Structure from Shadow Motion

Austin Abrams, Ian Schillebeeckx, Robert Pless
Washington University in St Louis
{abramsa|ians|pless}@cse.wustl.edu

Abstract

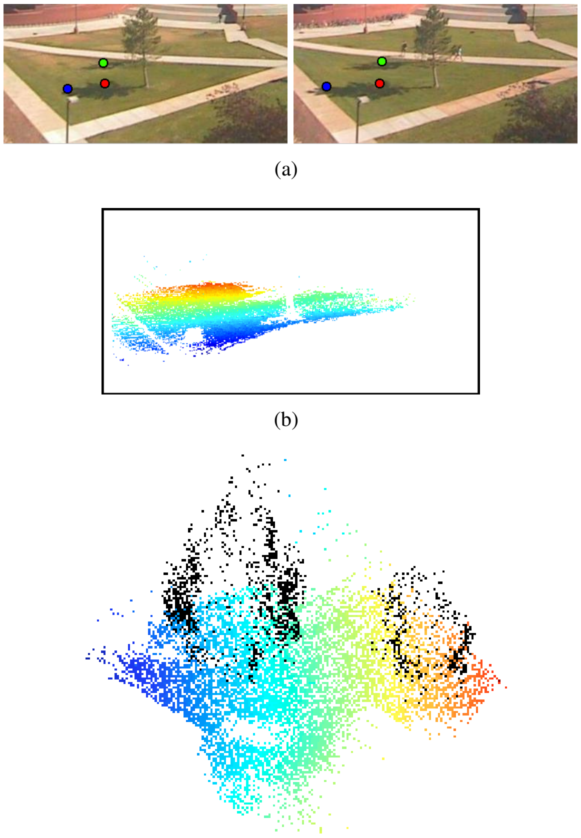
In outdoor images, cast shadows define 3D constraints between the sun, the points casting a shadow, and the surfaces onto which shadows are cast. This cast shadow structure provides a powerful cue for 3D reconstruction, but requires that shadows be tracked over time, and this is difficult as shadows have minimal texture. Thus, we develop a shadow tracking system that enforces geometric consistency for each track and then combines thousands of tracking results to create a 3D model of scene geometry. We demonstrate reconstruction results on a variety of outdoor scenes, including some that show the 3D structure of occluders never directly observed by the camera.

1. Introduction

We consider the problem of inferring outdoor scene structure based on the motion of shadows in long term time-lapse data. As the sun illuminates a scene from different directions during the day and during a year, it casts shadows onto the scene. The pattern of these shadows, and how they change, depends on what the camera directly views and nearby structures that cast shadows.

This provides a cue to solve for 3D scene structure from images captured from a single viewpoint. Because shadows are purely geometric objects, this approach does not require photometric camera calibration, and permits the sun as a calibrated light source. Furthermore, the constraint does not require the occluding object to be visible, and instead infers its position from the motion of shadows. Thus, anything that casts a shadow into the scene can be modeled, including structures hidden behind others in the scene.

What makes this problem difficult is that shadows are sparse and difficult to track. In any one frame, shadows only give constraints at shadow boundaries, so it is vital to track shadows across frames. This tracking is difficult because only the shape of the shadow boundary can be used to track, and only a few boundary points have distinguishable shapes. Also, in general scenes, the trajectory of a shadow point between images depends on the lighting direction, the


(a)

(b)

(c)

Figure 1. From a sequence of outdoor images (a), we track shadow movement; three color-coded example correspondences are shown on the two example images. These tracks are used as a cue for recovering sparse depth from a single view over time (b), where blue is closer to the camera and red is farther away. Our approach can even recover the structure of objects not directly visible to the camera, as seen in the 3D point cloud reconstruction (c), where black points are the reconstructed locations of shadow casters in the scene. In this case, shadow movements reveal the 3D structure of two trees, one of which the camera only observes from its shadows. We invite the reader to view the supplemental material, which shows rotating views of this 3D point cloud.

relative geometry of object casting the shadow, and the surface on which that shadow lands.

Our approach to find shadow tracks and scene structure is to exploit the geometry of outdoor illumination and derive explicit constraints relating the solar illumination direction, the 3D location of shadow casting objects, and a depth map of the scene. Even when the overall structure of the scene is unknown, these constraints give a rule for evaluating whether a possible track is geometrically consistent.

Our primary contributions are three-fold. First, for a geo-calibrated camera, we derive relationships between shadow trajectories, the scene depth, and the shadow casting positions without any assumptions on structure of the scene. Second, we create a fully automatic shadow tracking approach that is effective in tracking shadows because it makes use of these geometric relationships as a strong consistency check. Third, we show how to use these shadow trajectories to reconstruct scenes from a single view over time.

We find that our tracking approach works best when shadows are cast on relatively simple structures like hillsides and ground planes. However, we emphasize that the geometry is general for all forms of depth surfaces and shadow casters, and the shape of the recovered shadow-casting objects exhibit great complexity (such as the double trees in Figure 1).

