MAP Visibility Estimation for Large-Scale Dynamic 3D Reconstruction*

Hanbyul Joo   Hyun Soo Park   Yaser Sheikh

Carnegie Mellon University

{hanbyulj,hyunsoop,yaser}@cs.cmu.edu

Abstract

Many traditional challenges in reconstructing 3D motion, such as matching across wide baselines and handling occlusion, reduce in significance as the number of unique viewpoints increases. However, to obtain this benefit, a new challenge arises: estimating precisely which cameras observe which points at each instant in time. We present a maximum a posteriori (MAP) estimate of the time-varying visibility of the target points to reconstruct the 3D motion of an event from a large number of cameras. Our algorithm takes, as input, camera poses and image sequences, and outputs the time-varying set of the cameras in which a target patch is visibile and its reconstructed trajectory. We model visibility estimation as a MAP estimate by incorporating various cues including photometric consistency, motion consistency, and geometric consistency, in conjunction with a prior that rewards consistent visibilities in proximal cameras. An optimal estimate of visibility is obtained by finding the minimum cut of a capacitated graph over cameras. We demonstrate that our method estimates visibility with greater accuracy, and increases tracking performance producing longer trajectories, at more locations, and at higher accuracies than methods that ignore visibility or use photometric consistency alone.

1. Introduction

Thousands of images exist for most significant landmarks around the world. The availability of such imagery has facilitated the development of large-scale 3D reconstruction algorithms, which fully leverage the number of views to produce dense and accurate 3D point clouds [16, 13, 8]. Increasingly, landmark *events* are also being captured at scale by hundreds of cameras at major sports games, concerts, and political rallies. However, analogous large-scale reconstruction algorithms, that are able to fully leverage a large number of views of an event to produce long, dense, and accurate 3D trajectories, do not yet exist.

Such video-based 3D motion reconstruction is challenging, as natural motion produces a greater occurrence of measurement loss due to occlusion and also causes artifacts in imagery (e.g., motion blur and texture deformation). Utilizing a large number of cameras can address these challenges, because it is likely to (1) narrow the average baseline between nearby cameras, (2) reduce the occurrence of occlusion, and (3) provide robustness to measurement noise due to the surplus views. However, previous approaches are unable to fully leverage the increasing number of views to improve 3D tracking performance (in terms of the average length of reconstructed trajectories, the density of the trajectories, and the accuracy of localization). The principal cause of failure emerges from errors in reasoning about the time-varying *visibility* of dynamic 3D points. Poor visibility reasoning severely affects tracking performance, as an algorithm cannot benefit from an alternate viewpoint if it is unaware that the point is visible in the alternate view. Furthermore, an erroneous conclusion that a point is visible in a camera can bias the reconstruction, often producing a characteristic "jump" artifact where a point assumes the identity of a different location.

In this paper, we demonstrate that precise inference of point visibility allows reconstruction algorithms to fully leverage large numbers of views to produce longer 3D trajectories with higher accuracy. In particular, our core algorithmic contributions are: (1) the use of motion consistency as a cue for the visibility of moving points; (2) the use of viewpoint regularity as a prior and a measure for viewpoint proximity; and (3) a maximum a posteriori (MAP) estimate for visibility estimation by probabilistically incorporating these cues with photometric and geometric consistency. We report empirical performance in reconstructing 3D motion captured by 480 cameras in scenes that contains significant occlusion, large displacement, and changes in the topology of the scene.

*http://www.cs.cmu.edu/~hanbyulj/14/visibility.html

2. Related Work

Dynamic 3D reconstruction approaches can be broadly categorized in methods that use silhouettes for reconstruction (e.g., [5, 6, 18, 24, 3]) and methods that use correspondence for reconstruction (e.g., [22, 4, 7]. Silhouette-based approaches typically use visual hulls to produce highly dense reconstruction, but require subsequent processing to estimate 3D trajectories [5, 18]. Surface matching algorithms are used to provide dense correspondences between consecutive frames [20, 17, 21]. In these approaches, mesh models in each frame are independently generated using shape-from-silhouette techniques, and sparse matching between key mesh vertexes are performed using various cues such as shape and appearance features. Dense matching is then carried out based on the sparse matches using a regularized cost function based on geodesic distance. The accuracy of motion estimation depends highly on the initial surface and texture, and is limited by the vertex resolution. Silhouette-based methods also require stationary cameras to be able to estimate accurate silhouettes.

In comparison to silhouette-based reconstruction approaches, correspondence-based methods produce sparser reconstructions, but do not require stationary cameras and can directly produce 3D trajectories. Among correspondence-based methods, perhaps the most related approaches are scene flow reconstruction methods, introduced by Vedula et al. [22]. Independently estimated 2D optical flow from multiple calibrated cameras was triangulated to generate the 3D flow, assuming that visibility was given a priori via reconstructed object shape. Several subsequent algorithms also have been proposed to recover both shape (depth) and motion simultaneously [1, 23, 11]. The basic assumption in these approaches is brightness constancy (or photometric consistency), which is used to determine the correspondences across views; spatial regularization is used to condition the optimization and reduce the noise. While these approaches represent the target as a 3D point, other approaches use richer 3D representation such as dynamic surfels [4, 7] or meshes [9]. Mesh-based approaches have demonstrated robust results, producing trajectories of longer duration, but at the cost of assuming a fixed topology with a known mesh, and through the use of regularization.

Typically, in previous work, only a small number of cameras are considered. In scene flow approaches, stereo cameras are usually used, and other approaches also use at most 10 to 20 cameras (17 by Vedula et al. [22], 22 by Furukawa and Ponce [9], 8 by Huguet and Devernay [11], 7 by Carceroni and Kutalakos [4]). At this scale, information loss due to motion blur, texture deformation, occlusion, and self-occlusion are severe, and therefore necessitate significant spatiotemporal regularization of reconstructions. In most algorithms, precise camera visibility information is not considered, because the noise from a small number of outlier cameras can be ignored. Camera visibility is either assumed to be given by the 3D reconstruction algorithm [22] or handled by a robust estimator [23, 7, 1, 14]. Patch-based methods use photometric consistency to determine visibility by comparing the texture across views [4, 7, 9]. However, these approaches require the texture of the 3D patch, which depends heavily on the accuracy of the recovered patch shape.

