FOCUS: Clustering Crowdsourced Videos by Line-of-Sight

Puneet Jain
Duke University
puneet.jain@duke.edu

Justin Manweiler
IBM T. J. Watson
jmanweiler@us.ibm.com

Arup Acharya
IBM T. J. Watson
arup@us.ibm.com

Kirk Beaty
IBM T. J. Watson
kirkbeaty@us.ibm.com

ABSTRACT

Crowdsourced video often provides engaging and diverse perspectives not captured by professional videographers. Broad appeal of user-uploaded video has been widely confirmed: freely distributed on YouTube, by subscription on Vimeo, and to peers on Facebook/Google+. Unfortunately, user-generated multimedia can be difficult to organize; these services depend on manual "tagging" or machine-mineable viewer comments. While manual indexing can be effective for popular, well-established videos, newer content may be poorly searchable; live video need not apply. We envisage video-sharing services for live user video streams, indexed automatically and in realtime, especially by shared content. We propose *FOCUS*, for Hadoop-on-cloud video-analytics. FOCUS uniquely leverages visual, 3D model reconstruction and multimodal sensing to decipher and continuously track a video's line-of-sight. Through spatial reasoning on the relative geometry of multiple video streams, FOCUS recognizes shared content even when viewed from diverse angles and distances. In a 70-volunteer user study, *FOCUS' clustering correctness is roughly comparable to humans*.

Categories and Subject Descriptors

H.3.4 [**Information Storage and Retrieval**]: Systems and Software

General Terms

Algorithms, Design, Experimentation, Performance

Keywords

Crowdsourcing, Line-of-sight, Live Video, Multi-view Stereo

SenSys '13, November 11 - 15 2013, Roma, Italy

1. INTRODUCTION

With the ubiquity of modern smartphones, photo and video journalism is no longer limited to professionals. By virtue of having a Internet-connected videocamera always at arms' reach, the average person is always ready to capture and share exciting or unexpected events. With the advent of Google Glass, the effort required to record and share will soon disappear altogether. The popularity of YouTube and video sharing on social media (e.g., Facebook and Google+) is evidence enough that many already enjoy creating and distributing their own video content, and that such video content is valued by peers. Indeed, major news organizations have also embraced so-called "citizen journalism," such as CNN *iReport*, mixing amateur-sourced content with that of professionals, and TV broadcasting this content worldwide.

Amateur video need not be immediately newsworthy to be popular or valuable. Consider a sporting event in a crowded stadium. Often, spectators will film on their smartphones, later posting these videos to YouTube, Facebook, etc. Even if their content is generally mundane, these videos capture the unique perspective of the observer, and views potentially missed by professional videographers, even if present. Unfortunately, given a multitude of sources, such video content is difficult to browse and search. Despite the attempts of several innovative startups, the value can be lost due to a "needle in a haystack" effect [3–6]. To organize, websites like YouTube rely on an haphazard index of user-provided tags and comments. While useful, tags and comments require manual effort, may not be descriptive enough, are subject to human error, and may not be provided in realtime — thus, not amenable to live video streams. In contrast, we envisage a realtime system to extract content-specific metadata for live video. Unlike related work applying lightweight sensing in isolation [1], our approach blends sensing with computer vision. This metadata is sufficiently precise to immediately identify and form "clusters" of synchronized streams with related content, especially, a precise subject in shared "focus."

In this paper, we propose *FOCUS*, a system for realtime analysis and clustering of user-uploaded video streams, especially when captured in nearby physical locations (e.g., in the same stadium, plaza, shopping mall, or theater). Importantly, FOCUS is able to deduce content similarity even when videos are taken from dramatically different perspectives. For example, two spectators in a soccer stadium may film a goal from the

East and West stands, respectively. With up to 180 degrees of angular separation in their views, each spectator may capture a distinct (*uncorrelated*) background. Even the shared foreground subject, the goalkeeper, will appear substantially different when observed over her left or right shoulder. Without a human understanding of the game, it would be difficult to correlate the East and West views of the goalkeeper, while distinguishing from other players on the field. Novelly, FOCUS' analysis reasons about the relative camera location and orientation of two or more video streams. The geometric intersection of line-of-sight from multiple camera views is indicative of shared content. Thus, FOCUS is able to infer logical content similarity even when video streams contain little or no visual similarity.