2. Related Work

The geometry of shadows, light sources, and scene shapes has been explored in a large number of contexts. Early work characterized the projection of object boundary curves casting shadows onto ground planes of known geometry in line-drawings [20, 23]. These constraints define relations between the depth of points that lie along the same shadow boundary, and have been used to create 3D models of object shapes. For example, Bouget and Perona [7], and Kawasaki and Furukawa [16] use the shadows of polyhedral objects as a partially structured light source, where each straight shadow constrains groups of pixels to be co-planar in 3D. Bamber et al. [6] use a ground plane assumption to leverage 3D reconstruction and Belhumeur and Kriegman [18] characterize the equivalance class of object shapes that can cast the same set shadows onto a ground plane under varied lighting directions.

A different constraint is based on matching points on a cast shadow to the points on the object casting the shadow. The line from the shadow caster to the shadow needs to be unobstructed, providing additional constraints on scene points that lie along this line [10, 11, 22, 25].

In the context of outdoor imagery, shadow constraints have been used for calibration and scene structure estimation in a variety of contexts. In a recent application to photo forensics, Kee et al. [17] made use of shadow correspondence to expose photos with inconsistent shadows as being manipulated.

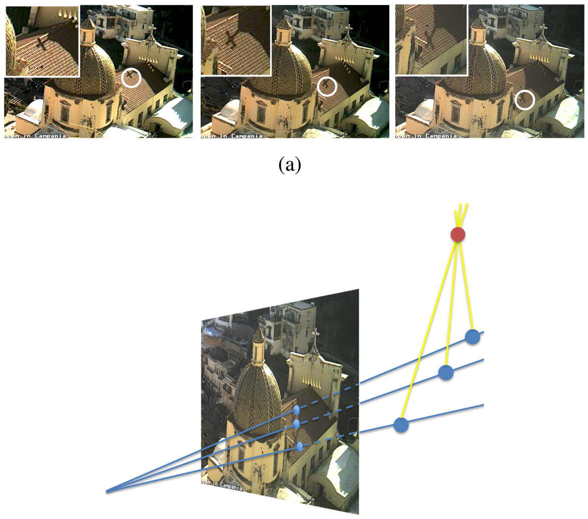
Antone and Bosse [5] assume a stationary camera with known internal calibration, timestamps, geo-location, and define the analytic constraint between vertical objects and the shadow they cast onto a flat ground plane. Junejo and Foroosh [15] calibrate the intrinsic parameters of a camera and solve for constraints on camera geo-location from the trajectories of shadows of two points moving across a ground plane, and Wu et. al [24] extend this to complete camera calibration, geolocation, and the relative heights of the two shadow casting points, even if those are not in the field of view. Caspi and Werman [9] focus more on modeling the scene structure and use the set of shadows of cast by two vertical edges in the scene to reconstruct a plane and parallax model of scene structure.

Abrams et al. [2] create a scene depth map from co-linearity constraints between the sun direction, a shadow caster, and the object casting a shadow. With enough lighting directions this supports the creating of a scene depth map. In this paper, we extend their approach to work with moving shadows, rather than correspondences between a shadow and its shadow caster. The main benefits are that the resulting geometry is more general and able to reconstruct objects the camera cannot directly see, and in many cases, tracking a shadow from frame to frame is easier than finding a correspondence between a shadow and its shadow caster. [2] use a naive shadow correspondence algorithm that would not work for any of the examples shown in this paper.

The current work combines the co-linearity constraints of [2] with constraints from shadow tracking [24] to better constrain the 3D shape of points in the field of view, and to provide constraints on shapes not directly visible to the camera. To our knowledge it is the first to derive constraints relating scene geometry from shadow tracks without making any simplifying assumptions about that geometry.

3. Structure from Shadow Motion

The fundamental constraint this paper considers is the relationship between shadow motion and scene structure. We represent this shadow motion by finding corresponding points on shadows cast by the same objects in different images. We call the set of correspondences from one shadow caster a *track*. In this section we characterize the geometric constraints a shadow track must obey. We defer the discussion of generation those tracks to Section 4 because our tracking algorithm uses these constraints as part of a consistency check.



(a)

(b)

Figure 2. Shadow tracking geometry. By tracking a shadow's movement through three frames, shown as white circles in (a) (insets shown in top-left), we can recover the 3D geometry (b) of the shadow casting object (red) and the surfaces that received a shadow (blue) by solving Equation 1 with respect to the given lighting directions at each time (yellow).

