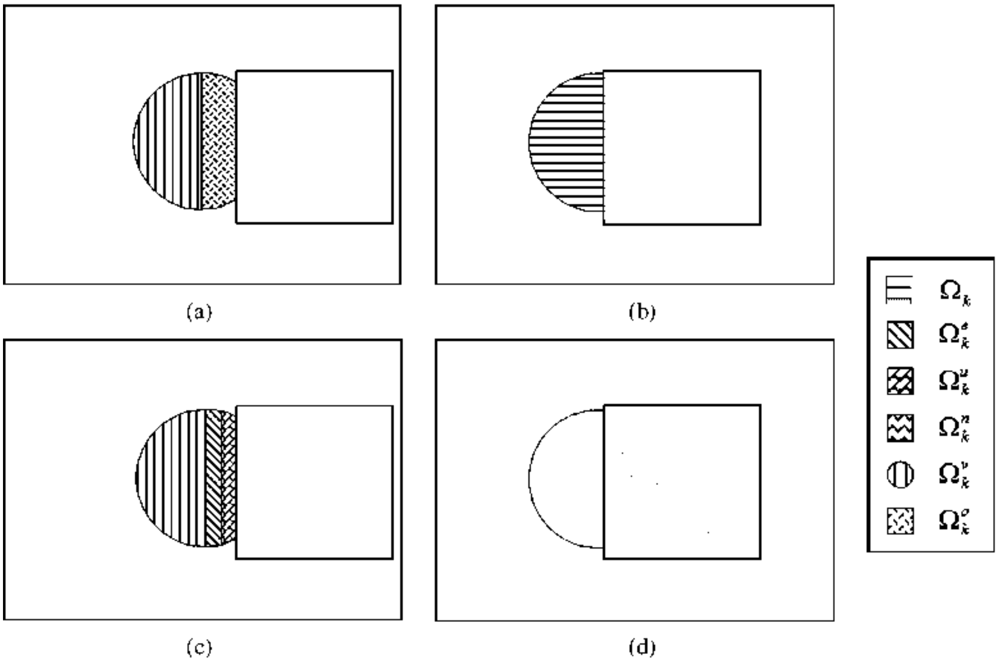


Fig. 3. Occlusion analysis. (a) Left frame of a stereo pair. (b) Right frame of a stereo pair. (c) Categorization of pixels, which were occluded in the right frame. Notice that $\Omega_k^r = \Omega_k^s \cup \Omega_k^n$. (d) Possible noncircular object occluded by the square (right frame). $[\Omega_k]$:\ Set of points belonging to the circle. $[\Omega_k^v]$: Set of visible points of the circle in the original frame 4b. $[\Omega_k^o]$: Set of occluded points of the circle in the original frame 4b. $[\Omega_k^u]$: Set of occluded points of the circle, which will become or became visible in future or past frames of the original sequence. $[\Omega_k^s]$: Set of occluded points of the circle, which remain occluded for the entire sequence, but are visible for at least one frame of the generated (left) sequence. $[\Omega_k^n]$: Set of occluded points of the circle, which remain occluded for the entire sequence and do not become visible in any frame of the generated sequence.

containing two objects, a square and a circle surface-shaped objects. Half of the circle surface is occluded by the square. In Fig. 3(a) the left reconstructed image can be seen, assuming that the original image [Fig. 3(b)] represents the right image. The part of the circle, which was occluded in the right frame should become in the reconstructed image visible, but there is still no information about its texture. No information exists even about the object to which the newly visible points belong. Also, there is no *a priori* knowledge that the left object in Fig. 3(b) is a full circle, partly occluded by the right object. The object could be a half circle or even something very different as seen in Fig. 3(d). A procedure is proposed in the sequel to estimate occluded areas and assign intensities to them using a Bayesian framework.

In the following, all procedures are executed for every object k in the scene, $k = 1.\ldots.K_N$, where K_N is the number of objects. To simplify the notation, the dependence on the index k will not be explicitly denoted in the sequel. Let Ω denote the support range of points belonging to object k (e.g., the left object). Let also Ω^v and Ω^o denote the support ranges of visible and occluded points in the specific frame. Obviously, $\Omega = \Omega^v \cup \Omega^o. \Omega^v \cap \Omega^o = \emptyset$.

Assuming that the already existing frame of Fig. 3(b) is the right part of a stereoscopic pair and that the square is closer to the camera than the circle, in the left part, the square will be displaced more than the circle, as illustrated in Fig. 3(a). As a result, some of the points that become visible in the left image [Fig. 3(a)], are not visible in the right image [Fig. 3(b)]. These points cannot be classified to an object and it is difficult to assign an intensity value to them.

