Multiview Fusion for Canonical View Generation Based on Homography Constraints

Ambrish Tyagi
tyagia@cse.ohio-state.edu

James W. Davis
jwdavis@cse.ohio-state.edu

Mark Keck
keck@cse.ohio-state.edu

Department of Computer Science and Engineering
Ohio State University
Columbus, OH 43210, USA

ABSTRACT

Activity and gait recognition are among the various applications that necessitate view-specific input. In a real surveillance scenario it is impractical to assume that the desired canonical view will always be available. We present a framework to generate the canonical view of any translating object in a scene monitored by multiple cameras. The method is capable of recovering this view despite the fact that none of the cameras can see it individually. In this two step process, first the camera and scene geometry is used to identify the sagittal plane of the object, which is used to define the canonical view. Next, each original view is warped to the canonical view through planar homographies learnt from geometric constraints. The warped images are then combined by way of evidence fusion to recover the *shape energy map*, which is used to obtain the final binary silhouette of the object's shape. Results presented for various indoor and outdoor sequences demonstrate the efficacy of this method in generating the shape of the object as seen from the canonical view, while resolving occlusions.

Categories and Subject Descriptors

I.4.8 [**Image Processing and Computer Vision**]: Scene Analysis—*Motion, Sensor fusion, Shape*; I.2.10 [**Artificial Intelligence**]: Vision and Scene Understanding—*3D/stereo scene analysis, Shape, Video analysis*

General Terms

Algorithms, Security

Keywords

Canonical view generation, multiview fusion, homography, *shape energy map*, sagittal plane

VSSN'06, October 27, 2006, Santa Barbara, California, USA.

1. INTRODUCTION

Surveillance is an important area of research in computer vision that deals with moving objects like people, animals, vehicles, etc. Other scenarios dealing with locomotion include Unmanned Aerial Vehicles (UAV), sport videos, smart houses and meeting rooms. One important requirement in all these applications is to recognize these objects and classify their actions (for e.g., Walking, Running, Jumping, etc.).

Bulk of research in the area of action and gait recognition still requires view based inputs (e.g. [3, 4, 5, 6, 10]), with an exception to some efforts in view-independent action recognition ([21, 18]). All view based algorithms require the object of interest to be imaged from a particular vantage point which is not always possible in an unconstrained environment. The first obvious solution to this problem would involve building the full 3D model of the object/scene using multiple cameras and generating the required canonical view. Such an approach becomes impractical for articulated object motion sequences as dense correspondence required for 3D reconstruction is not feasible.

This begs the question: Is the 3D reconstruction indispensable for this task, or, could we instead use the basic 3D scene structure and employ geometric constraints to generate novel views of any object and by-pass the full reconstruction?

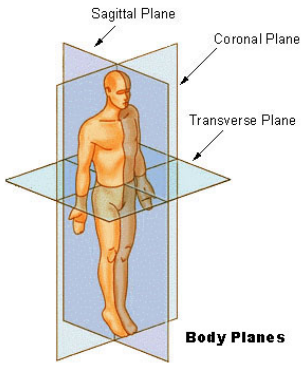

Figure 1: Imaginary body planes (Image from http://training.seer.cancer.gov)

We present a technique to generate the canonical view of any object translating freely in an area observed by multiple cameras. To demonstrate the approach we select the side-view of the object as the canonical view. We can define the side-view with respect to the sagittal plane of the object. Sagittal plane for a symmetric 3D object is the longitudinal plane passing through the object center dividing it into left and right sections (fig 1). The side view of an object has been found to be very useful in the task of action recognition and gait analysis. If any other view is deemed canonical (e.g. Bounded Canonical Sets [7]) then it can be calculated with reference to the sagittal plane.

First, we identify the scene geometry and the ground plane that (along with other geometric constraints) are used to determine the sagittal plane. The canonical view is defined with respect to this reference plane and a virtual camera is synthesized to generate this view. Next, each observed view is warped to the virtual-view through planar homographies calculated with respect to the sagittal plane. This mapping ensures that the pixels belonging to the object shape, which are on (or near) the sagittal plane, are mapped consistently with no (or small) parallax. Structure far away from the sagittal plane is mapped at different locations due to parallax. This property of planar homographies enable us to collect evidence of object shape as projected on its sagittal plane (side-view) from different views. Information is combined from the original views by the means of evidence fusion. We demonstrate the working of this approach through experimentation done on both indoor and outdoor sequences.

Even though the cost to deploy cameras in urban and military scenarios is decreasing, still we cannot always place enough cameras in order to accurately reconstruct the world in 3D. The method presented here on the other hand require a modest number of cameras (≥ 2) from which it fuses information, with a tradeoff between robustness to occlusions vs. number of cameras. Moreover, apart from the basic claim of generating the canonical view of an object, the method can be used for automatic view based shape tracking without the use of complex methods such as snakes, level sets, etc.

The remainder of this paper is organized as follows. We being with a review of related work on multi-camera systems and novel view generation in Sect. 2. The main algorithm is described in Sect. 3 followed by experimental results in Sect. 4. We finally summarize the work and conclude in Sect. 5.