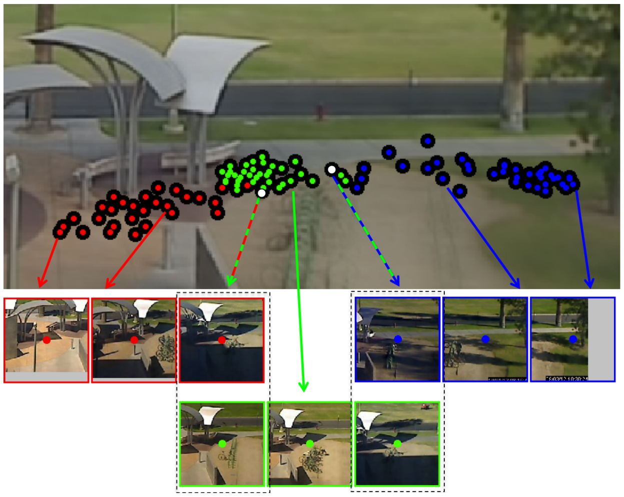


Figure 3. An example of track connectivity using tracks recovered from our algorithm. The top figure shows the color-coded locations of three tracks in a scene. The bottom row of images shows crops from these three tracks. This group of tracks has two "crossover" locations (indicated in white in the top image, and with dotted black outlines below), where two tracks constrain the same pixel. We use this tracking overlap to fix the depth scale across tracks when solving for scene structure.

3.1. From Tracks to Structure

A track T_i is defined as a set of location-time pairs $\{(\mathbf{x}_1, t_1), \ldots, (\mathbf{x}_m, t_m)\}$, which we can use to recover the depth $d_{\mathbf{x}} \in \mathbb{R}$ of all pixels $\mathbf{x}$ on the track, and the location of a shadow caster $C_i \in \mathbb{R}^3$. The set of all $(\mathbf{x}, t)$ location-time pairs in track T_i must satisfy the following 3D spatial constraint:

$$r_{\mathbf{x}} d_{\mathbf{x}} + L_t \alpha_{ti} = C_i, \tag{1}$$

where $r_{\mathbf{x}} \in \mathbb{R}^3$ is the unit vector which passes through pixel $\mathbf{x}$ in the image plane, $L_t \in \mathbb{R}^3$ is the lighting direction at time t, and α_{ti} is the 3D distance along the lighting direction between a shadow and its caster. This geometry is visualized in Figure 2.

This work assumes that the camera geo-centric calibration (and therefore $r_{\mathbf{x}}$ all pixels $\mathbf{x}$) is known, as well as the per-image lighting directions L, which can be recovered with known geolocation and timestamps via a solar lookup [21].

Equation 1 considers one track and relates depths in the scene to the positions of a shadow caster only up to an unknown scale factor. Given a set of tracks in the scene $\mathcal{T}$, the tracks-to-structure optimization matches scale factors between tracks with a constrained linear system over the depth d, 3D distances α, and occluders C:

$$\underset{d,\alpha,C}{\arg\min} \sum_{T_i \in \mathcal{T}} \sum_{(\mathbf{x},t) \in T_i} ||r_{\mathbf{x}} d_{\mathbf{x}} + L_t \alpha_{ti} - C_i||^2, \tag{2}$$

under the constraints that $d \geq 1$ (to set the scale) and $\alpha \geq 0$.

This fixes the scale factor between tracks that overlap, because if one image location $\mathbf{x}$ is part of two shadow tracks (at different times), it must have a consistent depth $d_{\mathbf{x}}$ in both. Therefore, it is desirable that a tracker return as dense a set of tracks as possible, to promote high crossover, and thus support many tracks sharing the same depth scale. Figure 3 shows that in real scenes, there is typically a large amount of crossover which connects distant pixels through a network of constraints.

3.2. Geometric Consistency

When a track passes through two or more frames, solving for the unknowns defines 3D scene locations for each shadow point, and the location of the shadow casting object C_i. Therefore, we can test for the *geometric consistency* of a track by estimating how well T_i satisfies this linear relationship.

If we believe some pixel $\mathbf{x}$ to be under shadow at time t, then the projection of all feasible shadow casters is exactly the projection of the lighting direction onto the image (termed the "episolar line" in [2]). If a track is geometrically consistent, then each of these lines will intersect in a common location, the location of the shadow caster in the image. Therefore, this geometric consistency check can be visualized as an intersection test between many image-space lines, as demonstrated in Figure 4.

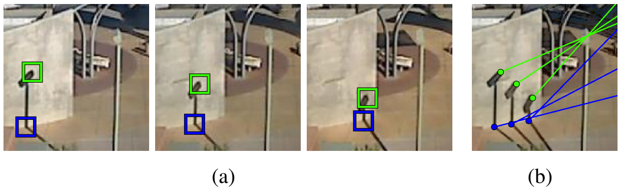


Figure 4. A visualization of the geometric consistency check. In (a), both the green and blue tracks have similar appearances through time. In (b), we annotate each detection with a line corresponding its set of feasible shadow casters. This additional geometric check reveals that the green track is consistent with a single shadow caster (since all green lines intersect in a common location), while the blue track is inconsistent.

In practice, we test a track T_i's geometric consistency by solving Equation 1 for $d_{\mathbf{x}}$, α_{ti}, and C_i. If a track is perfectly consistent, there will be no error, and the angle between L_t and $(C_i - r_{\mathbf{x}} d_{\mathbf{x}})$ will be 0 degrees for all $(\mathbf{x}, t) \in T_i$. In the following section we reject a track if, for any $(\mathbf{x}, t) \in T_i$, this angle is greater than a half degree, or if $d_{\mathbf{x}}$ is negative (corresponding to observing a shadow behind the camera), or if any α_{ti} is negative (corresponding to a shadow being projected the wrong direction).