In [14] a Bayesian approach has been developed to classify the points of a stereo pair to three subsets, visible foreground, visible background and areas visible only in one image of the stereo pair. In the following we shall describe a Bayesian occlusion handling framework applicable to monoscopic image sequences. We start with the division of the set of occluded points of the left object in the right image Ω^o into the following two subsets as illustrated in Fig. 3(c).

1) $\Omega^e \subset \Omega^o$: The set of occluded points of the left object, which will become visible in the future or became visible in the past frames of the original sequence.
2) $\Omega^r \subset \Omega^o$: The set of occluded points of the left object, which remain occluded for the entire sequence.

Now, for every point $\mathbf{n} \in \Omega^e$ in frame i

$$\exists L \in F_f = \{i+1.i+2.\ldots.N\} \text{ or}$$
$$L \in F_p = \{i-1.i-2.\ldots.1\}$$

so that, if $L \in F_f$:

$$I^i(\mathbf{n}) = I^{i+L}\left(\mathbf{n} + \sum_{j=i}^{L-1} \mathbf{g}_{\mathbf{n}}^j\right) \tag{13}$$

if $L \in F_p$:

$$I^i(\mathbf{n}) = I^{i-L}\left(\mathbf{n} - \sum_{j=L}^{i-1} \mathbf{g}_{\mathbf{n}.}^j\right) \tag{14}$$

where $I^i(\mathbf{n})$ is the intensity of point $\mathbf{n}$, in frame $i. F_f$, and F_p are the subsets of future and past frames, respectively, N is the total number of frames and i the index of the current frame. $\mathbf{g}_{\mathbf{n}}^j$ is a function, which utilizes the existing motion estimation data

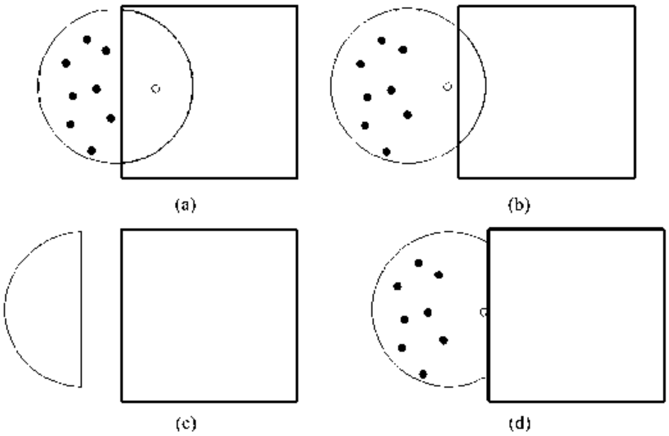


Fig. 4. (a)–(b) Two-frame sequence. (c) Generated image for the 1st frame prior noncausal post processing. Combined with image (a) results in a stereo pair for the first frame. (d) Generated image for the 1st frame after noncausal post processing. Combined with image (a) results in the final stereo pair for the first frame.

to retrieve the estimated motion of point $\mathbf{n}$ between frames $j-1$ and j if $\mathbf{n}$ belongs to object k using a weighted summation of the nearest to point $\mathbf{n}$, TF points. It computes the 2-D displacement vector of point $\mathbf{n}$

$$\mathbf{g}_{\mathbf{n}}^{j} = \sum_{i=1}^{L} v_i \mathbf{d}_i^j \tag{15}$$

The weights v_i are computed using

$$\sum_{i=1}^{L} v_i = 1$$
$$v_1|\mathbf{n}-\mathbf{n}_1| = v_2|\mathbf{n}-\mathbf{n}_2| = \cdots = v_L|\mathbf{n}-\mathbf{n}_L|$$

where the points $\mathbf{n}_i.\ i = 1.\ldots.L$ are the L closest TF points to $\mathbf{n}$, and the vectors $\mathbf{d}_i^j.\ i = 1.\ldots.L$, contain the displacement of features between frames $j-1$ and j.