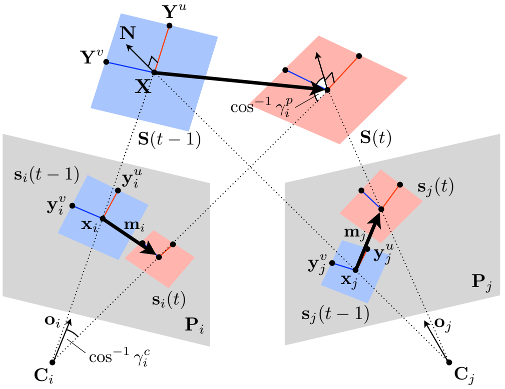


Figure 1. The motion of a patch between time $t-1$ and t is reconstructed from multiple cameras.

3. Notation

Our algorithm takes, as input, image sequences from N calibrated and synchronized cameras over F frames and produces, as output, 3D trajectories of P moving points with their instantaneous orientations and associated visibility in each camera frame. Since the method is applied to each point independently, we consider only a single point here to simplify the exposition.

As shown in Figure 1, we track a parallelogram patch centered on a target 3D point $\mathbf{X} \in \mathbb{R}^3$, whose extent is defined by two additional points $\mathbf{Y}^u$ and $\mathbf{Y}^v \in \mathbb{R}^3$. The texture information $\mathbf{Q} \in \mathbb{R}^m$ associated with the patch is defined by a unit vector concatenating normalized intensity values at a fixed number of grid positions on the patch, where m is the number positions in the grid[1]. The patch $\mathbf{S}(t)$ is denoted by the set $\{\mathbf{X}(t), \mathbf{Y}^u(t), \mathbf{Y}^v(t), \mathbf{Q}(t)\}$, which is associated with the camera visibility set $\mathbf{V}(t) = \{\mathbf{v}_1(t), \cdots, \mathbf{v}_N(t)\}$, where $\mathbf{v}_i(t)$ is a binary value representing visibility with respect to the i^{th} camera. A 3D point is projected onto the i^{th} camera associated with a 3×4 projection matrix $\mathbf{P}_i$. The projection matrix is parametrized by

[1] The texture vector $\mathbf{Q}$ is normalized as follows:

$$\mathbf{Q} = \frac{1}{\sqrt{\sum_{j=1}^{m}(Q_j - \overline{Q})^2}} \begin{bmatrix} Q_1 - \overline{Q} \\ \vdots \\ Q_m - \overline{Q} \end{bmatrix}, \qquad (1)$$

where $\overline{Q} = \sum_{j=1}^{m} Q_j / m$ and Q_j is the j^{th} intensity value of the texture.

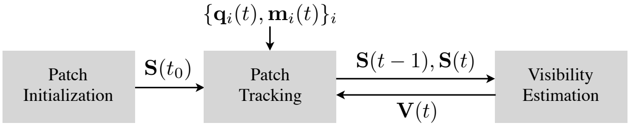


Figure 2. Overview of patch tracking and visibility estimation.

a camera center vector $\mathbf{C}_i \in \mathbb{R}^3$ and a 3×3 rotation matrix $\mathbf{R}_i \in SO(3)$. The "look-at" vector $\mathbf{o}_i$ is aligned with the z-axis of the camera, i.e., the third column of $\mathbf{R}_i^\mathsf{T}$.

The 3D patch is projected onto the camera plane to form the projected patch $\mathbf{s}_i(t) = \{\mathbf{x}_i(t), \mathbf{y}_i^u(t), \mathbf{y}_i^v(t), \mathbf{q}_i(t)\}$, where $\mathbf{x}_i(t)$, $\mathbf{y}^u(t)$, and $\mathbf{y}^v(t) \in \mathbb{R}^2$ are the projected points, i.e., $\widehat{\mathbf{x}}_i(t) \cong \mathbf{P}_i\widehat{\mathbf{X}}(t)$, $\widehat{\mathbf{y}}_i^u(t) \cong \mathbf{P}_i\widehat{\mathbf{Y}}^u(t)$, and $\widehat{\mathbf{y}}_i^v(t) \cong \mathbf{P}_i\widehat{\mathbf{Y}}^v(t)$, where $\widehat{\cdot}$ is the homogeneous coordinate representation of each vector. $\mathbf{q}_i \in \mathbb{R}^m$ is the texture information of the projected patch, which is defined by a concatenation of all the intensities from the i^{th}camera, corresponding to the projected grid positions of $\mathbf{S}$, and normalized as in Equation (1). Ideally, $\mathbf{Q} = \mathbf{q}_i$ if the 3D patch $\mathbf{S}$ is visible from the i^{th} camera, discounting illumination variation. We denote $\mathbf{m}_i$ as 2D optical flow at $\mathbf{x}_i(t-1)$ in the i^{th} camera, as shown in Figure 1.

The relationship between the i^{th} camera and patch can be defined by the co-visibility set $\mathbf{\Gamma}_i = \{\gamma_i^c, \gamma_i^p\}$, where

$$\gamma_i^c = \frac{(\mathbf{X} - \mathbf{C}_i)^\mathsf{T}\mathbf{o}_i}{||\mathbf{X} - \mathbf{C}_i||} \quad \text{and} \quad \gamma_i^p = \frac{(\mathbf{C}_i - \mathbf{X})^\mathsf{T}\mathbf{N}}{||\mathbf{C}_i - \mathbf{X}||},$$

γ_i^c encodes the angle cosine of the patch location with respect to the camera "look-at" vector $\mathbf{o}_i$ and γ_i^p encodes the angle cosine of the camera location with respect to the 3D patch normal $\mathbf{N}$.