Users record and upload video using our Android app, which pairs the video content with precise GPS-derived timestamps and contextual data from sensors, including GPS, compass, accelerometer, and gyroscope. Each video stream arrives at a scalable service, designed for deployment on an infrastructure-as-a-service cloud, where a Hadoop-based pipeline performs a multi-sensory analysis. This analysis blends smartphone sensory inputs along with *structure from motion*, a state-of-the-art technique from computer vision. For each stream, FOCUS develops a model of the user's line-of-sight across time, understanding the geometry of the camera's view — position and orientation, or *pose*. Across multiple user streams, FOCUS considers video pairs, frame by frame. For each pair, for each frame, FOCUS determines commonality in their respective lines-of-sight, and assigns a "similarity" score. Across multiple feeds, across multiple frames, these scores feed a pairwise *spatiotemporal* matrix of content similarity. FOCUS applies a form of clustering on this matrix, invoking ideas from community identification in complex networks, returning groups with a shared subject.

FOCUS remains an active research project, as we endeavor to further harden its accuracy and responsiveness. However, despite ample room for further research, *we believe that FOCUS today makes the following substantial contributions:*

1. **Novel Line-of-Sight Video Content Analysis:** FOCUS uses models derived from multi-view stereo reconstruction to reason about the *relative position and orientation of two or more videos*, inferring shared content, and providing robustness against visual differences caused by distance or large angular separations between views.
2. **Inertial Sensing for Realtime Tracking:** Despite an optimized Hadoop-on-cloud architecture, the computational latency of visual analysis remains substantial. FOCUS uses lightweight techniques based on smartphone sensors, as a form of dead reckoning, to provide *continuous realtime video tracking* at sub-second timescales.
3. **Clustering Efficacy Comparable to Humans:** In a 70-volunteer study, clustering by modularity maximization on a "spatiotemporal" matrix of content similarity yields a *grouping correctness comparable to humans*, when measured against the ground truth of videographer intent.

2. INTUITION

User-generated, *crowdsourced*, multimedia has clear value. YouTube and other sharing sites are immensely popular, as is community-sourced video on Facebook and Google+. The value of particular shared content, however, can be lost in volume, due to the difficulty of indexing multimedia. Today, user-generated "tags" or mineable text comments aid peers while browsing rich content. Unfortunately, newer and real-time content cannot benefit from this metadata.

We envisage large-scale sharing of live user video from smartphones. Various factors are enabling: the pervasiveness of smartphones and increasing use of social apps; improving cellular data speeds (e.g., 4G LTE); ubiquity of Wi-Fi deployments; increased availability and adoption of scalable, cloud-based computation, useful for low-cost video processing and distribution; enhanced battery capacity, computational capabilities, sensing, and video quality of smartphones; and the advent of wearable, wirelessly-linked videocameras for smartphones, such as in Google Glass.

The dissemination of live, mobile, crowdsourced multimedia is relevant in a variety of scenarios (e.g., sports). However, to extract this value, its presentation must not be haphazard. It must be reasonably straightforward to find live streams of interest, even at scales of hundreds or thousands of simultaneous video streams. In this paper, we propose *FOCUS*, a system to enable an organized presentation, especially designed for live user-uploaded video. FOCUS automatically extracts contextual metadata, especially relating to the line-of-sight and subject in "focus," captured by a video feed. While this metadata can be used in various ways, our primary interest is to classify or *cluster* streams according to *similarity*, a notion of shared content. In this section, we will consider what it means for a pair of video streams to be judged as "similar," consider approaches and metrics for quantifying this understanding of similarity, and identify challenges and opportunities for extracting metric data on commodity smartphones.

2.1 Characterizing Video Content Similarity

While there can be several understandings of "video similarity," we will consider two videos streams to be more similar if, over a given period of time, a synchronized comparison of their constituent frames demonstrates greater "subject similarity." We judge two frames (images) to be similar depending on how exactly each captures the same physical object. Specifically, that object must be the subject, the focal *intent* of the videographer. By this definition, subject-similar clusters of live video streams can have several applications, depending on the domain. In sporting events, multiple videos from the same cluster could be used to capture disparate views of a contentious referee call, allowing viewers to choose the most amenable angle of view — enabling a crowdsourced "instant replay." For physical security, multiple views of the same subject can aid tracking of a suspicious person or lost child. For journalism, multiple views can be compared, vetting the integrity of an "iReport."