Equations (13) and (14) show that if a point $\mathbf{n}$ belongs to object k and becomes visible in a future or past frame, it can be assigned an intensity value, using the motion estimation data already available. For example, Fig. 4(a) and (b) illustrate a two-frame sequence. The black dots are the motion estimation features (TF) and the white dot is a point of the left object. The white dot is not visible in the first frame [Fig. 4(a)] but becomes visible in the second [Fig. 4(b)]. Moreover, Fig. 4(d) illustrates the stereo pair of the first frame [Fig. 4(a)], where the white dot becomes visible. At this moment there is no information about the texture of the white dot, unless the second frame is used, where it becomes visible. Therefore, the function "$\mathbf{g}_{\mathbf{n}}^{j}$" estimates the motion of the "hidden" white dot using the motion estimation already available (black dots, $\mathbf{f}_{\mathrm{disp}}^{i}$). If the motion of the white dot is known, (13) or (14) can be applied to retrieve its texture.

Concerning the technique described above, the following issue may arise: How do we know that the occluded point, which becomes visible in the generated image belongs to object k, so as to apply (13) and (14)? The algorithm described in Section II-A provides the answer to this question. Assuming rigid motion of the occluded parts of each object, if a point is visible in a frame and belongs to an object, it will still belong to the same object in the next frame even if it becomes occluded. Moreover, the subsets of object masks in each frame are not disjoint, hence, a point in the image plane can belong to more than one object. The point closer to the camera is the last displayed (Z-Buffering principle).

Every point $\mathbf{n} \in \Omega^r$, which becomes visible in the reconstructed frame, must be assigned to an object. For this task a novel Bayesian method is proposed, based on the selection of the following hypotheses

$$H_k\text{: } \mathbf{n} \text{ belongs to object } k. \quad \mathbf{S}(\mathbf{n}) = \text{Extrapolate}\,(\Omega_k)$$

where $k = 1.\ldots.K.\ K$ is the number of scene objects and Extrapolate( ) a bilinear extrapolation function. According to the Bayes decision test, hypothesis H_k will be selected if

$$r_k(\mathbf{n}.\Omega_k) < r_j(\mathbf{n}.\Omega_j). \quad \forall j \neq k \tag{16}$$

where $r_k(\mathbf{n}.\Omega_k)$ is the average cost of accepting hypothesis H_k and can be defined as follows:

$$r_k(\mathbf{n}.\Omega_k) = \sum_{i=1}^{N_A} G_{ik} p(\mathbf{n}.H_i) = \sum_{i=1}^{N_A} G_{ik} p(\mathbf{n}\,|\,H_i) p(H_i) \tag{17}$$

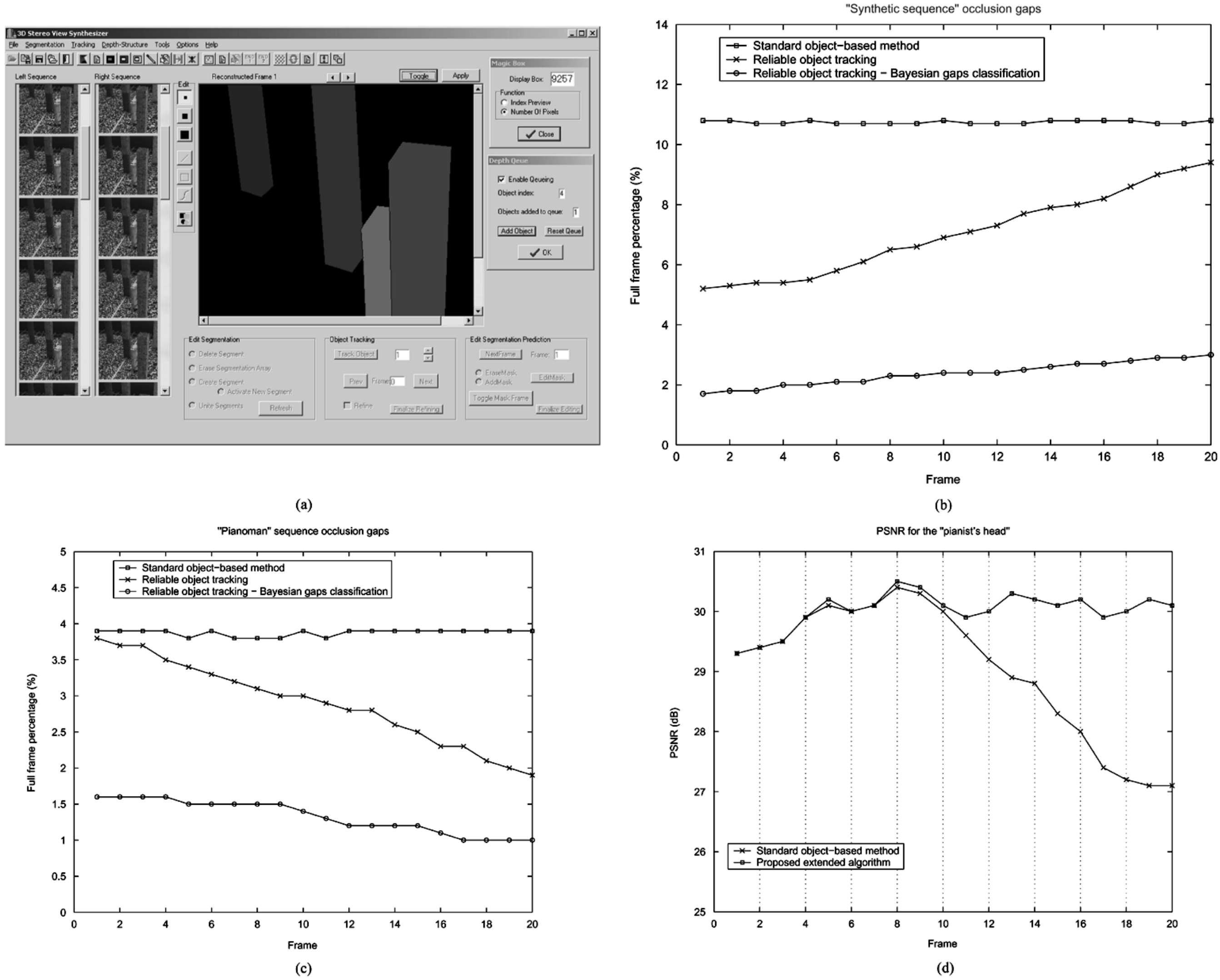
where G_{ik} is the cost of accepting H_k when H_i is true, $p(\mathbf{n}.H_i)$ is the mutual probability of $\mathbf{n}$ and H_i, and N_A the number of scene objects. It is very reasonable to assume that $G_{ii} = 0$ (zero cost for proper classification) and $G_{ik} = G = 1.\forall i \neq k$, since erroneous classifications are equally noxious. Then, on assuming that $H_i = (1/N_A)$, (17) is trivially seen to be minimized if the hypothesis H_k is selected when