4. Overview

At the initial time instance t_0, a target 3D patch is reconstructed and, over time, the algorithm alternately estimates the patch position and normal and its visibility with respect to all cameras. It should be noted that t_0 can be any arbitrary frame and that the tracking and the visibility computation are performed both forwards and backwards in time from t_0. We consider only forward tracking, from $t-1$ to t to simplify the description. The flow chart of our algorithm is shown in Figure 2.

Patch Initialization. Given the images from different cameras at the same time instance t_0, the algorithm reconstructs 3D points by matching features and triangulates them within a RANSAC framework. A 3D patch centered on $\mathbf{X}$ is reconstructed by maximizing the photometric consistency among the cameras where the patch is visible[2]. This initializes $\mathbf{S}(t_0)$ and $\mathbf{V}(t_0)$.

Patch Tracking. Given the previously obtained 3D patch $\mathbf{S}(t-1)$ and visibility $\mathbf{V}(t-1)$, the algorithm estimates the next 3D patch $\mathbf{S}(t)$ based on 2D optical flow in the cameras defined by $\mathbf{V}(t-1)$. For the i^{th} camera in $\mathbf{V}(t-1)$, optical flow [12] is estimated at multiple scales at the points $\mathbf{x}_i(t-1)$, $\mathbf{y}_i^u(t-1)$, and $\mathbf{y}_i^v(t-1)$. To eliminate unreliable flow, a backward-forward consistency check [19] is performed for flow at each scale and only the most reliable flow is retained. The next 3D positions, $\mathbf{X}(t)$, $\mathbf{Y}^u(t)$, and $\mathbf{Y}^v(t)$, are estimated by triangulating optical flow outputs within a RANSAC framework. The RANSAC process is crucial since $\mathbf{V}(t-1)$ may not be valid anymore at time t, due to motion. After RANSAC, the normal is refined by maximizing the photometric consistency among the images that belong to the inliers of RANSAC, as in the patch initialization process.

Visibility Estimation. Based on the reconstructed $\mathbf{S}(t)$ and its motion from $\mathbf{S}(t-1)$, our approach finds the MAP estimate of the current visibility set $\mathbf{V}(t)$ by fusing photometric consistency, motion consistency, and geometric consistency, in conjunction with a Markov Random Field (MRF) prior. Typically, the tracking process is severely affected by false positive cameras where the target is not visible. Poor visibility reasoning at the RANSAC stage can cause a characteristic "jump" error to a different scene point, and also reduces the normal refinement performance causing frequent local minima during the optimization process. Our precise visibility estimation results in longer trajectories of higher accuracy.

Patch tracking and visibility estimation are interdependent processes. At each time instance, we can iterate these two procedures until convergence; in practice, a single iteration is usually sufficient.

5. Visibility Estimation

In this section, we present a method to compute the maximum a posteriori (MAP) estimate of visibility $\mathbf{V}$ using photometric consistency, motion consistency, and geometric consistency, with a proximity prior. These cues are represented using 2D texture $\{\mathbf{q}_i\}_{i=1}^N$, 2D optical flow $\{\mathbf{m}_i\}_{i=1}^N$, and the co-visibility set $\{\mathbf{\Gamma}_i\}_{i=1}^N$. Given these cues and by applying Bayes theorem, the probability of vis-

[2] The cameras that participate in RANSAC are used as an initial visible set, and the reference camera $\mathbf{P}_{\text{ref}}$ is selected as the one closest to the initial 3D point in the inlier set. A 3D patch centered on $\mathbf{X}$ is initialized as a fixed scale square patch (40mm×40mm), with $\mathbf{N}$ parallel to $\mathbf{o}_{\text{ref}}$. We refine the patch based on the method described by Furukawa and Ponce [10] and select a new reference camera as the one closest to the current patch normal. The corresponding visibility set is updated by selecting cameras that have higher Normalized Cross Correlation (NCC) score than a threshold compared to $\mathbf{P}_{\text{ref}}$. Within the patch initialization process, the normal refinement and visibility update are iterated.

ibility is

$$P(\mathbf{V}|\mathbf{q}_1, \mathbf{m}_1, \mathbf{\Gamma}_1, \cdots, \mathbf{q}_N, \mathbf{m}_N, \mathbf{\Gamma}_N) \\ \propto P(\mathbf{q}_1, \mathbf{m}_1, \mathbf{\Gamma}_1, \cdots, \mathbf{q}_N, \mathbf{m}_N, \mathbf{\Gamma}_N|\mathbf{V})P(\mathbf{V}).$$

Given the visibility of each camera, we assume that (1) the cues in that camera are conditionally independent to the cues in other cameras and the visibility of other cameras, (2) that each cue within the same camera is conditionally independent to each other. The probability can be written as

$$\left(\prod_{i=1}^{N} P(\mathbf{q}_i|\mathbf{v}_i)P(\mathbf{m}_i|\mathbf{v}_i)P(\mathbf{\Gamma}_i|\mathbf{v}_i)\right) P(\mathbf{V}). \tag{2}$$

The MAP estimate of visibility $\mathbf{V}^*$ can be obtained by maximizing the expression in Equation (2), i.e.,

$$\mathbf{V}^* = \underset{\mathbf{V}}{\operatorname{argmax}} \left(\prod_{i=1}^{N} P(\mathbf{q}_i|\mathbf{v}_i)P(\mathbf{m}_i|\mathbf{v}_i)P(\mathbf{\Gamma}_i|\mathbf{v}_i)\right) P(\mathbf{V}),$$

or equivalently,

$$\mathbf{V}^* = \underset{\mathbf{V}}{\operatorname{argmax}} \sum_{i=1}^{N} \log P(\mathbf{q}_i|\mathbf{v}_i) + \sum_{i=1}^{N} \log P(\mathbf{m}_i|\mathbf{v}_i) + \\ + \sum_{i=1}^{N} \log P(\mathbf{\Gamma}_i|\mathbf{v}_i) + \log P(\mathbf{V}). \tag{3}$$

We describe the probability of each cue and the prior in the subsequent sub-sections, and compute the MAP estimate by finding the minimum cut of a capacitated graph over cameras [2].