It is important to note that this definition of similarity says nothing of the perspective of the video (i.e., the location from where the video is captured), so long as the foreground subject is the same. We believe our definition of similarity, where the angle of view is not considered significant, is especially relevant in cases of a human subject. Naturally, two videos of a particular athlete share "similar" content, regardless of from which grandstand she is filmed. However, if these videos

Figure 1: Time-synchronized frames from four videos of an athlete on a stadium running track. Note that these frames are considered "similar," capturing the same athlete, but "look" heterogeneous.

are captured from a wide angular separation, they may "look" quite distinct. Contingent on the angle of separation, the visual structure and color of an object or person, lighting conditions (especially due to the position of the sun early or late in the day), as well as the background, may vary considerably. Perhaps counterintuitively, two "similar" views might actually look quite different (Figure 1). Our techniques must accommodate this diversity of view. Using the techniques we describe in Section 3, it would also be possible to judge a pair of videos shot from a more-nearby location as more similar. However, we see fewer immediate applications of this definition, and henceforth exclude it.

By our definition, videos which look heterogeneous may be judged similar, if they share the same subject. Further, videos which look homogenous may be judged dissimilar, if their subjects are physically different. For example, videos that capture different buildings, but look homogenous due to repetitive architectural style, should not be considered similar. Thus, a system to judge similarity must demonstrate a high certainty in deciding whether the object in a video's focus is truly the same precise subject in some other video.

Visual Metrics for Content Similarity. Understanding that two videos that "look" quite different might be judged "similar," and *vice versa*, several otherwise-reasonable techniques from vision are rendered less useful. Histograms of color content, spatiograms [10], and feature matching [33], are valuable for tracking an object across frames of a video though a "superficial" visual similarity. However, they are not intended to find similarity when comparing images that, fundamentally, may share little in common, visually. Though complementary to our approach, visual comparison is insufficient for our notion of subject-based similarity.

Leveraging Line-of-Sight. Our definition of similarity requires a precise identification of a shared subject (difficult). One possible proxy is to recognize that a pair of videos capture *some subject at the same physical location.* If we know that a pair of videos are looking towards the same location, at the same time, this strongly indicates that they are observing the same content. Precisely, we can consider the line-of-sight of a video, geometrically, a vector from the camera to the subject. More practically, we can consider the collinear infinite ray from the same point-of-origin and in the same direction. The geometric relationship of a pair of these rays reflects the

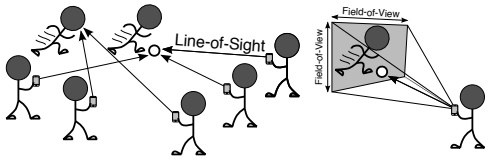


Figure 2: Illustrating line-of-sight: (a) users film two soccer players, ideally defining two clusters, one for each; (b) line-of-sight can be better understood as a 3D pyramid-shaped region, capturing the camera's horizontal and vertical *field-of-view* angles.

similarity of the corresponding videos, at the corresponding precise instant in time. A maximally-similar pair of views will have line-of-sight rays that perfectly intersect in 3D — the intersection point will be within the volume of their mutual subject (e.g., person or building). Line-of-sight rays which do not nearly intersect will not be similar. Figure 2 illustrates that, with multiple videos, intersecting line-of-sight rays suggest shared video content. Consistency across time reinforces the indication.

Strictly, and as illustrated by the left athlete in Figure 2(a), intersecting line-of-sight rays does not guarantee a shared subject. The principle subject of each video may appear in the foreground of (or behind) the intersection point. Thus, an inference of similarity by line-of-sight must be applied judiciously. In Section 3, we explain how FOCUS's similarity metric leverages vision to roughly estimate the termination point of a line-of-sight vector, substantially reducing the potential for *false positive* similarity judgments.