3.3. Challenges and Limitations

The geometric constraints relating shadows to scene structure have several formal ambiguities. Characterizing the ambiguities helps helps to define limits of the approach and suggests properties that are important for robust shadow tracking.

The first ambiguity affects the tracking step and is a corollary to the aperture problem in standard optic flow. When tracking a shadow cast by any straight edge (such as a pole), the shape of the shadow will be a strong edge, and the motion of the shadow along this edge is not well constrained. Unfortunately, the additional geometric constraints do not remove the aperture problem; there are still an infinite number of geometrically-consistent tracks that pass through the shadow's shape. Figure 5 illustrates this case.

The second ambiguity affects reconstruction in the presence of degenerate lighting configurations. In a video sequence taken over the span of one day, the set of illumination directions is often degenerate. In the extreme case of images taken over a day during the equinox at the Equator, the sun passes directly overhead, and all shadows are cast exactly along east-west lines. Thus, while tracks may cross each other, tracks will never "move north and south" to unify the scale factors of different parts of the scene. While this effect is mitigated when the lighting configuration is non-degenerate (e.g. when the camera is far from the Equator, or during the summer and winter solstices), we find the reconstruction step for images from a single day to be poorly conditioned.

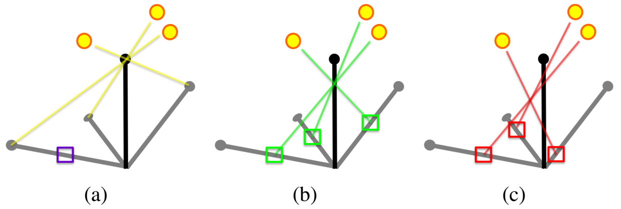


Figure 5. Shadow tracking under geometric constraints still suffers from the aperture problem. Consider a vertical pole casting shadows across three frames shown as a single composite image in (a). Where does the shadow in the blue box go in the other two frames? The tracks in (b) and (c) are both geometrically consistent, and have exactly the same appearance over time, but only the track in (b) is correct.

To alleviate the aperture problem, we track shadows in a spatially smooth manner so that ambiguous edges are more likely to follow the movements of more discriminative shapes. We handle the second ambiguity by working with data spanning over months to get a more diverse set of lighting conditions. In this case, the set of sun directions is not planar, but rather lies on a full-rank subset of the unit sphere. This is not a large limitation, because there exist large archives of outdoor scenes which have already been capturing live webcam streams for years [14, 19].

4. Shadow Tracking

The shadow tracking approach has four steps. First, we detect where shadows are in each image, and describe each point with a local binary pattern. Second, frame-to-frame matches are found between pairs of images with similar lighting directions. Third, these matches are linked together in an approach that guarantees geometric consistency, and finally, these tracks are extended to cover a greater temporal extent.

4.1. Shadow Detection and Description

Given a set of input imagery, we begin by computing the Canny edges [8] on each image. An edge could occur due to shadows, as well as depth or texture. Shadows from depth discontinuities or scene texture remain stationary through time, while cast shadows move as the lighting direction varies. Therefore, for each image, we take the pixels on the edge map, and remove any that were on an edge map more than 10% of the time. The result is a set of pixels that mostly come from moving cast shadows; see Figure 6 for an example.

To describe each point $(\mathbf{x}, t)$, we compute a local binary pattern feature in a circle with a 10-pixel radius centered

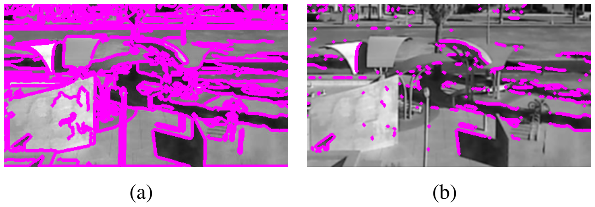


Figure 6. To detect shadows in a time-lapse sequence, we run an edge detection algorithm (a) on each image, and keep any edge pixel that was not an edge in many other frames, largely removing edges from persistent structure or texture (b).

at the detection of interest. Our feature f is a bit-vector which encodes if the pixel $\mathbf{x}$ has a greater intensity than each of the sample locations around it at time t. To alleviate image noise, we additionally smooth the image with a 3x3 average filter before extracting features. The result is a set of detections $\mathcal{D}$ with feature descriptors, denoted as $(\mathbf{x}, t, f)$ triplets. As shorthand, we denote the detection i as $(\mathbf{x}_i, t_i, f_i)$.

4.2. Frame-to-frame matching

In the next step, we create many frame-to-frame shadow matches. We repeat this process for many image pairs from nearby lighting directions to create a rich set of 2-frame correspondences. Specifically, given two images t_1 and t_2, each with their own sets of detections $D_1, D_2 \subset \mathcal{D}$, we want to find a matching function from D_1 to D_2.