5.1. Photometric consistency

Photometric consistency has been widely used for reasoning about visibility [16, 13, 8, 9, 7]. It measures the correlation between the texture $\mathbf{Q}$ of a 3D patch and the texture $\mathbf{q}_i$ of the corresponding patch in the i^{th} camera. Normalized Cross Correlation (NCC) is one such measure of photometric consistency, which is robust to illumination variation. Since $\mathbf{Q}$ and $\mathbf{q}_i$ are defined as normalized unit vectors by Equation (1), $\mathbf{Q}^{\mathsf{T}}\mathbf{q}_i$ measures the NCC. We model the probability distribution of $\mathbf{q}_i$ using a von Mises-Fisher distribution around $\mathbf{Q}$, i.e., $\mathbf{q}_i \sim \mathcal{V}(\mathbf{Q}, \kappa)$, which is defined by $\mathbf{Q}^{\mathsf{T}}\mathbf{q}_i$. κ is a concentration parameter that controls the degree of variation of the texture. Lower values of κ allows more variation between $\mathbf{Q}$ and $\mathbf{q}_i$. From the distribution, we can describe the logarithm of the probability of $\mathbf{q}_i$ given $\mathbf{v}_i$ as

$$\log P(\mathbf{q}_i|\mathbf{v}_i) \propto \kappa \mathbf{Q}^{\mathsf{T}}\mathbf{q}_i. \tag{4}$$

5.2. Motion Consistency

In dynamic scenes, motion is an informative cue for determining visibility. Given the 3D motion of a patch, the observed optical flow at the i^{th} camera must be consistent with the projected 3D motion of the target patch, if the patch is visible from the camera view. In other words, motion consistency requires that 2D optical flow $\mathbf{m}_i$ must be consistent with the projected displacement of the 3D motion $\mathbf{x}_i(t) - \mathbf{x}_i(t-1)$.

We model the probability distribution of $\mathbf{m}_i$ using a normal distribution around the projected 3D displacement, i.e., $\mathbf{m}_i \sim \mathcal{N}\left(\mathbf{x}_i(t) - \mathbf{x}_i(t-1), \sigma\right)$, where σ is the standard deviation capturing the certainty of the 3D motion estimation in pixel units. Therefore, the log likelihood can be written as

$$\log P(\mathbf{m}_i|\mathbf{v}_i) \propto -\frac{\|\mathbf{m}_i - (\mathbf{x}_i(t) - \mathbf{x}_i(t-1))\|^2}{2\sigma^2}. \tag{5}$$

Motion consistency is a necessary condition. We now characterize cases when the motion consistency cue is ambiguous. Let $\mathbf{X}(t)$ and $\mathbf{X}'(t)$ be two distinct points in 3D space. Motion consistency cue is ambiguous if and only if the following two conditions hold:

$$\begin{aligned} \mathbf{P}_i\widehat{\mathbf{X}}(t) &\cong \mathbf{P}_i\widehat{\mathbf{X}}'(t) \\ \mathbf{P}_i\widehat{\mathbf{X}}(t+1) &\cong \mathbf{P}_i\widehat{\mathbf{X}}'(t+1), \end{aligned} \tag{6}$$

where $\|\mathbf{X} - \mathbf{C}_i\| > \|\mathbf{X}' - \mathbf{C}_i\|$, i.e., $\mathbf{X}'(t)$ occludes $\mathbf{X}(t)$ for the i^{th} camera. In a static scene, motion does not exist and thus, the motion consistency cue is always ambiguous because $\mathbf{X}(t) = \mathbf{X}(t+1)$ and $\mathbf{X}'(t) = \mathbf{X}'(t+1)$. Another case that occurs in practice is when the occluding patch and the occluded patch lie on a body undergoing global translational motion, under a camera that approaches orthographic projection.

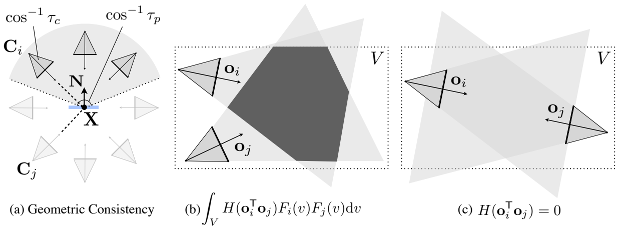
We characterize the set of ambiguous motions where Equation (6) holds, assuming that $\mathbf{X}(t)$ and $\mathbf{X}'(t)$ undergo the same affine transform between frames, as

$$\begin{aligned} \mathbf{X}(t+1) &= \mathbf{A}\mathbf{X}(t) + \mathbf{a} \\ \mathbf{X}'(t+1) &= \mathbf{A}\mathbf{X}'(t) + \mathbf{a}, \end{aligned} \tag{7}$$

where $\mathbf{A} \in \mathbb{R}^{3\times3}$ and $\mathbf{a} \in \mathbb{R}^3$ represent a 3D affine transform. The motion consistency cue is ambiguous if and only if the following condition holds:

$$\mathbf{X} \in \operatorname{null}\left([\mathbf{a}]_\times \mathbf{A}\right), \tag{8}$$

where $\operatorname{null}(\cdot)$ is the null space of $\cdot$. See the Appendix for a proof. In ideal cases with infinite precision and zero measurement noise, this condition rarely occurs (if there is motion).



(a) Geometric Consistency (b) $\int_V H(\mathbf{o}_i^\mathsf{T}\mathbf{o}_j)F_i(v)F_j(v)\mathrm{d}v$ (c) $H(\mathbf{o}_i^\mathsf{T}\mathbf{o}_j) = 0$

Figure 3. (a) The valid region filtered by γ_p and τ_p is shown as a shaded region. The angle limitation with respect to the $\mathbf{N}$ is computed as $\cos^{-1}\tau_p$. (b) g_s computed by two cameras are shown as a shaded polygon, where $\int_V \mathcal{H}(\mathbf{o}_i^\mathsf{T}\mathbf{o}_j)F_i(v)F_j(v)\mathrm{d}v > 0$. (c) An example where $\mathcal{H}(\mathbf{o}_i^\mathsf{T}\mathbf{o}_j) = 0$ is shown. An oriented patch cannot be visible by two cameras facing each other simultaneously.