2. RELATED WORK

The problem of identifying the 3D structure from 2D images have been studied for more than 20 years [11, 12]. With a better understanding of the 3D geometry several methods for recovering the Structure from Motion (SfM) of 3D static scene have been proposed. An important class of algorithms deal with discovering the Shape from Silhouette (SFS) using multiple cameras [24]. SFS is one way to reconstruct the 3D model of an object (essentially its visual hull) that can eventually be projected to any arbitrary view. The quality of 3D reconstruction in SFS is limited by the quality of the reconstructed visual hull, which is in turn depended on the number of cameras. Typically one requires a large number of cameras (10-20) for a reasonable reconstruction using standard SFS algorithms. We want to avoid using such a high number of cameras with the intension of just recovering a particular view of interest rather than the entire 3D structure of the object.

Another body of work has been in the area of novel view generation from a given set of reference views [1, 8]. Algorithms to generate novel views of *static* objects from multiple cameras have been presented. They are based on dense object correspondence that effectively can lead to 3D structure recovery and hence the novel pose. Such methods are not suitable for articulated object motion sequences and wide baseline cameras.

One related work on side view generation from monocular sequences was presented in [2]. It provides a simple method to learn the intrinsic camera parameters and estimate the direction of motion from the video sequence. A geometric correction can applied to warp the observed pose to the reference side view. The is suitable only for limited cases where the direction of motion is roughly fronto-parallel to the camera and there are no occlusions. Our method on the other hand can handle any arbitrary direction of motion and is capable of (self- or external-) occlusion reasoning.

Finally, the idea of parallax induced by planar homographies has been exploited in various applications including shadow removal [13] and people tracking [14]. Our proposed framework is motivated by the problems and issues raised by these approaches and seeks to provide a simple and elegant method to identify one of the principal planes of a translating object. We further exploit this orientation information to generate the object pose as seen from the canonical view.

3. SAGITTAL PLANE DISCOVERY AND NOVEL VIEW GENERATION

In this section we describe the various steps involved in obtaining the shape of an object as it would be seen from a camera directly looking at its sagittal plane. The algorithm goes through the process of camera calibration, which is a one time process until the viewpoints are changed. We then impose geometric constraints to obtain the position and orientation of the sagittal plane given the knowledge about the ground plane. The probabilistic foreground maps for each original view are warped to the novel view through the homographies of the sagittal plane and finally fused and thresholded to produce the shape silhouette. A flowchart describing the entire pipeline of this framework is shown in fig. 2.

3.1 Notation

In the remaining sections of this paper, we will use the following notational standards. Matrices will be denoted with a capital boldface, like $\mathbf{M}$, and the vectors will be denoted with lowercase boldface, like $\mathbf{v}$. Image matrices will be denoted with calligraphic font such as $\mathcal{I}$. Any mathematical operator used in conjunction with images will denote pixel-wise operations.

We follow the standard notation for multiple view geometry [9], where a 3D point in space, $\mathbf{X}_j$, is viewed by a set of cameras with projection matrices $\mathbf{P}^i$. The image projection of the j-th point as seen by the i-th camera is denoted by a 3-vector $\mathbf{x}^i_j = (x^i_j, y^i_j, 1)^T$.

3.2 Camera and Scene Geometry

Given m views of a scene we first calibrate the multi-camera system to estimate the camera and scene geometry.

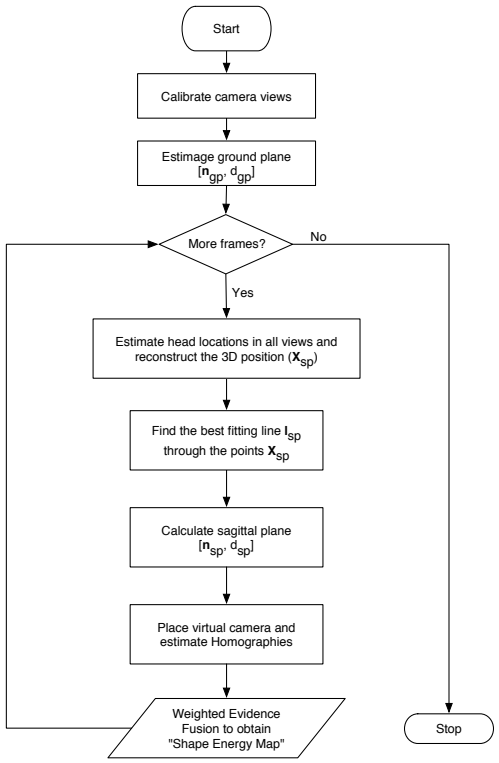

Figure 2: Flowchart

We assume that $n > 3$ correspondences are known across all the views, of which, at least 3 lie on the ground plane. Following a "stratified" reconstruction paradigm, we first estimate the projective reconstruction that is finally upgraded to a metric reconstruction using the auto-calibration methods.