Assuming that shadow motion is small across nearby lighting directions, the shadow's location and appearance should not vary dramatically. For consistency in this section, we denote variables with i and j subscripts when they refer to detections in D_1 and D_2, respectively. For each detection i, we gather a set of possible matches $M_i \subset D_2$:

$$M_i = \{j \in D_2 \,| s_{ij} > \tau\}, \tag{3}$$

where τ is a minimum matching score, and s_{ij} is a score that attains a large value when detections i and j have a similar appearance (in terms of their local binary patterns) in similar locations in the image:

$$s_{ij} = w_{\text{location}}(i,j)\;\; w_{\text{appearance}}(i,j) \tag{4}$$

$$w_{\text{location}}(i,j) \;\; = \exp\left(-\frac{(\mathbf{x}_i - \mathbf{x}_j)^2}{\sigma_{\mathbf{x}}^2}\right) \tag{5}$$

$$w_{\text{appearance}}(i,j) \;\; = \exp\left(-\frac{(f_i - f_j)^2}{\sigma_f^2}\right) \tag{6}$$

Furthermore, we expect the matching function should be spatially smooth; nearby shadows at t_1 should match to nearby locations at t_2. We model the matching function as a nonparametric warp by assigning a warp vector $\mathbf{u}_i \in \mathbf{R}^2$ to each detection i, and optimize the following:

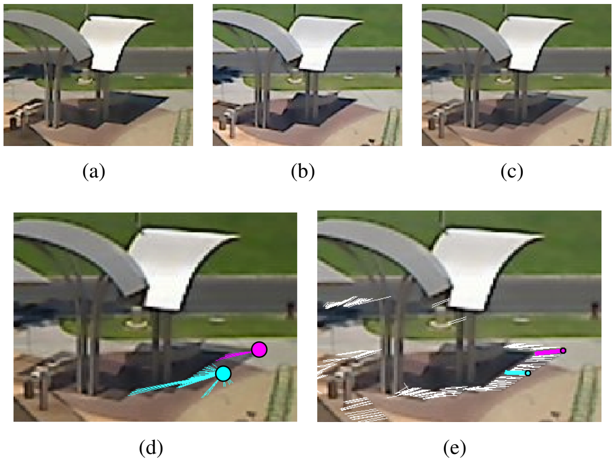


Figure 7. A visualization of the matching algorithm. Suppose we want to match the shadow edges in (a) to (b), shown as an average image in (c) for visualization. We first find a set of candidate matches, shown in (d) for two detections. The pink detection comes from a corner and only has a few candidate matches, but the cyan detection comes from a shadow edge and is less discriminative, so it could match to many more. We optimize for a smooth warp across the image that maps each point to one of its matches, and accept frame-to-frame matches (white) that came close to one of their candidates (e).

$$\arg\min_{\mathbf{u}} \sum_{i \in D_1} \min_{j \in M_i} ||\mathbf{x}_i + \mathbf{u}_i - \mathbf{x}_j||^2 + \left|\left|\mathbf{u}_i - \sum_{i' \in D_1} v_{ii'}\mathbf{u}_{i'}\right|\right|^2 \tag{7}$$

The first term encourages the warp to push $\mathbf{x}_i$ toward one of its potential matches, and the second term is a Laplacian smoothness term, where $v_{ii'} \propto w_{\text{location}}(i, i')$[1], with $v_{ii} = 0$.

We optimize this objective with gradient descent, initializing $\mathbf{u}_i$ as $\mathbf{x}_j - \mathbf{x}_i$, where $j \in M_i$ is the detection where s_{ij} is maximal. If M_i is empty, we initialize $\mathbf{u}_i$ to 0. After convergence, we create a match between detections $i \in D_1$ and $j \in D_2$ whenever $||\mathbf{x}_i + \mathbf{u}_i - \mathbf{x}_j||$ is less than 2 pixels.

4.3. Linking together matches

Given a large set of frame-to-frame matches, we now link these matches into long tracks. There are a few obvious baselines for this problem which we found to be insufficient, so we first describe two baseline track linking approaches, and then our approach which enforces geometric consistency through time.

[1] Although one could define w_{location} and v with different bandwidths, we chose to use the same bandwidth $\sigma_{\mathbf{x}}$ for both for simplicity.

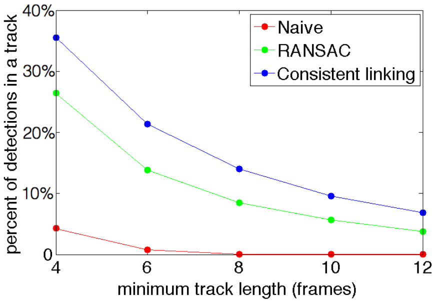


Figure 8. The percent of detections included in a geometrically-consistent track, as a function of the minimum acceptable track length, for three track linking approaches. The naive method incrementally links together matches with a common endpoint, the RANSAC method finds geometrically-consistent subtracks within those, and the constrained linking approach incrementally links together matches with a common endpoint, so long as it maintains the track's geometric consistency. The third approach performs the best, incorporating more detections into long, geometrically consistent tracks.