$$p(\mathbf{n}\,|\,H_k) > p(\mathbf{n}\,|\,H_j). \quad \forall k \neq j \tag{18}$$

When adopting the ML criterion, the *a priori* probability $p(H_i)$ is defined to be $p(H_i) = 1/N_A$. It could be also defined to be proportional to the size of the object. However, in this case the assigning of occluded points to small objects would be strongly discouraged. Therefore, all the information regarding the hypothesis selection is inserted in the formula of the probability $p(\mathbf{n}\,|\,H_i)$, which is modeled as follows:

$$p(\mathbf{n}\,|\,H_k) = \left|\frac{d\left(\Omega_k^v \cup \Omega_k^c\right)}{dt}\right| \frac{1}{E\left\{d_{\mathbf{i},\mathbf{n}}^k \,\middle|\, d_{\mathbf{i},\mathbf{n}}^k \in D_{p,\mathbf{n}}^k\right\}} \tag{19}$$

where $(\partial\left(\Omega_k^v \cup \Omega_k^c\right)/\partial t)$ is the time derivative of the support range of object's k visible pixels, $d_{\mathbf{i},\mathbf{n}}^k$ the distance of point $\mathbf{n}$ from contour point $\mathbf{i}$ of object k. $D_{p,\mathbf{n}}^k$ is the set of $N = pC$ contour points closest to point $\mathbf{n}$ of object k. C is the total number of contour points and $E\{\cdot\,|\,\cdot\}$ represents a conditional mean value operator. The right-hand side of (19) is normalized so as to represent a probability value. The time derivative part of (19) represents the dynamic alteration in the size of the visible parts [Fig. 3(c)] of object k in the area close to $\mathbf{n}$. If this size is changing, object k is likely to be involved in occlusion (Fig. 3),



(a)

(b)

(c)

(d)

Fig. 5. (a) "3DSVS" snapshot. (b)–(c) Percentage of remaining undefined occlusion gaps or incorrectly assigned to an object in the "tower" and "pianist" sequence, respectively. (d) PSNR for the "pianist's head" along the sequence.

hence, the probability that the emerging points belong to this object increases. If however the size remains constant the object is probably in the foreground and it is less possible that the emerging points belong to it. Thus, the probability $p(\mathbf{n} \mid H_k)$ is modeled as proportional to it. The conditional mean value part represents the mean value of the distance of point $\mathbf{n}$ from the $N = pC$ contour points of object k, which are closest to it. The probability $p(\mathbf{n} \mid H_k)$ is modeled to be inversely proportional to this mean distance.