Projective reconstruction is carried out through a factorization based method presented in [23] that was later enhanced by [16]. The projection of a 3D point on the image plane is related as $\lambda_j^i \mathbf{x}_j^i = \mathbf{P}^i \mathbf{X}_j$, where λ_j^i is a constant scale factor. For a given set of n points seen in m different views the projection equations can be written in the matrix form as:

$$\mathbf{W} = \mathbf{P}_{proj}\mathbf{X} \tag{1}$$

where, the $\mathbf{W}$ is the measurement matrix denoted by

$$\mathbf{W} = \begin{bmatrix} \lambda_1^1\mathbf{x}_1^1 & \lambda_2^1\mathbf{x}_2^1 & \cdots & \lambda_n^1\mathbf{x}_n^1 \\ \lambda_1^2\mathbf{x}_1^2 & \lambda_2^2\mathbf{x}_2^2 & \cdots & \lambda_n^2\mathbf{x}_n^2 \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_n^m\mathbf{x}_n^m & \lambda_n^m\mathbf{x}_n^m & \cdots & \lambda_n^m\mathbf{x}_n^m \end{bmatrix} \tag{2}$$

and $\mathbf{P}_{proj}$ and $\mathbf{X}$ are the matrices formed by stacking the camera matrices $\mathbf{P}^i$ and 3D points $\mathbf{X}_j$

$$\mathbf{P}_{proj}\mathbf{X} = \begin{bmatrix} \mathbf{P}^1 \\ \mathbf{P}^2 \\ \vdots \\ \mathbf{P}^m \end{bmatrix} \begin{bmatrix} \mathbf{X}_1 & \mathbf{X}_2 & \cdots & \mathbf{X}_n \end{bmatrix} \tag{3}$$

If the depths λ_j^i are known, then camera matrices $\mathbf{P}_{proj}$ and the 3D points $\mathbf{X}$ can be estimated by decomposing the closest (in Frobenius norm) rank 4 approximation of the measurement matrix $\mathbf{W}$. Thus, if $\mathbf{W} = \mathbf{U}\mathbf{D}\mathbf{V}^T$, all but first four diagonal entries of $\mathbf{D}$ are set to zero resulting in $\hat{\mathbf{D}}$, then $\mathbf{P}_{proj} = \mathbf{U}\hat{\mathbf{D}}$ and $\mathbf{X} = \mathbf{V}^T$. It should be noted here that this factorization is not unique due to a projective ambiguity. We iterate over this process of factorization starting from an initial estimate of $\lambda_j^i = 1$ and re-projecting the points into each image to obtain the new estimates of depths at each iteration. During the entire process the image data is normalized using the isotropic scaling method of [9] and at each iteration the depths are re-normalized such that the values of λ_j^i are as close to unity as possible.

It is often advantageous to refine the projective reconstruction by doing a nonlinear bundle adjustment that seeks a Maximum-Likelihood (ML) solution to the projection equations $\mathbf{x}_j^i = \mathbf{P}^i\mathbf{X}_j$ in the presence of (Gaussian) noise. The best estimate of the projection matrices $\hat{\mathbf{P}}$ and the 3D points $\hat{\mathbf{X}}$ that minimizes the (geometric) re-projection error in image points is obtained using the non-linear Nelder-Mead simplex optimization [17].

The projective reconstruction $\{\mathbf{P}_{proj}^i, \mathbf{X}_j\}$ is upgraded to a metric reconstruction using the auto-calibration technique of [20]. Auto-calibration methods can simultaneously estimate the internal camera parameters while learning the metric scene structure, hence, the cameras are not required to be calibrated in advance. It exploits the properties of the absolute dual quadric, $\mathbf{Q}_\infty^*$, which is a degenerate dual (i.e. plane) quadric represented by a 4x4 homogeneous matrix of rank 3. It encodes both the plane at infinity (π) and the absolute conic (Ω) in a concise fashion. Hence if we can estimate the absolute dual quadric, then we can find both π and Ω, which can be used to upgrade the projective reconstruction to affine and then metric reconstruction. The tricky part is to estimate $\mathbf{Q}_\infty^*$.

The projection of $\mathbf{Q}_\infty^*$ (through the camera matrix $\mathbf{P}$) is the dual image of the absolution conic ω^*

$$\omega^* = \mathbf{P}\mathbf{Q}_\infty^*\mathbf{P}^T = \mathbf{K}\mathbf{K}^T \tag{4}$$

where $\mathbf{K}$ is the matrix containing the internal camera parameters. The idea of auto-calibration based on $\mathbf{Q}_\infty^*$ is to transfer the constraints on ω^* via the (known) camera matrix $\mathbf{P}^i$. The symmetric matrix $\mathbf{Q}_\infty^*$ can be parameterized by 10 homogeneous parameters, including the 4 diagonal and 6 above-diagonal entries and can be solved using a linear system of equations. Pollefeys et. al [19] suggest solving a set of uncertainty weighted linear equations imposing the zero skew constraints to estimate $\mathbf{Q}_\infty^*$. Once $\mathbf{Q}_\infty^*$ has been estimated, then the rectifying homography H can be determined by its eigenvalue decomposition as $\mathbf{Q}_\infty^* = H\tilde{I}H^T$, where

$$\tilde{I} = \begin{bmatrix} I_{3\times3} & \mathbf{0} \\ \mathbf{0} & 0 \end{bmatrix} \tag{5}$$

The projective reconstruction can be upgraded to the metric reconstruction $\{\mathbf{P}_{metric} = \mathbf{P}_{proj}^i H, H^{-1}\mathbf{X}_j\}$ using the rectifying homography. As a final step, we improve the metric reconstruction by doing a non-linear bundle adjustment. The camera projection matrices, $\mathbf{P}_{metric}^i$, can be factored into the camera internal matrix $\mathbf{K}_i$, rotation, $\mathbf{R}_i$, and translation, $\mathbf{t}_i$, such that $\mathbf{P}_{metric}^i = \mathbf{K}_i[\mathbf{R}_i|\mathbf{t}_i]$. Finally, we transform the world coordinate system to align with the coordinate system of camera 1.