Our system, FOCUS, leverages multi-view stereo reconstructions and gyroscope-based dead-reckoning to construct 3D geometric equations for videos' lines-of-sight, in a shared coordinate system. More precisely, we will define four planes per video frame to bound an infinite, pyramid-shaped volume of space, illustrated in Figure 2(b). The angular separation between these planes corresponds to the camera's *field-of-view*, horizontally and vertically, and is distributed symmetrically across the line-of-sight ray. Geometric and computer vision calculations, considering how thoroughly and consistently these pyramid-shaped volumes intersect the same physical space, will form the basis for a content similarity metric.

To simplify, FOCUS understands the geometric properties of the content observed in a video frame, according to line-of-sight and field-of-view. FOCUS compares the geometric relationship between the content one video observes with that of others. If a pair of videos have a strong geometric overlap, indicating that they both capture the same subject, their content is judged to be "similar." Ultimately, groups of videos, sharing a common content, will be placed in self-similar groups, called *clusters*. Clusters are found through a technique called *weighted modularity maximization*, borrowed from *community identification* in complex networks. FOCUS finds "communities" of similar videos, derived from their geometric, or "spatial" relationship with time. Thus, uniquely, we say FOCUS groups live user videos streams based on a *spatiotemporal* metric of content similarity.

2.2 Opportunities to Estimate Line-of-Sight

With the availability of sensors on a modern smartphone, it may seem straightforward to estimate a video's line-of-sight:

GPS gives the initial position; compass provides orientation. Unfortunately, limited sensor quality and disruption from the environment (e.g., difficulty obtaining a high-precision GPS lock due to rain or cloud cover, presence of ferromagnetic material for compass) may make line-of-sight inferences too error-prone and unsuitable for video similarity analysis. Figure 3 illustrates this imprecision; lines of the same color should converge. Further, GPS is only useful outdoors — applications in indoor sporting arenas, shopping malls, and auditoriums would be excluded. Of course, a sensing-only approach can be valuable in some scenarios: outdoors when a reduced precision is tolerable. In Section 3.5, we use GPS, compass, and gyroscope for a lightweight clustering, formulated to minimize the impact of compass imprecision.

Smartphone sensing is, in general, insufficient for estimating a video's line-of-sight. The content of video itself, however, presents unique opportunities to extract detailed line-of-sight context. Using the well-understood *geometry of multiple views* from computer vision, it is possible to estimate the perspective from which an image has been captured. In principle, if some known reference content in the image is found, it is possible to compare the reference to how it appears in the image, deducing the perspective at which the reference has been observed. At the most basic level, how large or small the reference appears is suggestive of from how far away it has been captured. In the next section, we describe how our solution leverages *structure from motion*, a technique enabling analysis and inference of visual perspective, to reconstruct the geometry of a video line-of-sight.

Both smartphone sensing and computer vision provide complimentary and orthogonal approaches for estimating video line-of-sight. This paper seeks to blend the best aspects of both, providing high accuracy and indoor operation (by leveraging computer vision), taking practical efforts to reduce computational burden (exploiting gyroscope), and providing failover when video-based analysis is undesirable (using GPS/compass/gyroscope). We actualize our design next, incorporating this hybrid of vision/multimodal sensing.

3. ARCHITECTURE AND DESIGN

Fundamentally, an accurate analysis of content similarity across video streams must consider the video content itself — it most directly captures the intent of the videographer. Accordingly, reflecting the importance of visual inputs, we name our system *FOCUS* (*F*ast *O*ptical *C*lustering of *U*ser *S*treams). FOCUS leverages those visual inputs to precisely estimate a video stream's line-of-sight. Geometric and sensory metadata provides context to inform a spatiotemporal clustering, derived from community identification, to find groups of subject-similar videos. The FOCUS design, while willing to exercise substantial computation, is considered with a view towards real-world deployability, emphasizing (1) scalability, leveraging a cloud-based elastic architecture, and (2) computational shortcuts, blending computer vision with inertial sensing inputs into a hybrid analysis pipeline.