The above test is performed M times for different values of p and the decision is made on the basis of a strong majority rule. Specifically, a point is classified to object j if H_j results at least $hM(0.8 < h < 1)$ times. If this threshold is not reached then the point is assigned to the background object. Typical values are $M = 10, p = 0.05 \cdot i, i = 5, 6, \ldots, 10$.

C. Stereo View Generation

As described in the previous Section, the EKF algorithm, which makes use of the TFs' trajectories only, is applied separately to each object for depth estimation. The EKF estimator is seen to produce less accurate estimates in the first few frames of the sequence. Considering also the fact that the initial depth value is chosen arbitrarily for the first GOF and using interpolation for the next GOFs, the low accuracy of the EKF in the first frames could be a problem for the generation of the stereoscopic video if the EKF estimator were applied only unidirectionally. However, in the present framework, the EKF procedure is executed bidirectionally, thus resulting in accurate estimates for the first as well as for the last frames. Next, depth values are assigned to every pixel of the object using interpolation based on Delaunay triangulation. This results in an interpolation procedure assigning depth values only for the object points that lie inside the polygon defined by the feature points. The depth values of the object points that lie outside of it are computed as a weighted average of the depths of the k TFs nearest to them (typical value for k is 3). The contribution of each TF to the depth of each point decreases as the distance between them increases. Furthermore, it is worth noting that using the proposed layered approach for 3-D structure estimation, the EKF produces more robust results since the high 3-D shape frequencies appear at the object boundaries. When each object is treated separately the depth estimation becomes more accurate.