3.3 Sagittal Plane Estimation

Once the relative positions of all cameras are estimated in a metric coordinate frame we can proceed with the detection of the sagittal plane. First, the 3D locations of the ground plane points ($\mathbf{X}_{gp}$) are estimated from known correspondences using a non-linear triangulation method. This non-linear search is bootstrapped from the result of a linear triangulation method that minimizes the total (squared) projection error. We then fit a plane, $[\mathbf{n}_{gp}, d_{gp}]$, to the points $\mathbf{X}_{gp}$ such that the sum of errors $(\mathbf{n}_{gp}.\mathbf{X} - d_{gp})^2$ is minimized $\forall \mathbf{X} \in \mathbf{X}_{gp}$. Here, $\mathbf{n}_{gp}$ is the normal to the ground plane and d_{gp} is the distance of this plane from origin.

We also require the information about the direction of motion of the object of interest. This can be estimated by tracking any feature on the object that is visible in all views. Surface features are not always visible across different views due to self occlusion, hence tracking the object centroid or ground contact such as feet is an alternative. Tracking feet is generally not a good idea in a crowded scenario since they are occluded most often than not. Object centroid can also be noisy due to segmentation errors. Instead, we believe that the top of the head, often referred to as a *frontier* point, can be used as a good view invariant reference point. A head detection algorithm can be used to obtain this information.

To proceed with the algorithm description, we assume that the head point correspondences of the object are known in all views. In a similar way as done earlier, we triangulate these image correspondences to estimate their corresponding 3D location. We add this 3D location to the set, $\mathbf{X}_{sp}$, containing 3D head locations from previous time frames. $\mathbf{X}_{sp}$ should contain at least one frame history, and is continuously updated to prune old 3D locations. We then estimate the best fitting 3D vector $\mathbf{l}_{sp}$ (minimum least squared error sense) to the points in $\mathbf{X}_{sp}$.

It is fair to assume that the orientation of a translating object is normal to the ground plane due to gravity constraints. Now, since we have already calculated the normal to the ground plane ($\mathbf{n}_{gp}$) and a vector lying in the sagittal plane ($\mathbf{l}_{sp}$) , we can calculate the equation of the sagittal plane, $[\mathbf{n}_{sp}, d_{sp}]$, as

$$\begin{aligned} \mathbf{n}_{sp} &= \mathbf{n}_{gp} \times \mathbf{l}_{sp} \\ d_{sp} &= \mathbf{n}_{sp} \cdot \bar{\mathbf{X}}_{sp} \end{aligned} \tag{6}$$

where, $\times$ and $\cdot$ represent the vector cross and dot products respectively, and $\bar{\mathbf{X}}_{sp}$ is the centroid of 3D head locations.

Furthermore, it should be noted that the object is assumed to move forward or backward, i.e., the motion direction is along the sagittal plane. However, if the object was moving laterally, like from left to right, then everything will still hold, except that the sagittal plane will be replaced by the coronal plane (see fig. 1). Any other arbitrary motion can be handled by breaking it into small linear motion segments along the sagittal or coronal planes.

3.4 Side View Generation

After identifying the scene geometry and orientation of the sagittal plane, we place a virtual camera looking directly at the sagittal plane so that its output corresponds to the side view of the translating object. We want to place this virtual camera C_v such that its viewing direction and up vector are aligned to $\mathbf{n}_{sp}$ and $\mathbf{n}_{gp}$ respectively. The distance of this camera from the object (d_v) can be adjusted as per requirement, and we choose it to be equal to half the average distance of the object from all cameras. This will enable us to get a close-up look at the object.

The virtual camera view can be specified in terms of its rotation, $\mathbf{R}_v$, and translation, $\mathbf{t}_v$, with respect to (say) the first real camera C_1. Let $\hat{\mathbf{z}} = (0,0,1)^T$ be the optical axis of the virtual camera, then the rotation from $\mathbf{n}_{sp}$ to $\hat{\mathbf{z}}$ is a rotation about the axis $\omega = \mathbf{n}_{sp} \times \hat{\mathbf{z}}$ with a rotation angle $\theta = cos^{-1}(\mathbf{n}_{sp} \cdot \hat{\mathbf{z}})$. The resulting rotation matrix is $\mathbf{R}_{v1} = e^{[\omega]_\times \theta}$, where $[\omega]_\times$ is the anti-symmetric matrix such that for any vector $\mathbf{v}$, $[\omega]_\times \mathbf{v} = \omega \times \mathbf{v}$. If the original up direction of the camera was $\hat{\mathbf{u}}_d = (0,1,0)^T$, then the new up-vector becomes $\hat{\mathbf{u}}_v = \mathbf{R}_{v1}^T \hat{\mathbf{u}}_d$. To finally align the up vector with $\mathbf{n}_{gp}$, we correct the rotation by $\mathbf{R}_{v2} = e^{[\mathbf{n}_{sp}]_\times \delta}$, where $\delta = cos^{-1}(\mathbf{n}_{gp} \cdot \hat{\mathbf{u}}_v)$. Hence the final rotation for the virtual camera is given by $\mathbf{R}_v = \mathbf{R}_{v1}\mathbf{R}_{v2}$. The translation can therefore be calculated as $\mathbf{t}_v = -\mathbf{R}_v(\bar{\mathbf{X}}_{sp} + d_v \mathbf{n}_{sp})$.