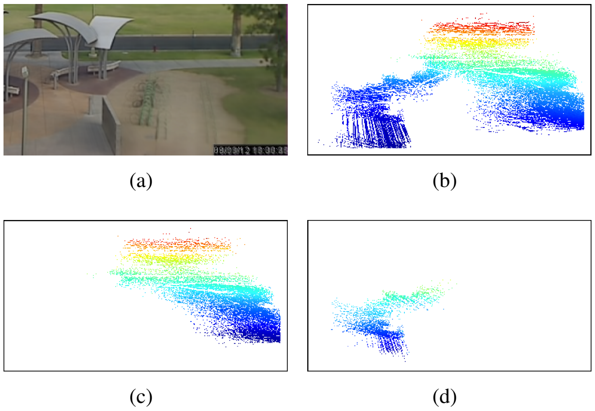


Figure 9. Track expansion encourages well-connected reconstructions. Given an image sequence (average image shown in (a)), our full algorithm returns a depth map that covers most of the ground plane (b). (c) and (d) show two "connected components" of a reconstruction fueled with un-expanded tracks. Since the tracks that build these 3D models are shorter and cover a smaller area, overlap between tracks is less likely and the reconstructions are disconnected.

The simplest method to link matches together is to iteratively group together two matches if they share a common endpoint (i.e. they share a detection), unless it creates a track that passes through the same frame twice. This process is repeated until the track cannot be extended any more, and if the track is sufficiently long, we keep it.

We found that this method does not give many tracks which satisfy the geometric consistency check, even for shadows with discriminative local binary patterns. This is alleviated somewhat if each track is filtered through a RANSAC routine: we choose two random detections in the recovered track, find their shadow caster, and see which other detections in the track are consistent with that. After many rounds, we keep whichever subtrack has the largest number of inliers.

The best approach we found was to incorporate the geometric consistency check into the track linking procedure. We still iteratively group together matches with common endpoints, but at each step check if the next addition would break the track's geometric consistency so far. If there are multiple possible extensions, we choose the one that is most consistent (in terms of angular reconstruction error as in Section 3.2).

These three approaches are shown in Figure 8 using frame-to-frame matches for the scene from the first page. This plot shows how many detections in a scene are incorporated into a track after each track linking strategy (for varying thresholds on the minimum acceptable track length). The more detections that are incorporated into some consistent track, the more complete the model will be. The geometrically-aware linking algorithm outperforms the other two, typically explaining the motion of 5% to 10% more detections.

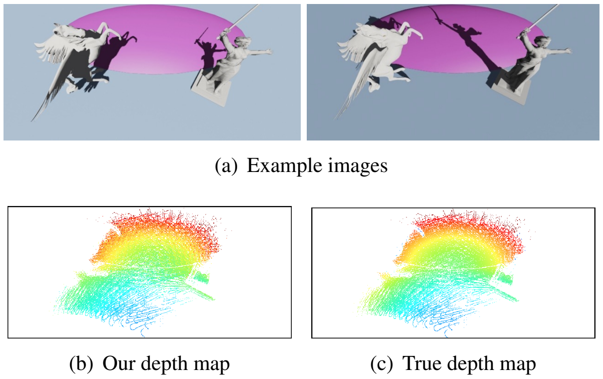
4.4. Track expansion

As a final step, we take each track and try to extend it into frames it does not yet pass through. For each track, we find all detections j in new frames that have a high score s_{ij} to some detection i already in the track. If that detection is geometrically consistent with the rest of the track, we append it, and repeat until no such j is found.

Although the tracks generated before this step are already long and consistent, this extra expansion step helps to create crossover between tracks that might not have crossed over before. Figure 9 demonstrates this property, where the un-expanded track set is not quite expansive enough to connect together two large depth components in a scene.

4.5. Implementation Details

Rather than perform the full constrained least squares reconstruction as in Equation 2, we take advantage of an incremental reconstruction for robustness and speed. We begin by reconstructing a seed track, chosen as the track that overlaps with the most others. One by one, we reconstruct the track that overlaps the most with the reconstruction so far, under the additional constraint that the depth for pixels already reconstructed stay constant. If the track is no longer geometrically consistent under those additional constraints, we remove the (likely erroneous) detections from that track that accrued the most error and pick another track. This process repeats until there are no tracks that overlap with



(a) Example images

(b) Our depth map

(c) True depth map

Figure 10. Results on a synthetic sequence. Our recovered single-view depth matches the ground truth to within 2% error.

the model. A typical reconstruction takes about a minute, which is fast compared to the full least squares optimization, which (even with commercial sparse linear system packages) often cannot fit the full linear system in memory.