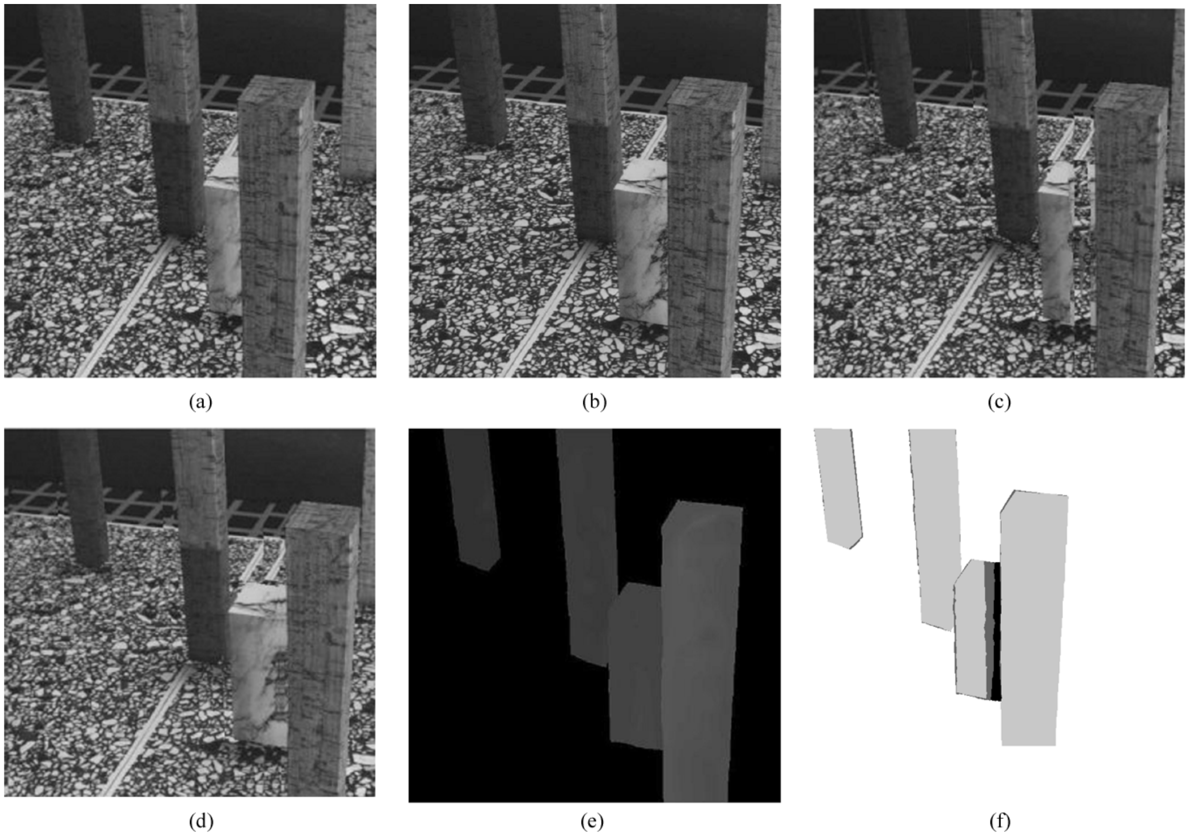


Fig. 6. (a) First frame. (b) 20th frame. (c) Generated frame with the standard object-based method. (d) Generated frame with the proposed method. (e) Dense depth map. (f) Occlusion masks for the four tower objects; light gray, dark gray, and black represent Ω^{u}, Ω^{c}, and Ω^{ur}, respectively.

As in the case of the presented experiments the processed GOF contains about 20–30 frames. Because of this restricted size, objects appearing in the first frame are highly likely to appear in the last frame as well, and vice versa. However, in the rare case of a very fast moving object, which is identified in only one of the first or last frame the algorithm will reduce into unidirectional object tracking, which will be forward or backward depending on whether the object is identified in the first or last frame of the GOF. It should be noted that the results of the intermediate procedures of the proposed method can, if needed, be manually refined, e.g., for scenes with extremely low texture, where motion estimation may fail. The developed authoring tool [Fig. 5(a)] is used for this purpose.

Using the produced dense depth map and assuming parallel geometry, i.e., assuming that the second virtual camera is displaced only along the horizontal axis, the 3-D points are projected in the second virtual image plane using the Z-Buffer algorithm. The final step is to produce the stereoscopic video. After implementing all above techniques the stereo-video is generated by reassembling the processed scenes.

III. Experimental Results

In order to test the proposed algorithm, several sequences were processed. In the following the result of two experiments are reported, which were conducted using the standard object-based method [6] as well as with the extended method proposed in Section II. All results can be seen in high resolution online [15].

The synthetic "tower" and the real "pianist" sequences consisting of 20 frames with size 512 × 512 and 720 × 576, respectively, were used. The main objects are identified in the first and last frame and after extracting and tracking the feature points, reliable object masks are produced for each frame. The tracked features are also used in the EKF implementation, thus producing dense depth maps. By increasing the stereo baseline, depth differences become more enhanced. Subjectively, the stereoscopic viewing error was seen to be negligible. When present, this error is generated from gaps, or erroneous interpolation at the occluded areas of the scene. However we note that the human visual system does not recognize this error, if one of the two images is perfect and the errors in the second image not very high.

Fig. 5(b) and (c) present the percentage of image points not registered to an object for each frame for the "tower" and "pianist" scene, respectively. The difference in the efficiency of the standard object-based algorithm [6] and the proposed extended method is obvious. If there are several occluding objects in a scene, this difference will become more pronounced.

For each of the two sequences, Figs. 6 and 7 illustrate the first and last frame of the sequence, the reconstructed stereo frame obtained by executing the standard object-based method [6], as well as the one formed by executing the proposed extended algorithm described in detail in Section II, the accurate dense depth map and the occlusion masks. Typical anaglyph images are available in full resolution at [15] in order to illustrate the stereoscopic effect.