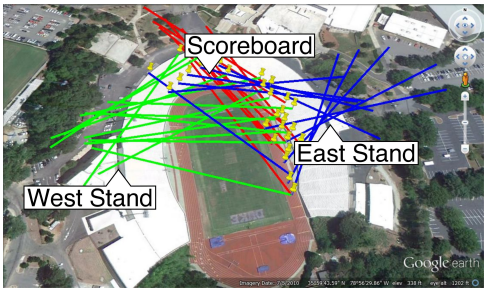


Figure 3: Google Earth view of a stadium with imprecise line-of-sight estimates from GPS/compass: (green) towards West Stands, left; (red) towards Scoreboard, top; (blue) towards East Stands, right.

Architectural Overview

The FOCUS architecture includes: (1) a mobile app (prototyped for Android) and (2) a distributed service, deployed on an infrastructure-as-a-service cloud using Hadoop. The app lets users record and upload live video streams annotated with time-synchronized sensor data, from GPS, compass, accelerometer, and gyroscope. The cloud service receives many annotated streams, analyzing each, leveraging computer vision and sensing to continuously model and track the video's line-of-sight. Across multiple streams, FOCUS reasons about relative line-of-sight/field-of-view, assigns pairwise similarity scores to inform clustering, and ultimately identifies groups of video streams with a common subject.

Figure 4 illustrates the overall flow of operations in the FOCUS architecture. We describe the key components in this section. We (1) describe computer vision and inertial sensing techniques for extracting an image's line-of-sight context; (2) consider metrics on that context for assigning a similarity score for a single pair of images; (3) present a clustering-based technique to operate on a two-dimensional matrix of similarity scores, incorporating spatial and temporal similarity, to identify self-similar groups of videos; (4) explain how computations on the Hadoop-on-cloud FOCUS prototype have been optimized for realtime operation; and (5) describe a lightweight, reduced accuracy sensing-only technique for line-of-sight estimation, for use in cases when computer vision analysis is undesirable or impractical.

3.1 Line-of-Sight from Vision and Gyroscope

By leveraging an advanced technique from computer vision, *Structure from Motion* (SfM), it is possible to *reconstruct* a 3D representation, or model, of a physical space. The model consists of many points, a *point cloud*, in 3D Euclidean space. It is possible to *align* an image (or video frame) to this model and deduce the image's *camera pose*, the point-of-origin location and angular orientation of line-of-sight, relative to the model. Multiple alignments to the same model infer line-of-sight rays in a single coordinate space, enabling an analysis of their relative geometry. As noted in Figure 4, FOCUS leverages Bundler [37], an open source software package for SfM, both for the initial model construction and later video frame-to-model alignment.

While the SfM technique is complex (though powerful and accurate), its usage is straightforward. Simply, one must take several photographs of a physical space (while a minimum of four is sufficient, efficacy tends to improve with a much larger number of photos). With these images as input, SfM operates in a pipeline: (1) extraction of the salient characteristics of a single image, (2) comparison of these characteristics across

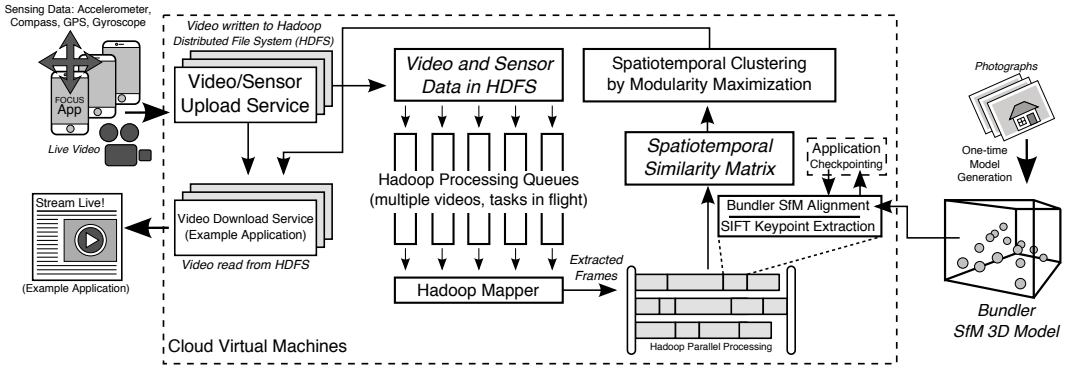


Figure 4: Overall FOCUS architecture. Note that video frame extraction, feature analysis, and alignment to an SfM model are parallelizable, enabling high analysis throughput from our Hadoop-on-cloud prototype.

images to find shared points of reference, and (3) an optimization on these reference points, constrained by the well-understood geometry of multiple views, into a reconstructed 3D point cloud. We assume that the SfM model will be generated and available in advance (perhaps by the operator of a sporting arena), to be used for analysis as live video feeds arrive at the FOCUS cloud service. In Section 5, we consider relaxing this assumption, using the content of incoming video feeds themselves for model generation.