For webcam data, geometric camera calibration was derived by manually corresponding scene points to Google Earth models [3] or using manually specified shadow to shadow caster correspondences [2].

The algorithm, starting from a calibrated camera and a set of 200 images requires, on average, three hours to create 3D models. About 55% of the time is spent on finding and optimizing frame-to-frame matches, another 35% spent on linking together matches, 8% spent on expanding tracks, and the remaining time spent loading images, detecting/describing shadows, and incremental reconstruction.

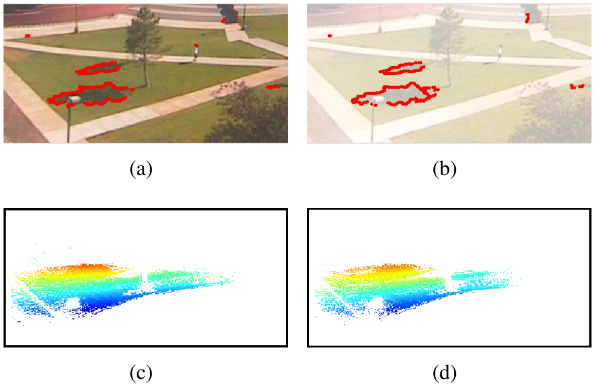
In our implementation, we find matches between each image and its 5 nearest neighbors, where distance is measured in terms of angular difference in sun position. We use the parameters $\sigma_{\mathbf{x}}$ as 5% of the main diagonal length, σ_f as 5% of the feature dimensionality, τ as 0.1, and the minimum track length as 8 frames.

5. Results

To evaluate the approach, we explore results based on a synthetic scene and a collection of images takes by webcams over long periods of time. We invite the reader to view the supplemental video to view the reconstructions in 3D.

To test the accuracy of our approach, we rendered a synthetic scene using virtual sun positions over the span of a year. This synthetic scene is challenging because most shadows are projected onto a curved surface, so shadows distort their shape for even small movements. After fixing the scales between the ground truth depth and our reconstruction, our model's depth has an error of about 2%.

Recent research in single-view shape-in-the-wild approaches shows that the camera's color calibration needs to be known a priori [4, 13], or that the camera's response cannot change through time [1]. Figure 11 shows an evaluation of the robustness of our approach to unknown color calibration. For this experiment, we artificially distorted each image in a sequence with a random exposure and radiometric response chosen from [12] (i.e. a different response profile *for each image*), and ran our algorithm on both the original and distorted data. Since our features come from Canny edges and local binary patterns, which are both invariant to response and exposure changes, the result is identical before and after distortion. This level of distortion would cause dramatic errors in any of the photometric methods above.



(a)

(b)

(c)

(d)

Figure 11. An evaluation on the robustness to unknown radiometric calibration. The top row shows an example image an image sequence (a), and the same image artificially distorted by an unknown exposure and tone mapping curve (b). Each image is annotated with the set of detections that were successfully tracked through that image. When our algorithm runs on the original (c) and distorted (d) sequences, the results are almost identical.

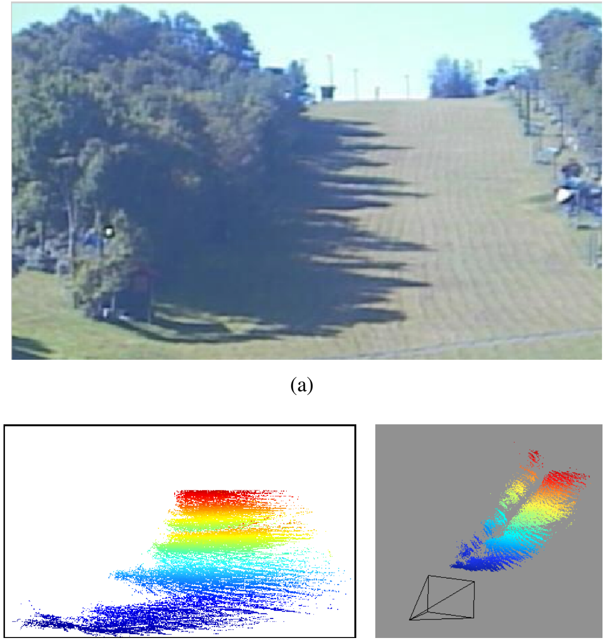
6. Conclusions

In this paper, we introduce a framework for single-view shape in the wild. This approach extracts shadow trajectories using a new geometric consistency measure and response-invariant features. In some cases, this approach allows us to use shadows to recover the shape of objects the camera never directly saw.

We share a similar error mode to traditional structure from motion, in that scenes with relatively little shadow texture are poorly modeled. If there aren't many shadows with discriminative shapes, then the matching routine is more ambiguous, and the whole pipeline suffers.

Our approach only tracks shadows on the boundary between darkness and light, because they are more discriminative than pixels on the shadow's interior. However, these interior pixels carry useful information, and could be used to determine more complete 3D models of shadow casters.