5.3. Geometric consistency

Oriented patches are only visible from cameras whose "look-at" vector $\mathbf{o}_i$ is in the opposite direction to the patch normal $\mathbf{N}$ and in front of it. We incorporate this geometric cue based on the co-visibility set $\mathbf{\Gamma}_i$ considering the camera position relative to the patch normal direction and the patch position relative to the camera "look-at" vector. The probability of $\mathbf{\Gamma}_i$, given visibility $\mathbf{v}_i$, can be written as

$$P(\mathbf{\Gamma}_i|\mathbf{v}_i) = \begin{cases} \frac{1}{(1-\tau_c)(1-\tau_p)} & \text{if } \gamma_i^c \geq \tau_c, \text{and } \gamma_i^p \geq \tau_p \\ 0 & \text{otherwise,} \end{cases} \quad (9)$$

where $\tau_c < 1$ is the cosine angle representing the field of view of the camera, and $\tau_p < 1$ is a threshold (cosine angle) to determine the angular visibility with respect to the patch normal. Figure 3(a) shows an example of the cue, where the shaded area represents the valid region according to τ_p.

5.4. Visibility Regularization Prior

Under a Markov Random Field prior over camera visibility, we decompose the joint probability of visibility $P(\mathbf{V})$ into pairwise probabilities, i.e.,

$$P(\mathbf{v}_1, \cdots, \mathbf{v}_N) = \prod_{i,j\in\mathcal{G}(i)} P(\mathbf{v}_i, \mathbf{v}_j), \quad (10)$$

where $\mathcal{G}(i)$ is the set of adjacent camera indices of the i^{th} camera. This decomposition captures the prior distribution of visibility, representing the prior that two cameras that have similar viewpoints are likely to have consistent visibility. This proximity constraint constitutes prior knowledge that can regularize noisy visibility when both photometric consistency and motion consistency cues are weak (e.g., due to motion blur in an individual camera). We model the log likelihood of the joint probability as follows:

$$\log P(\mathbf{v}_1, \cdots, \mathbf{v}_N) \propto \sum_{i,j\in\mathcal{G}(i)} g_s(\mathbf{v}_i, \mathbf{v}_j), \quad (11)$$

where g_s is defined by the cost between two cameras using the overlapping volume of the two camera frustums. This is estimated as follows:

$$g_\mathrm{s}(\mathbf{P}_i, \mathbf{P}_j) = \frac{\int_V \mathcal{H}(\mathbf{o}_i^\mathsf{T}\mathbf{o}_j)F_i(v)F_j(v)\mathrm{d}v}{\int_V F_i(v) + F_j(v) - F_i(v)F_j(v)\mathrm{d}v}, \quad (12)$$

where v is an infinitesimal volume in the working space V (see Figure 3(c)). $F_i(v)$ is a binary function defined as

$$F_i(v) = \begin{cases} 1 & \text{if } v \text{ is visible from the } i^{\text{th}} \text{ camera} \\ 0 & \text{otherwise.} \end{cases} \quad (13)$$

$\mathcal{H}$ is a Heaviside step function to take into account a pair of cameras oriented in similar directions. Equation (11) captures the ratio between the volume of the intersections of camera frustums and the volume of the union of camera frustums. Figure 3(b) illustrates g_s where the shaded polygon represents $\int_V \mathcal{H}(\mathbf{o}_i^\mathsf{T}\mathbf{o}_j)F_i(v)F_j(v)\mathrm{d}v$, and Figure 3(c) shows an example where $\mathcal{H}(\mathbf{o}_i^\mathsf{T}\mathbf{o}_j) = 0$.

In practice, we discretize the working volume using voxels and count the number of common voxels that are projected inside both cameras. This enables us to reward consistent visibilities in proximal cameras.

5.5. MAP Visibility Estimation via Graph Cuts

We incorporate Equations (4), (5), (9), and (11) into Equation (3) to find the MAP estimate of visibility $\mathbf{V}^*$ and, therefore, Equation (3) can be rewritten as:

$$\mathbf{V}^* = \operatorname*{argmin}_{\mathbf{V}} \sum_{i=1}^{N} E_d(\mathbf{v}_i) + \sum_{i,j\in\mathcal{G}(i)} E_s(\mathbf{v}_i, \mathbf{v}_j), \quad (14)$$

where E_d encodes photometric consistency, motion consistency, and geometric consistency, and E_s encodes the prior between cameras.

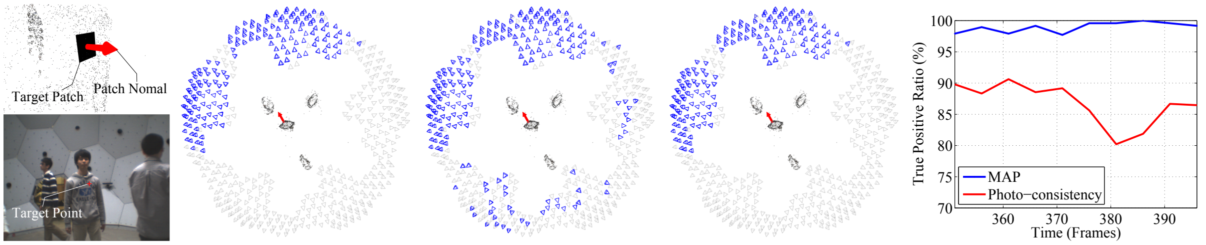
$$E_d(\mathbf{v}_i) = \frac{\|\mathbf{m}_i - (\mathbf{x}_i(t) - \mathbf{x}_i(t-1))\|^2}{2\sigma^2} - \kappa {\mathbf{Q}_i}^\mathsf{T}\mathbf{q}_i + \delta(\mathbf{\Gamma}_i)$$

$$E_s(\mathbf{v}_i, \mathbf{v}_j) = \begin{cases} 0 & \text{if } \mathbf{v}_i = \mathbf{v}_j \\ g_\mathrm{s}(\mathbf{P}_i, \mathbf{P}_j) & \text{otherwise,} \end{cases}$$

where $\delta = \log(1-\tau_c)(1-\tau_p)$ if $\gamma_i^c > \tau_c$ and $\gamma_i^p > \tau_p$, or $\delta = \infty$, otherwise. This minimization problem can be optimally computed via graph cuts [2].