Reconstructing 3D from 2D Images. For each image, a set of *keypoints* is found by computing a *feature extractor* heuristic [33]. Each keypoint is a 2D $< x, y >$ coordinate that locates a clear point of reference within an image — for example, the peak of a pitched roof or corner of a window. Ideally, the keypoint should be robust, appearing consistently in similar (but not necessarily identical) images. For each keypoint, a *feature descriptor* is also computed. A feature descriptor may be viewed as a "thumbprint" of the image, capturing its salient characteristics, located at a particular keypoint. We use the SIFT [33] extractor/descriptor.

Across multiple images of the same physical object, there should be shared keypoints with similar feature descriptor values. Thus, for the next stage of the SfM pipeline, we can perform a N^2 pairwise matching across images, by comparing the feature descriptors of their keypoints. Finally, the true SfM step can be run, performing a nonlinear optimization on these matched keypoints, according to the known properties of *perspective transformation* in a 3D Euclidean space. Once complete, the output is the 3D model in the form of a *point cloud*, consisting of a large number of $< x, y, z >$ points.

Figure 5: 3D reconstruction using Bundler SfM. 33K points from 47 photos of a university plaza, model post-processed for enhanced density/visual clarity.

Each 3D point corresponds to 2D keypoints extracted and matched from the original images.

Figure 5 shows the construction of a 33K-point model of a campus plaza from 47 high resolution photos. Figure 6 shows an (overhead view) 190K-point cloud generated from 412 photos of a 34K-seat collegiate football stadium. Note that model generation is feasible in both outdoor and indoor spaces, given sufficient light (not shown, we generated a 200-photo, 142K-point model of an indoor basketball arena).

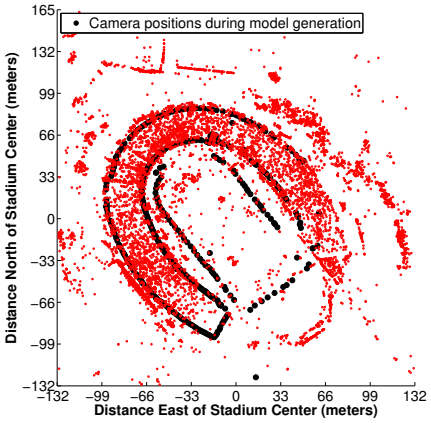


Figure 6: 3D reconstruction of a collegiate football stadium. Red dots show an overhead 2D projection of the 3D model. Black dots show locations from which photographs of the stadium were captured, systematically, around the top and bottom of the horseshoe-shaped grandstands and edge of the field.

Aligning a Frame to the Model: Estimating Pose. Once a model is constructed, it is possible to *align* a image (or video frame) taken in the same physical space. The alignment results in an estimate of its relative camera *pose*, a 3×1 *translation vector* and a 3×3 *rotational matrix* of orientation. The resulting 4×4 *rotation and translation matrix* can be used to construct the equation of a ray, with a point of origin at the camera, through the subject in the center of the view. This ray follows the line-of-sight from the camera, enabling similarity metrics based on view geometry. In Section 4, we evaluate SfM alignment performance against the stadium model construction shown in Figure 6. However, even prior to pursuing

this technique, it was important to validate that SfM-derived models are robust to transient environmental changes. For example, we generated our stadium model for photos captured during off hours. In Figure 7, we present an example image that aligns accurately to our model, taken during a well-attended football game. Despite occluded bleachers, alignment is still feasible as much of the core "structure" of the stadium (i.e., the rigid stands, buildings, and boundaries captured in the model) remains visible.