Now, there still remains one question to be answered. How are the novel views generated? This is one of the novel contributions of our method that is based on the planar homography constraints. It is well known that a planar homography is defined with respect to a particular plane in the world. An interesting effect of this is the parallax resulted from warping any structure not lying on the reference plane from one view to another. The homography will consistently warp all 3D points lying on the reference plane but will induce a parallax for other points. We exploit this principle by identifying the fact that homographies learnt with respect to the sagittal plane will warp the projection of points on the object consistently and inducing a parallax for everything else. Hence, if we can warp the multiple camera views via this planar homography, then the object shape will reinforce itself at the same location while other objects (whose sagittal plane do not coincide with the one already identified) or noise will map to different locations. Selecting the virtual camera as the reference view will hence warp only the side view projection of the object consistently from all other camera views, which can finally be combined in a fusion framework.

To warp any point $\mathbf{x}_i$ in the i-th camera view to its corresponding location $\mathbf{x}_v$ in the virtual view we need to calculate the corresponding planar homography $\mathbf{H}_{vi}$, such that $\mathbf{x}_v = \mathbf{H}_{vi}\mathbf{x}_i$. The homography from camera C_1 to the virtual camera C_v with respect to the sagittal plane is given by

$$\mathbf{H}_{v1} = \mathbf{K}_v \left(\mathbf{R}_v + \frac{\mathbf{t}_v \mathbf{n}_{sp}}{d_{sp}} \right) \mathbf{K}_1^{-1} \tag{7}$$

where we assume that $\mathbf{K}_v = \mathbf{K}_1$. For any other camera ($C_j$, $j \in \{2..m\}$) whose rotation ($\mathbf{R}_j$) and translation ($\mathbf{t}_j$) with respect to C_1 are known, the homography can be constructed by composition as follows

$$\begin{aligned} \mathbf{H}_{1j} &= \mathbf{K}_1 \left(\mathbf{R}_j + \frac{\mathbf{t}_j \mathbf{n}_{sp}}{d_{sp}} \right) \mathbf{K}_j^{-1} \\ \mathbf{H}_{vj} &= \mathbf{H}_{v1}\mathbf{H}_{1j} \end{aligned} \tag{8}$$

3.5 Evidence Fusion

Since the cameras are static, we first estimate a simple median background ($\mathcal{B}_i$, $i \in \{1 \cdots m\}$) from the video sequence in each of the m original view. For each new frame ($\mathcal{I}_i$) we proceed by doing a simple background subtraction to obtain the foreground mask $\mathcal{F}_i = |\mathcal{I}_i - \mathcal{B}_i|$, where $|\cdot|$ is the absolute value operator. For each pixel, we normalize $\mathcal{F}_i$

by $\mathcal{N}_i = max(\mathcal{B}_i, 255 - \mathcal{B}_i)$ to obtain the foreground probabilities $\mathcal{P}_i = \mathcal{F}_i / \mathcal{N}_i$. The foreground probability images, $\mathcal{P}_i$, from the original camera views are warped to the novel side view, $\mathcal{W}_i$, using the sagittal plane homographies, $\mathbf{H}_{vi}$.

Before we combine the evidence from all the views, we should make this careful observation. For a given object pose, if an original camera view is aligned with the virtual camera view then it should be given more weight during fusion as opposed to another camera that is, say, looking in the orthogonal direction. In other words, a camera looking from behind or from front of the object will not contribute any information regarding the side view as opposed to the camera whose viewing direction is similar to the side viewing virtual camera. Due to this observation, we propose a weighted fusion method, where each camera is assigned an importance weight based on the square of the cosine of the angle between its viewing direction and that of the virtual camera's viewing direction. Let these weights be denoted by α_i such that

$$\sum_{i=1}^{m} \alpha_i = \sum_{i=1}^{m} \left(\mathbf{n}_{sp} \cdot \mathbf{R}_i^T \hat{\mathbf{z}} \right)^2 = 1 \tag{9}$$

We investigated some traditional approaches to evidence fusion of warped foreground maps, including, simple operators like *and*, *or*, *sum*, *product*, *arithmetic* and *geometric* mean, etc., and combinations thereof. We found that the *product* based rules are not useful to deal with situations containing occlusions, since an occlusion in single image (area) can drive the entire product to a low value after fusion. For this reason we decided to use the following addition based exponentially weighted fusion rule

$$\mathcal{E} = \sum_{i=1}^{m} [1 - (1 - \mathcal{W}_i)^{\alpha_i}] \tag{10}$$

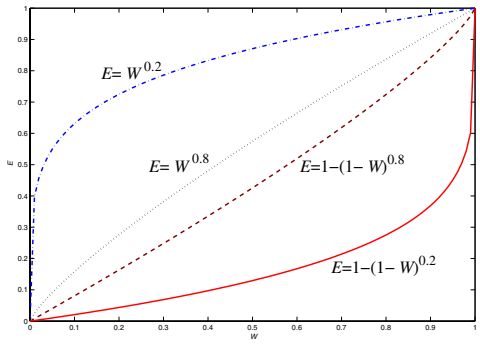

Figure 3: Transformation of sensor detection probabilities (W) for two exponentially weighted fusion functions (a) $\mathcal{E} = \mathcal{W}^{\alpha}$ and (b) $\mathcal{E} = 1 - (1 - \mathcal{W})^{\alpha}$