6. Results

We evaluate our algorithm on a variety of challenging scenes in the presence of significant occlusion (Circular Movement and Falling Boxes), large displacement (Confetti and Fluid motion), and topological change (Falling boxes and Volleyball). Our visibility estimation enables us to better leverage a large number of cameras in producing accurate and long trajectories. The dataset used in the evaluation is summarized in Table 1 and is available on the project



(a) Selected patch (b) Ground truth (c) Photo-consistency (d) MAP (e) Visibility Accuracy Comparison

Figure 4. The red arrow denotes the normal vector of the selected patch. The pyramid structures represent camera poses, where blue cameras belong to the visible set (we warp the camera positions for better visualization). (a) The selected patch is shown in 3D view and 2D image. (b) We manually generate ground truth visibility. (c) Visibility estimated by the baseline. (d) Visibility estimated by our method. (e) We compare accuracy of visibility estimates of both methods.

website. The sequences were captured at the CMU Panoptic Studio [15] containing 480 cameras capturing 640×480 video at 25 Hz. The cameras are extrinsically and intrinsically calibrated, and are synchronized via an external clock.

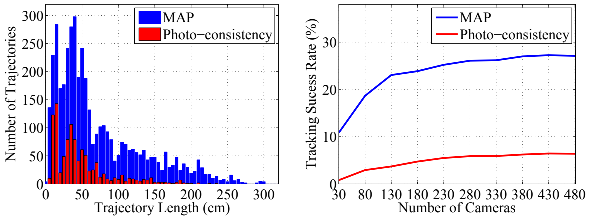
Table 1. Summary of the datasets.

Sequence	Frames	Duration	# of points	Av. traj. length
Circ. Movement	250	10.0 sec	10433	404.9 cm
Volleyball	210	8.4 sec	8422	326.4 cm
Bat Swing	200	8.0 sec	3849	224.1 cm
Falling Boxes	160	6.4 sec	17934	164.7 cm
Confetti	200	8.0 sec	10345	103.0 cm
Fluid Motion	200	8.0 sec	3153	123.1 cm

6.1. Quantitative Evaluation

Visibility Estimation Accuracy. We select an arbitrary patch in the Circular Movement sequence reconstructed at a time instance, and manually generate ground-truth visibility data at each sampled time instance by selecting cameras where the target patch is visible. We compare our visibility estimation method (MAP) against a baseline method based on photometric consistency alone, which is a cue commonly used by previous approaches [4, 7, 9]. Visibility estimation results generated from each method at a time instance are visualized in Figure 4. As a criterion, we compute the true positive detection rate between the ground truth data and $\mathbf{V}(t)$ estimated by both methods. The true positive rate from each method is shown in Figure 4(e), demonstrating that our method outperforms the baseline method by a significant margin.

Tracking Accuracy and Length. We evaluate our method considering both tracking accuracy and trajectory length. Inspired by the evaluation criterion proposed by Furukawa and Ponce [9], a test sequence is generated by appending it at the end of itself in reverse order, and the tracking algorithm is performed on the generated sequence. The tracked patches must return back to the original position, if tracking is accurate. In this experiment, the 3D error is defined by the 3D distance between initial and the final locations of the target point. We generate five test sequences using the Circular Movement sequence by changing the duration (10 to 50 frames) from a fixed initial frame. For the evaluation, we count the number of successfully reconstructed trajectories that have less than 2 cm drift error. Figure 5(a) shows a histogram of the number of trajectories using 480 cameras. Our MAP estimate method outperforms the method based on photometric consistency in terms of both number of trajectories and length of trajectories. We also perform experiments with different number of cameras by uniformly sampling cameras to examine its impact on tracking success rate. Figure 5(b) shows how our method leverages a large number of cameras. Note that the number of successfully tracked trajectories increases faster than the method based on photometric consistency.



(a) Tracking Performance (b) Number of cameras

Figure 5. (a) Our MAP estimate outperforms the baseline method in terms of the number of trajectories and the length of trajectories. (b) Our method leverages the large number of views, and shows a faster increasing curve than the baseline method.

6.2. Qualitative Evaluation

Visibility Boundary. We qualitatively demonstrate the performance of our MAP visibility estimation using the Bat Swing sequence by illustrating cameras in the visibility set in 3D, and showing the projection of the target patch in the images, as shown in Figure 6. This result shows a clean visibility boundary, showing the occluded views by the baseball bat.

3D Trajectory Recontruction. We generate an initial patch

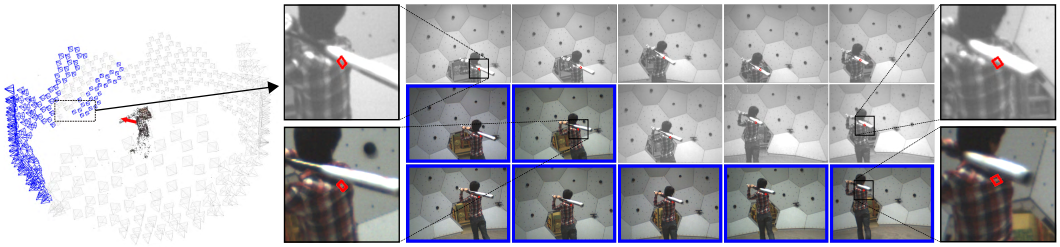


Figure 6. We qualitatively demonstrate the performance of MAP visibility estimation using the Bat Swing sequence. The normal of the selected patch is shown as a red arrow in the 3D view (left) and projected patch is shown as a red polygon in each image (right). The images with a blue boundary are the views that belongs to the visibility set. The bat occludes the patch and its effect can be seen as a "shadow" on the visibility set of cameras (left).

cloud for a selected time instance, and perform forwards and backwards patch tracking, up to 150 frames, for all the sequences summarized in Table 1. Figure 7 shows the reconstructed trajectories. The reconstructed time instances are color coded. Note that our method can be applied multiple times to different time instances to increase the density of the trajectories.