Augmenting Vision with Smartphone Sensing. Video frame-to-model alignment, while quick in FOCUS' scalable Hadoop-on-cloud pipeline (Section 3.4), is a heavyweight process. To reduce the computational burden, it is useful to combine SfM alignment with inputs from smartphone sensing. The inertial gyroscope, present on most new smartphones today, can provide a rotational "diff" across time, in the form of a rotation matrix. By matrix multiplication, FOCUS combines this gyroscope-derived rotational matrix with that of an SfM-estimated camera pose. We illustrate this process, akin to a rotational "dead reckoning," in Figure 8. Of course, errors will accumulate with time, due to inherent noise in the gyroscope sensor. FOCUS periodically re-runs SfM alignment, resetting this noise, and maintaining a bounded inaccuracy (relative to the last frame-to-model alignment). Moreover, since SfM alignment itself is prone to some error, this input from gyroscope can be used to inform hysteresis across multiple alignment attempts.

Unsurprisingly, video frame-to-model alignment can fail for several reasons: if the frame is blurred, poorly lit (too dark), captures sun glare (too bright), the extracted keypoints or feature descriptors have low correspondence with the model, or if the model is too sparse, self-similar, or does not capture the content of the to-be-aligned frame. In a video stream across time, these failures result in alignment "cavities" between successful alignments. To "fill" the cavities, and achieve a continuous alignment, gyroscope-based dead reckoning is especially useful. Note that dead reckoning is possible in either direction, forward or backward with time, from the nearest successful alignment. To dead reckon forward with time, the SfM-derived rotational orientation matrix is multiplied with a gyroscope-derived rotational matrix, accounting for the relative rotational motion accumulated over the time interval from the last alignment. To dead reckon in reverse, the gyroscope rotational matrix must first be inverted. Luckily, this inversion is trivial: as an invariant, the inverse of a rotation matrix is its transpose.

Figure 7: Challenging example photo that aligns accurately to our SfM model (Figure 6), despite capacity attendance (vs. empty during model capture).

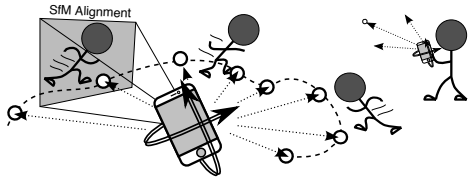


Figure 8: Illustrating rotational dead reckoning with gyroscope. As the user follows a moving target, gyroscope tracks a rotational matrix "diff" (forward or in reverse in time) from the closest SfM alignment.

Other smartphone sensors are also valuable during alignment. GPS, compass, and accelerometer, can be used to estimate a rough camera pose. While these estimates are prone to error, due substantial sources of noise in each sensor, they are valuable to "sanity check" outputs from SfM — immediately rejecting otherwise-silent alignment failures. In these cases, dead reckoning can be applied to overwrite what, otherwise, would be an erroneous alignment result.

3.2 Quantifying Spatial Content Similarity

To cluster video feeds into self-similar groups, FOCUS will assume a metric to quantify the logical content "similarity." Pairs of videos with a high mutual similarity are likely to be placed into the same cluster. As an invariant, each video will be placed in the cluster with which it has the greatest spatial (from line-of-sight) content similarity, averaged across time, averaged across all other cluster members. In this subsection, we will present the design of FOCUS' spatial similarity metric for a pair of video frames. FOCUS' metric was not successfully designed "at once;" instead its techniques were evolved and refined through system experimentation.

3.2.1 By Line-of-Sight Intersection (Failed Attempt)

FOCUS leverages SfM and gyroscopic tracking to estimate a 3D rays of camera pose — originating from the camera and along the line-of-sight. For a pair of frames capturing the same object of interest, the corresponding rays should intersect, or nearly intersect, through the mutual object in view. One possible similarity metric is to consider the shortest distance between these two rays. The resulting 3D line segment must be either (1) between the two points of origin, (2) from the point of origin of one ray to a perpendicular intersection on the other, (3) perpendicular to both rays, or (4) of zero length. We may treat cases (1) and (2) as having no view similarity; line-of-sight rays diverge. In cases (3) and (4), shorter line segments reflect a nearer intersection, and suggest a greater view similarity. Assuming that the constructed rays are accurate, this metric is not subject to *false negatives*; for any pair of videos sharing the same content, the length of the line segment between the rays must be small. Unfortunately, this simple metric is not foolproof. *False positive* indications of similarity may result; the intended subject may fall in front or behind the point of ray intersection. For example, if stadium spectators in the grandstand focus on different players on the field, this metric will return that the views are similar if the corresponding rays intersect "underground."