Our largest assumption is that the scene remains static,



(a)

(b)

(c)

Figure 12. Results on a webcam sequence looking at a ski slope in the spring (a). Finding shadow trajectories in this sequence provides a cue to reconstruct a sparse depth map (b). The final figure show a novel view (c) of the underlying point cloud. We invite the reader to watch the supplemental video, which shows rotating views of this point cloud.

which can break down when imagery comes over the span of a few months. Our incremental reconstruction and tracking algorithms are robust enough that if the scene changes geometry briefly, we only capture the most static mode. However, more formally addressing dynamic geometry is an exciting avenue for future work.

The largest limitation of our shadow tracker comes from our relatively rigid feature representation. Our local binary pattern features do not have any invariance to scale or rotation, so we depend on the shape of the shadow to not distort wildly between images with similar lighting conditions. Therefore, this approach works best when reconstructing planar surfaces or surfaces with small curvature. However, our geometric consistency checks are valid for any kind of depth surface, meaning that any future change to the feature representation is a drop-in replacement.

References

[1] A. Abrams, C. Hawley, and R. Pless. Heliometric stereo: shape from sun position. In *Proc. European Conference on Computer Vision*, 2012. 7

[2] A. Abrams, K. Miskell, and R. Pless. The episolar constraint: monocular shape from shadow correspondence. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2013. 2, 3, 7

[3] A. Abrams and R. Pless. Webcams in context: Web interfaces to create live 3D environments. In *Proc. ACM SIGMM International Conference on Multimedia (ACMMM)*, 2010. 7

[4] J. Ackermann, F. Langguth, S. Fuhrmann, and M. Goesele. Photometric stereo for outdoor webcams. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2012. 7

[5] M. Antone and M. Bosse. Calibration of outdoor cameras from cast shadows. In *IEEE International Conference on Systems, Man and Cybernetics*, volume 3, pages 3040–3045. IEEE, 2004. 2

[6] D. C. Bamber, J. D. Rogers, and S. F. Page. A method for 3D scene recognition using shadow information and a single fixed viewpoint. In *Visual Information Processing*, 2012. 2

[7] J.-Y. Bouguet and P. Perona. 3d photography using shadows in dual-space geometry. *International Journal of Computer Vision*, 35(2):129–149, 1999. 2

[8] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679 – 698, Nov 1986. 4

[9] Y. Caspi and M. Werman. Vertical parallax from moving shadows. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 2309–2315, 2006. 2

[10] M. Chandraker, S. Agarwal, and D. Kriegman. ShadowCuts: Photometric stereo with shadows. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2007. 2

[11] M. Daum and G. Dudek. On 3-d surface reconstruction using shape from shadows. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 1998. 2

[12] M. D. Grossberg and S. K. Nayar. What is the space of camera response functions? In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2003. 7

[13] N. Jacobs, B. Bies, and R. Pless. Using cloud shadows to infer scene structure and camera calibration. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, June 2010. 7

[14] N. Jacobs, N. Roman, and R. Pless. Consistent temporal variations in many outdoor scenes. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, June 2007. 4

[15] I. Junejo and H. Foroosh. Estimating geo-temporal location of stationary cameras using shadow trajectories. In *Proc. European Conference on Computer Vision*, pages 318–331. Springer, 2008. 2

[16] H. Kawasaki and R. Furukawa. Shape reconstruction and camera self-calibration using cast shadows and scene geometries. *International Journal of Computer Vision*, 83(2):135–148, 2009. 2

[17] E. Kee, J. O'Brien, and H. Farid. Exposing photo manipulation with inconsistent shadows. *ACM Transactions on Graphics*, 32(3):28:1–12, Sept. 2013. 2

[18] D. J. Kriegman and P. N. Belhumeur. What shadows reveal about object structure. *JOSA A*, 18(8):1804–1813, 2001. 2

[19] J.-F. Lalonde, A. A. Efros, and S. G. Narasimhan. Webcam clip art: Appearance and illuminant transfer from time-lapse sequences. *ACM Transactions on Graphics*, 28(5), December 2009. 4

[20] D. Lowe and T. Binford. The interpretation of three-dimensional structure from image curves. In *International Joint Conference on Artificial Intelligence*, 1981. 2

[21] I. Reda and A. Andreas. Solar position algorithm for solar radiation applications. In *NREL Report No. TP-560-34302*, 2003. 3

[22] S. Savarese, H. Rushmeier, F. Bernardini, and P. Perona. Shadow carving. In *Proc. IEEE International Conference on Computer Vision*, 2001. 2

[23] S. A. Shafer and T. Kanade. Using shadows in finding surface orientations. *Computer Vision, Graphics, and Image Processing*, 22(1):145 – 176, 1983. 2

[24] L. Wu, X. Cao, and H. Foroosh. Camera calibration and geo-location estimation from two shadow trajectories. *Computer Vision and Image Understanding*, 114:915–927, 2010. 2

[25] Y. Yu and J. T. Chang. Shadow graphs and surface reconstruction. In *Proc. European Conference on Computer Vision*, 2002. 2