Circular movement: Three people rotate around the person at the center (Figure 7(a)). This experiment is used to evaluate our method in terms of visibility reasoning

Volleyball: Two people play volleyball (Figure 7(b)). We demonstrate an event where motion is fast and occlusion is severe. We are able to reconstruct the trajectories of the ball and players.

Bat swing: A person swings a baseball bat. The reconstructed long trajectories can provide a computational basis for sport analytics, capturing subtle motion (Figure 7(c)).

Falling boxes: A person collides with stacked boxes and the boxes collapse. The scene includes severe occlusion and topological change of the structure (Figure 7(d)).

Confetti: A person throws confetti in the air. 3D reconstruction of such sequences is challenging because of occlusion and appearance changes. Visibility estimation is challenging as the confetti are small and their appearance changes abruptly (Figure 7(e)).

Fluid motion: We generate turbulent flow in a room using a fan and small confetti (Figure 7(e))[3].

7. Discussion

We present a method to estimate the time-varying visibility for 3D trajectory reconstruction to leverage large numbers of views. We present novel cues (motion consistency, geometric consistency, and visibility regularization prior) for visibility estimation, and fuse them with the commonly used photometric consistency cue, within a MAP estimation framework. We demonstrate that our algorithm provides a more accurate visibility and, consequently, produces longer and denser 3D trajectories than a baseline using only photometric consistency. Unlike the photometric consistency cue, The motion consistency cue is complementary to the photometric cue, as it does not require the texture and the explicit 3D shape of the target 3D patch. Although the motion consistency cue can be ambiguous, this ambiguity, in practice, usually occurs for the cameras behind the target patch when the whole object body (including the patch) undergoes pure translation; this case is handled well by the geometric consistency of the patch and camera.

A key benefit of our approach is that it does not use any spatial or temporal regularization over the position of the point—the regularization used in our approach is over visibility. This results in "faithful" reconstruction of 3D point motion, that is not biased or smoothed out by prior models of deformation. The most common cause of failure are imaging artifacts, such as motion blur and saturation. As these kinds of artifacts are unavoidable especially when considering outdoor environments, an important direction of future work is to investigate techniques to re-associate points.

Appendix

Proof of Equation (8): Without loss of generality, we can define the projection matrix as $\mathbf{P} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \end{bmatrix}$. Then, Equation (6) can be rewritten as,

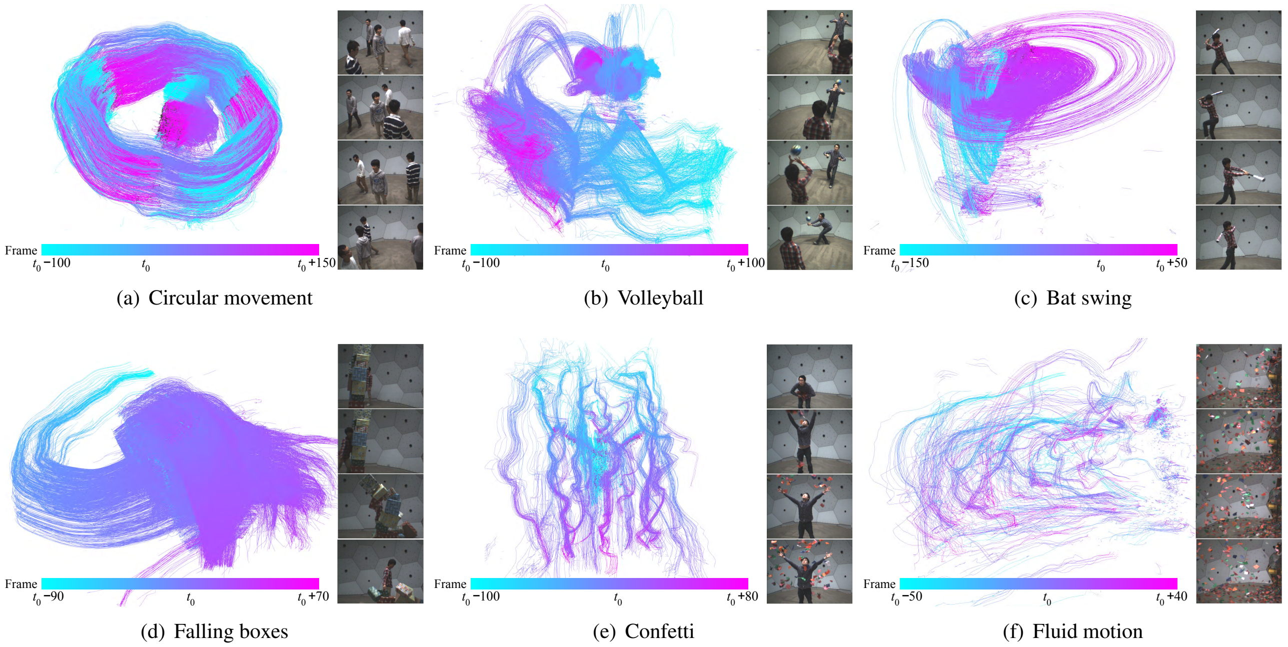
$$\begin{aligned} [\mathbf{X}'(t)]_\times \mathbf{X}(t) &= \mathbf{0} & (15)\\ [\mathbf{X}'(t+1)]_\times \mathbf{X}(t+1) &= \mathbf{0}, & (16)\end{aligned}$$

given $\mathbf{P} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \end{bmatrix}$ where $[\cdot]_\times$ is the skew-symmetric representation of cross product. $\mathbf{X}'(t)$ is linearly proportional to $\mathbf{X}(t)$ because of Equation (15) and thus, $\mathbf{X}'(t) = \alpha\mathbf{X}(t)$ where α is a scalar. $\alpha \neq 1$ because then $\mathbf{X}'(t) \neq \mathbf{X}(t)$.