The fusion rule in eqn. 10 is more accurate as opposed to the obvious combination, $\sum \mathcal{W}_i^{\alpha_i}$, since the latter incorrectly amplifies the detection probabilities for sensors having low weights. As seen from fig. 3, the regular fusion function, $\mathcal{W}^{\alpha}$, and our proposed rule, $1 - (1 - \mathcal{W})^{\alpha}$, both transform the sensor detection probabilities (approximately) linearly for reliable sensors (e.g. $\alpha = 0.8$). But, in the case of unreliable sensors (e.g. $\alpha = 0.2$), the regular exponentiation rule believes more in the probability measurements of the sensor as opposed to our transformed rule that correctly suppresses the unreliable sensor's measurements. This effect is more drastic for low-probability detection measurements from the unreliable sensors.

Fusing the warped foreground probability images, $\mathcal{W}_i$, as per eqn. 10 results in a pseudo-probability image, which we refer to as the "*shape energy map*". Due to the homography constraints, the regions belonging to object shape are warped to the same location resulting in high energy regions. We desire a binary shape silhouette as the final result of the process, which can be obtained by thresholding the *shape energy map*. This threshold can be user defined but we have found that Otsu's thresholding method produces desirable results for most cases.

4. EXPERIMENTS

In this section we demonstrate the proposed framework by presenting results of the experiments conducted on various indoor and outdoor sequences.

For all the experiments, we need to determine the relative orientation of the cameras in the scene using the method described in Sect. 3.2. In all cases cameras are assumed to be wide baseline, so an automatic matching approach like that of [26] cannot be used to recover the point correspondences. One could implement a more sophisticated matching algorithm similar to [25], but we chose to manually select corresponding points from the scene to avoid the propagation of calibration errors into our final results.

Furthermore, we would like to mention that it is expected in a surveillance scenario that the cameras are placed such that they observe the monitored area from various different vantage points. The method will not magically produce the desired canonical view if no information about it is present. For example, if only two cameras are placed at 180 degrees from each other and the object is moving along the baseline then the side view generated by this algorithm will not be accurate since there is no evidence for the same. On contrary, if 4 cameras are placed in say 4 corners of the surveillance area than at a given time the worst views will still be only 45 degrees off the canonical view and fusion step will combine the complimentary information to produce a reasonable output for the desired view.

4.1 Indoor

The indoor configuration consists of four cameras looking at a human shaped toy ("*mini-man*") from roughly the four corners of the room. We attempt to simulate a surveillance like scenario where the cameras are looking down at objects moving on a ground plane (fig. 4). *Mini-man* is an interesting object for this experiment as it can give a qualitative assessment of the results based on the extracted shape its body parts. We attached a pointer on to the toys head to enable the identification of longitudinal (back-to-front) motion direction.

To demonstrate that our method is useful in cases where the object is unrestricted to move in any direction, we captured four sequences (A, B, C, and D), each representing a different orientation of *mini-man*. Fig. 6 shows the view from camera 4 for sequences B, C, and D.

Fig. 5 shows the original foreground probabilities in each view (top row), which are warped to the novel canonical view

Figure 4: Four different cameras views for the indoor sequence A (Camera 1 to 4 from left to right)

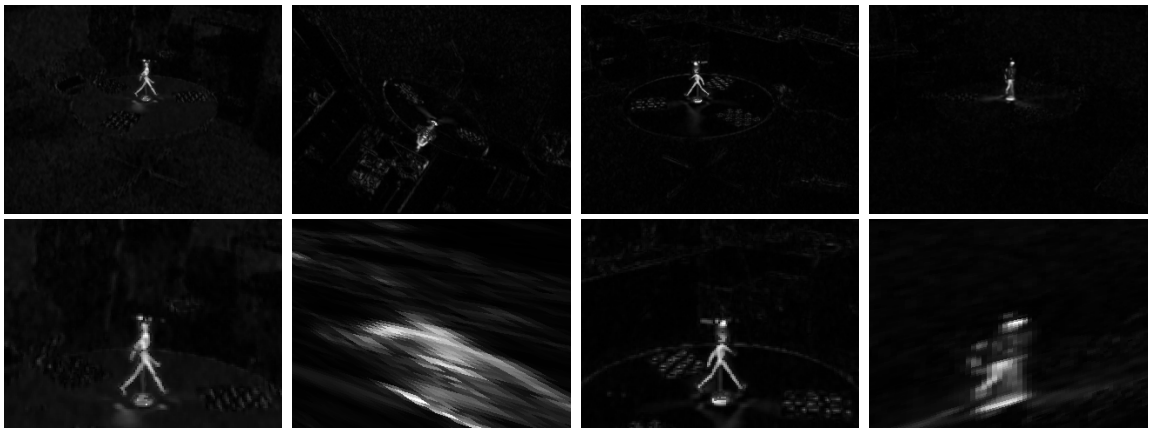

Figure 5: Intermediate results for processing of indoor sequence D. Original probability maps (top row) from each camera view are warped to the canonical view (bottom row) through planar homographies

(bottom row) through the appropriate planar homographies. It can be seen from these images how the object is centered, magnified, and warped in each image. These results are then combined according to eqn. 10 to produce the *shape energy map* shown in fig. 7. The final binary results are obtained by (Otsu's) thresholding the *shape energy map* and are shown in fig. 8(a)-(d) for sequences A, B, C, and D, respectively. One can also imagine doing a weighted combination of color information from the original images to obtain a textured output as shown in 8(e). Obviously, this may not represent the true object texture.