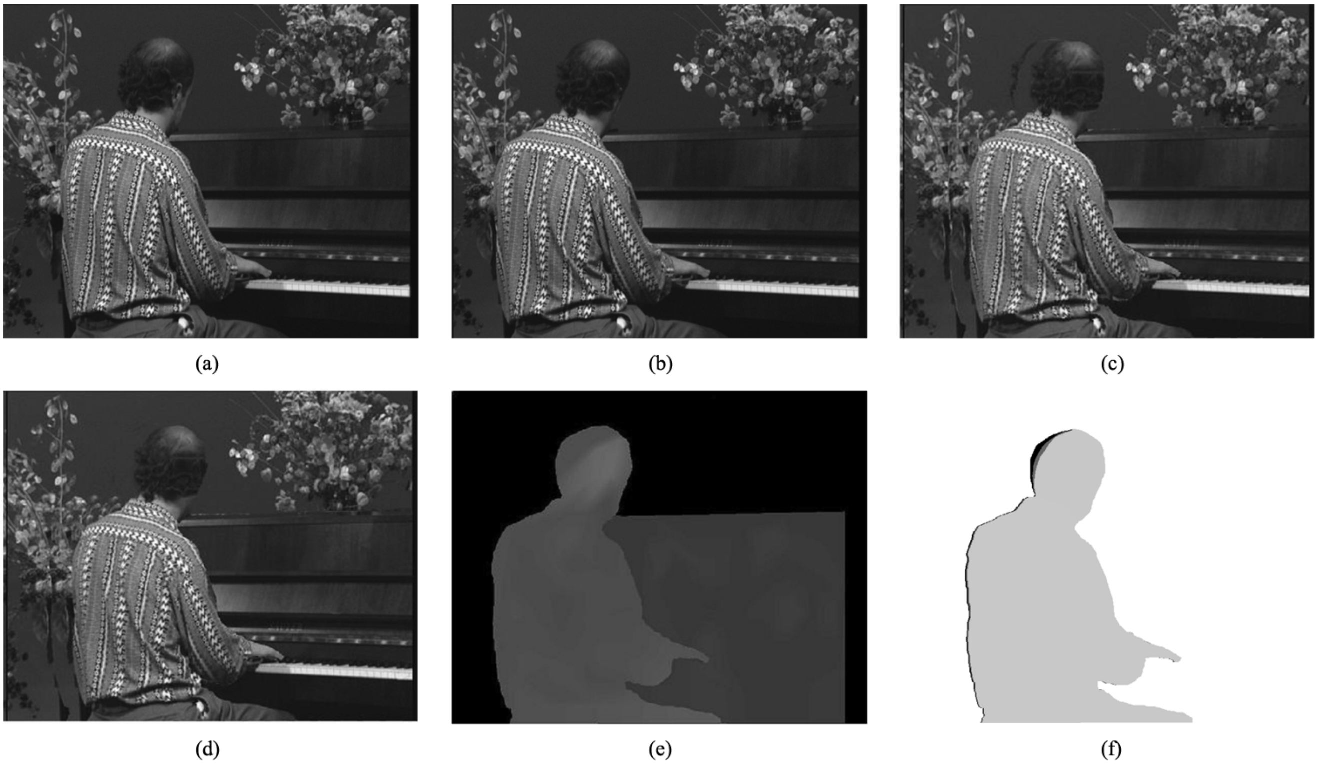


Fig. 7. (a) First frame. (b) 20th frame. (c) Generated frame with the standard object-based method. (d) Generated frame with the proposed extended method. (e) Dense depth map. (f) Occlusion masks for the four pianist scene objects; light gray, dark gray, and black represent Ω^{ii}, Ω^{i}, and Ω^{ir}, respectively.

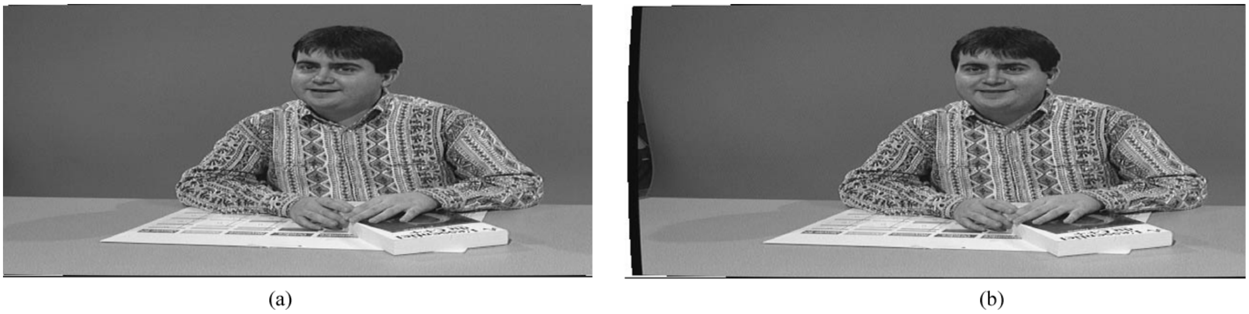


Fig. 8. Two frames of the Ludo sequence.

The quality differential between the two reconstructed frames is obvious for both experiments. In the "tower" experiment, the smaller white tower is occluded by the larger one, which is closer to the image plane and a part of it appears in the latter frames of the sequence. These emerging areas produce inaccurate object masks, if the sequence is processed with the simple object-based method. By applying the proposed extended method, the resulting mask is very accurate and the produced image is much more realistic.