From Equation (7), Equation (16) can be rewritten as,

$$\begin{aligned} \mathbf{0} &= [\mathbf{A}\mathbf{X}'(t) + \mathbf{a}]_\times (\mathbf{A}\mathbf{X}(t) + \mathbf{a}) \\ &= [\mathbf{A}\mathbf{X}'(t)]_\times \mathbf{A}\mathbf{X}(t) + [\mathbf{a}]_\times \mathbf{A}\mathbf{X}(t) \\ &+ [\mathbf{A}\mathbf{X}'(t)]_\times \mathbf{a} + [\mathbf{a}]_\times \mathbf{a} \\ &= (1-\alpha)\,[\mathbf{a}]_\times \mathbf{A}\mathbf{X}(t), \qquad (17)\end{aligned}$$

[3] For this result, we turned off geometric consistency by setting $\tau_c = 0$ and $\tau_p = 0$, as the objects are well approximated by planes.



(a) Circular movement (b) Volleyball (c) Bat swing

(d) Falling boxes (e) Confetti (f) Fluid motion

Figure 7. We reconstruct 3D trajectories in real world scenes in the presence of significant occlusion, large displacement, and topological change. The color codes the time that trajectory points are reconstructed. Note that each trajectory is individually reconstructed without any spatial or temporal regularization.

where $[\mathbf{AX}'(t)]_\times \mathbf{AX}(t) = \alpha [\mathbf{AX}(t)]_\times \mathbf{AX}(t) = \mathbf{0}$. Equation (17) implies Equation (8).

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grants No. 1353120 and 1029679. Hanbyul Joo was supported, in part, by the Samsung Scholarship.

References

[1] T. Basha, Y. Moses, and N. Kiryati. Multi-view Scene Flow Estimation: A View Centered Variational Approach. *IJCV*, 2012. 2

[2] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *TPAMI*, 2001. 5, 5.5

[3] C. Budd, P. Huang, M. Klaudiny, and A. Hilton. Topology-adaptive mesh deformation for surface evolution, morphing, and multiview reconstruction. *IJCV*, 2013. 2

[4] R. Carceroni and K. Kutalakos. Multi-view scene capture by surfel sampling: From video streams to non-rigid 3D motion, shape and reflectance. *IJCV*, 2002. 2, 6.1

[5] E. de Aguiar, C. Stoll, C. Theobalt, N. Ahmed, H.-P. Seidel, and S. Thrun. Performance capture from sparse multi-view video. *TOG*, 2008. 2

[6] E. de Aguiar, C. Theobalt, C. Stoll, and H.-P. Seidel. Marker-less deformable mesh tracking for human shape and motion capture. In *CVPR*, 2007. 2

[7] F. Devernay, D. Mateus, and M. Guilbert. Multi-Camera Scene Flow by Tracking 3-D Points and Surfels. In *CVPR*, 2006. 2, 5.1, 6.1

[8] Y. Furukawa, B. Curless, S. M. Seitz, and R. Szeliski. Towards internet-scale multi-view stereo. In *CVPR*, 2010. 1, 5.1

[9] Y. Furukawa and J. Ponce. Dense 3D motion capture from synchronized video streams. In *CVPR*, 2008. 2, 5.1, 6.1

[10] Y. Furukawa and J. Ponce. Accurate, dense, and robust multiview stereopsis. *TPAMI*, 2010. 2

[11] F. Huguet and F. Devernay. A variational method for scene flow estimation from stereo sequences. In *ICCV*, 2007. 2

[12] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. *IJCAI*, 1981. 4

[13] J. michael Frahm, P. Fite-georgel, D. Gallup, T. Johnson, R. Raguram, C. Wu, Y. hung Jen, E. Dunn, S. Lazebnik, and M. Pollefeys. Building rome on a cloudless day. In *ECCV*, 2010. 1, 5.1

[14] J. Quiroga, F. Devernay, and J. Crowley. Scene flow by tracking in intensity and depth data. In *CVPR Workshop*, 2012. 2

[15] Y. Sheikh, S. Nobuhara, H. Joo, H. Liu, L. Tan, L. Gui, M. Vo, B. Nabbe, I. Matthews, and T. Kanade. The panoptic studio. *Technical Report, CMU-RI-TR-14-04*, 2014. 6

[16] N. Snavely, S. M. Seitz, and R. Szeliski. Photo tourism: Exploring photo collections in 3D. *TOG*, 2006. 1, 5.1

[17] J. Starck and A. Hilton. Spherical matching for temporal correspondence of non-rigid surfaces. In *ICCV*, 2005. 2

[18] J. Starck and A. Hilton. Surface capture for performance-based animation. *CGA*, 2007. 2

[19] N. Sundaram, T. Brox, and K. Keutzer. Dense point trajectories by gpu-accelerated large displacement optical flow. In *ECCV*, 2010. 4

[20] T. Tung and T. Matsuyama. Dynamic surface matching by geodesic mapping for 3D animation transfer. In *CVPR*, 2010. 2

[21] K. Varanasi and A. Zaharescu. Temporal surface tracking using mesh evolution. In *ECCV*, 2008. 2

[22] S. Vedula, S. Baker, P. Rander, R. Collins, and T. Kanade. Three-dimensional scene flow. *TPAMI*, 2005. 2

[23] C. Vogel, K. Schindler, and S. Roth. 3D scene flow estimation with a rigid motion prior. In *ICCV*, 2011. 2

[24] A. Zaharescu, E. Boyer, and R. Horaud. Topology-adaptive mesh deformation for surface evolution, morphing, and multiview reconstruction. *TPAMI*, 2011. 2