Finally, table 1 presents the (%) reliability of each camera view in contributing information for side-view generation for different indoor sequences. The method automatically determines the importance of each original view and accordingly combines information to produce the side view. For instance, camera view 2 in indoor sequence D (fig. 5) is looking at *mini-man* from behind and, hence, has almost no information to contribute to the process of side-view generation. It is also noteworthy how the algorithm is inherently able to deal with arbitrary camera angles, such as that of camera 2 in the indoor sequences.

4.1.1 Robustness to Occlusions

One can argue that after calculating the reliability for each original view, we can select the "best" camera view and generate the output by warping it to the canonical view. This is a simpler approach that may work in certain cases, but certainly not in presence of occlusions. The weighted fusion technique to combine information from different views is preferred over this method. Nevertheless, our method is still applicable to scenarios that require to identify only the "best" camera views, e.g., to collect face mug-shots etc.

Figure 6: View from camera 4 for different poses of *mini-man*

To demonstrate the robustness of our approach to occlusions in different camera views we manually occluded regions in each image (in fig. 9). As seen in fig. 10, the proposed method gracefully fills the missing information in each view and successfully generates the complete side-view of *mini-man*.

Figure 7: *Shape energy map*: **Result of fusing information from warped camera views as per eqn. 10 for indoor sequence D**

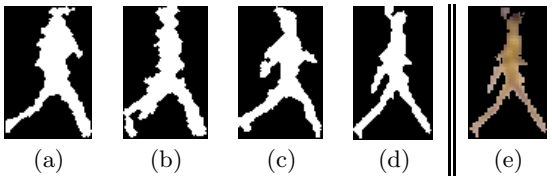

Figure 8: (a)-(d) Binary results for indoor sequences A-D and (e) Result of weighted color fusion for sequence D

Figure 9: Manually created occlusions in the four camera views for an indoor sequence

Figure 10: Resulting side view after occlusion reasoning as a result of weighted fusion

	Cam 1	Cam 2	Cam 3	Cam 4
Seq A	29	29	32	10
Seq B	03	51	05	41
Seq C	32	10	28	30
Seq D	48	00	46	06

Table 1: *Percentage* **reliability of each camera view for different indoor sequences**

4.2 Outdoor

We tested our system on a real outdoor sequence of a surveillance area monitored by 3 cameras. Two of these cameras were color and the third was grayscale. Representative views from these cameras can be seen in fig. 11.

Figure 11: Camera views from the outdoor surveillance cameras

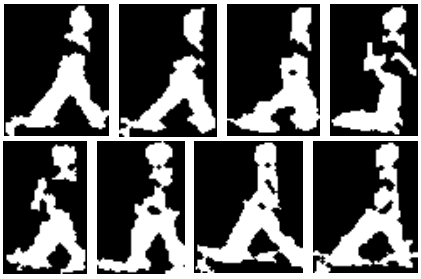

Figure 12: Resulting binary silhouettes (various stance) for some frames of the outdoor sequence

Most of these cameras are placed on top of tall buildings thus the objects of interest (people) are typically imaged at low resolutions, e.g., 15×30. We employ simple background subtraction techniques (Sect. 3.5) and manually click on head points for the purpose of this experiment. One can implement more sophisticated methods of foreground extraction [15, 22] and head detection [27], if desired.

We present the final binary outputs from a few frames of the sequence in fig. 12. Notice how well the shape of the legs is captured in these silhouettes as it would have been seen in the side-view of the action. Moreover, the silhouettes are

bigger in size since the virtual camera is placed closer to the person. We were able to eliminate most of the strong shadows (due to parallax), but some artifacts due to unremoved shadows and poor background subtraction can be seen in the results. Improved pre-processing techniques will definitely enhance the overall quality of the generated silhouettes. Furthermore, as pointed out earlier, most of these surveillance cameras are placed at very high locations, capturing more of a top-down view, and hence their contribution towards side-view information is reduced as compared to a camera placed at the eye-level. The effectiveness of the method is expected to scale up with the availability of more complementing camera views.

5. SUMMARY AND CONCLUSION

We presented a novel method for generating canonical views of any moving object observed by multiple cameras. After identifying relative camera orientations and ground plane correspondences, we use simple geometric constraints to identify the reference sagittal plane passing through the object. The desired canonical view is selected with respect to the sagittal plane and a virtual camera is positioned accordingly. Next, we define a set of planar homographies with respect to the reference plane that maps the points in original camera views to the virtual camera view. These homographies ensure that object projections on the reference plane are warped consistently from all the views, whereas, other projections will suffer from parallax. Thus, evidence is gathered from different views that are fused together to produce the *shape energy map*. This map is finally thresholded to obtain a binary shape silhouette of the object in the desired canonical pose.