Figs. 7(a) and (b) show that there is a rotation in the pianist's head between the first and the last frame. Specifically, the rotation becomes significant in the 10th frame and produces occluded and emerging areas in the right and left side of the pianist's head, respectively. The difference between the standard object-based algorithm and the proposed extended method can be seen in the peak signal-to-noise ratio (PSNR) chart of the pianist's head of Fig. 5(d). Notice that PSNR can be extracted because the original left image sequence is available. While the quality of the sequence falls after the tenth frame using the standard algorithm, it stays in high levels using the proposed method.

In order to obtain quantitative results about the depth estimation, the proposed algorithm was applied to the "Ludo" sequence (Fig. 8), in which the distance of the person from the camera is known and thus quantitative results about the accuracy in the depth estimation can be extracted. For this sequence the PSNR of the 3-D structure estimation was found to be $\mathrm{PSNR} = 33.9$ dB and the mean square error in real world dimensions was 1.39 cm or 1.3% of the average depth of the scene, which was 153 cm.

IV. Conclusion

A robust method was presented for the generation of a stereoscopic image sequence using as input only a monoscopic video. The scene is divided into objects and after inserting and tracking a number of features, an efficient algorithm for reliably tracking object masks is applied for every object and every frame of the sequence. Then the EKF based algorithm produces estimates of scene structure and the second virtual image is synthesized using a novel Bayesian framework to classify the occluded points to the objects. Finally, the 3-D points are projected to a new virtual image plane. With the proposed algorithm the problems posed by the appearance of occlusions and emerging areas are dealt with by using a direct method for the calculation of the intensity values of points which become visible in future or past frames, combined with a probabilistic algorithm for points which do not. Experimental results illustrate the robustness of this approach.

REFERENCES

[1] A. Azarbayejani and A. Pentland, "Recursive estimation of motion, structure, and focal length," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 17, no. 6, pp. 562–575, Jun. 1995.

[2] J. Weng, N. Ahuja, and T. S. Huang, "Optimal motion and structure estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, no. 9, pp. 864–884, Sep. 1993.

[3] R. Szelinski and S. B. Kang, "Shape ambiguities in structure from motion," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 5, pp. 506–512, May 1997.

[4] L. Quan, F. Kahl, and A. Heyden, "Minimal projective reconstruction including missing data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 4, pp. 418–424, Apr. 2001.

[5] Z. Liu, Z. Zhang, C. Jacobs, and M. Cohen, "Rapid modeling of animated faces from video," *J. Visual. Comput. Animation*, vol. 12, no. 4, pp. 227–240, 2001.

[6] S. Diplaris, N. Grammalidis, D. Tzovaras, and M. G. Strintzis, "Generation of stereoscopic image sequences using structure and rigid motion estimation by extended kalman filters," in *IEEE ICME*, vol. 2, Lausanne, Switzerland, 2002, pp. 233–236.

[7] N. Qian and R. A. Andersen, "A physiological model for motion-stereo integration and a unified explanation of the pulfrich-like phenomena," *Vis. Res.*, vol. 37, pp. 1683–1698, 1997.

[8] T. Jebara, A. Azarbayejani, and A. Pentland, "3-D structure from 2-D motion," *IEEE Signal Process. Mag.*, vol. 16, no. 3, pp. 66–84, May 1999.

[9] T. J. Broida, S. Chandrashekhar, and R. Chellappa, "Recursive estimation of 3d motion from a monocular image sequence," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 26, no. 4, pp. 639–656, Jul. 1990.

[10] V. Mezaris, I. Kompatsiaris, and M. G. Strintzis, "Still image segmentation tools for object-based multimedia applications," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 18, no. 4, pp. 701–725, Jun. 2004.

[11] J. Shi and C. Tomasi, "Good features to track," in *{roc. IEEE Int. Conf. Computer Vision and Pattern Recognition*, Seattle, WA, Jun. 1994, pp. 593–600.

[12] G. Iannizzotto and L. Vita, "On-line object tracking for color video analysis," *Real-Time Imaging*, vol. 8, no. 2, pp. 145–155, Apr. 2002.

[13] K. P. Lim, A. Das, and M. N. Chong, "Estimation of occlusion and dense motion fields in a bidirectional Bayesian framework," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 712–718, May 2002.

[14] G. A. Triantafyllidis, D. Tzovaras, and M. G. Strintzis, "Occlusion and visible background and foreground areas in stereo: A Bayesian approach," *IEEE Trans. Circuits Syst. Video Technol, Special Issue on 3-D Video Techn.*, vol. 10, no. 4, pp. 563–576, Jun. 2000.

[15] (2005) Stereoscopic Video Generation. [Online]. Available: http://server-2.iti.gr/moustak/stereo.htm