Experimental results are presented for various sequences demonstrating the applicability of our algorithm in extracting side-views for objects oriented in arbitrary directions. The approach is found to be robustness in dealing with occlusions. This is a work in progress and we are currently striving to integrate it with other applications such as action recognition. We also plan to obtain more quantitative results regarding the performance of this algorithm with different number of cameras and their relative placements.

6. REFERENCES

[1] S. Avidan and A. Shashua. Novel view synthesis by cascading trilinear tensors. *IEEE Trans. Visualization and Computer Graphics*, 4(4):293–306, 1998.

[2] A. R. Chowdhury, A. Kale, and R. Chellappa. Video synthesis of arbitrary views for approximately planar scenes. In *Proc. Int. Conf. Acoustics, Speech, and Signal Process.*, volume 3, pages 497–500, April 2003.

[3] R. Collins, R. Gross, and J. Shi. Silhouette-based human identification from body shape and gait. In *Proc. Int. Conf. on Auto. Face and Gesture Recognition*, 2002.

[4] J. Davis and A. Bobick. The representation and recognition of action using temporal templates. In *Proc. Comp. Vis. and Pattern Rec.*, pages 928–934. IEEE, 1997.

[5] J. Davis and A. Tyagi. A reliable-inference framework for recognition of human actions. In *Advanced Video and Signal Based Surveillance*, pages 169–176. IEEE, 2003.

[6] J. Davis and A. Tyagi. Minimal-latency human action recognition using reliable-inference. *Image and Vision Computing*, 24(5):455–472, May 2006.

[7] T. Denton, M. F. Demirci, J. Abrahamson, A. Shokoufandeh, and S. Dickinson. Selecting canonical views for view-based 3-d object recognition. In *Proc. Int. Conf. Pat. Rec.*, pages 273–276, 2004.

[8] A. Habed and B. Boufama. Novel view synthesis: a comparative analysis study. In *Vision Interface*, pages 217–224, 2000.

[9] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[10] P. Huang, C. Harris, and M. Nixon. Recognising humans by gait via parametric canonical space. *Artif. Intell. in Eng.*, 13:359–366, 1999.

[11] T. Huang and A. Netravali. Motion and structure from feature correspondences: A review. In *Proc. IEEE*, volume 82, pages 252–268, Feb 1994.

[12] T. Jebara, A. Azarbeyejani, and A. Pentland. 3D structure from 2D motion. *IEEE Signal Processing Magazine*, 16(3), 1999.

[13] K. Jeong and C. Jaynes. Moving shadow detection using a combined geometric and color classification approach. In *Wkshp. on Motion and Video Computing*, Jan 2004.

[14] S. M. Khan and M. Shah. A multiview approach to tracking people in crowded scenes using a planar homography constraint. In *Proc. European Conf. Comp. Vis.*, 2006.

[15] K. Kim, T. H. Chalidabhongse, D. Harwood, and L. Davis. Real-time foreground-background segmentation using codebook model. *Elsevier Real-Time Imaging*, 11(3):172–185, June 2005.

[16] S. Mahamud, M. Hebert, Y. Omori, and J. Ponce. Provably-convergent iterative methods for projective structure from motion. In *Proc. Comp. Vis. and Pattern Rec.*, 2001.

[17] J. A. Nelder and R. Mead. A simplex method for function minimization. *Comput. J.*, pages 308–313, 1965.

[18] V. Parameswaran and R. Chellappa. View invariants for human action recognition. In *Proc. Comp. Vis. and Pattern Rec.*, pages 613–619, 2003.

[19] M. Pollefeys, L. V. Gool, M. Vergauwen, F. Verbiest, K. Cornelis, J. Tops, and R. Koch. Visual modeling with a hand-held camera. *Int. J. of Comp. Vis.*, 59(3):207–232, 2004.

[20] M. Pollefeys, R. Koch, and L. V. Gool. Self calibration and metric reconstruction in spite of varying and unknown internal camera parameters. In *Proc. Int. Conf. Comp. Vis.*, pages 90–96, 1998.

[21] C. Rao and M. Shah. A view-invariant representation and learning of human action. In *Proc. Wkshp. on Detection and Recognition of Events in Video*, pages 55–63. IEEE, 2001.

[22] C. Stauffer and W. Grimson. Adaptive background mixture models for real-time tracking. In *Proc. Comp. Vis. and Pattern Rec.*, pages 246–252. IEEE, 1999.

[23] P. Sturm and W. Triggs. A factorization based algorithm for multi-image projective structure and

motion. In *Proc. European Conf. Comp. Vis.*, pages 709–720, 1996.

[24] R. Szeliski. Rapid octree construction from image sequences. *CVGIP: Image Understanding*, 58(1):23–32, July 1993.

[25] M. Vergauwen, F. Verbiest, V. Ferrari, C. Strecha, and L. van Gool. Wide-baseline 3D reconstruction from digital stills. In *Int. Wkshp. on Visualization and Animation of Reality-based 3D Models*, Engadin, Switzerland, Feb 2003.

[26] Z. Zhang, R. Deriche, O. D. Faugeras, and Q.-T. Luong. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial Intelligence*, 78(1-2):87–119, 1995.

[27] T. Zhao and R. Nevatia. Tracking multiple humans in complex situations. *IEEE Trans. Patt. Analy. and Mach. Intell.*, 26(9):1208–1221, Sept. 2004.