3.2.2 By SfM "Point Cloud" Volumetric Overlap

During early experimentation, we found that the above ray-intersection metric, while intuitively appealing in its simplicity, is overly susceptible to false positives. We could eliminate the potential for false positives by replacing each camera pose ray with a vector, terminating at the object in view. While this is difficult to estimate, we can leverage context from the 3D model structure to terminate the vector roughly "on" the model, for example, capturing the ground below the subject. Looking down from a stadium grandstand, subterranean intersections would be eliminated. Similarity, intersections in the air, above the field, can be ignored.

Recall that the SfM model is a point cloud of $< x, y, z >$ coordinates, capturing rigid structures. Instead of only considering the geometry of a line-of-sight ray, we may identify structures captured by a video frame. For a pair of videos, we can compare if both capture the same structures. Several techniques from vision apply here. For example, 3D *point cloud registration* heuristics exist for estimating boundaries of a mesh surface, and approximating structures. However, as a simpler, computationally-tractable alternative, we may count the model points mutually visible in a pair of video frames. More shared points suggest greater similarity in their views. In the next subsection, we discuss how high similarity values, filling an $N \times N$ spatial similarity matrix, encourage placement of these videos in the same cluster.

To count the number of common points in the intersecting field-of-views of two videos, we must first isolate the set of points visible in each. As we describe next, the set can be found by considering the pyramid-shaped field-of-view volume, originating from the camera and expanding with distance into the model. Later, we can quickly count the number of shared points across multiple video frames by applying a quick set intersection. Simply, we construct a bitmap with each bit representing the presence of one point in the view.[1] The number of shared points can be found by counting the number of bits set to 1 in the bitwise AND of two bitmaps.

3.2.3 Finding Model Points in a Video View

Let a single estimated line-of-sight be expressed as $L(R, t)$. R represents a 3×3 rotation matrix, the 3D angle of orientation. t represents a 3×1 vector of translation. $-R^{-1}t$ defines the $< x, y, z >$ camera position coordinate, the location in the model from where the video frame was captured. R can be further decomposed as three row vectors, known respectively as RIGHT, UP, and OUT, from the perspective of the camera. To capture the camera's view of the model, we form a pyramid emerging from the camera position ($-R^{-1}t$) and extending in the direction of OUT vector. The four triangular sides of the pyramid are separated, horizontally and vertically, according to the camera's field-of-view (Figure 2).

The pyramid-shaped camera view can be abstracted as four planes, all intersecting at the camera position coordinate. Now, to fully describe equations for these planes, we must only find a plane normal vector for each. In order to find four plane normals, we rotate the OUT vector along the RIGHT and UP vectors, so that the transformed OUT vector becomes perpendicular to one of these planes. Rotation of any 3D vector, along a unit-length 3D vector, is given by Rodrigues' rotation formula. Using this equation, we rotate the OUT vector along the RIGHT vector by angle $\pm(\pi/2 - vAngle/2)$ to estimate normals for two planes (top/bottom). Similarly, rotations along the UP vector with angle $\pm(\pi/2 - hAngle/2)$ results in normals to left and right planes. Here, $vAngle$ and $hAngles$ are taken as parameters for the smartphone camera's field-of-view angle, horizontally and vertically. We test the signs from these four planar equations for each point in the model, determining the set of points potentially visible from a particular video frame. Later, we perform set intersections to estimate similarity between the N^2 pairs of time-synchronized frames of N videos. This $N \times N$ value table completes our notion of a *spatial similarity matrix*.

[1] Alternatively, Bloom Filters are suitable as probabilistic substitutes for direct point-to-bit maps, for space-saving bitmaps.

3.3 Clustering by Modularity Maximization

So far, we have discussed what it means for a pair of video frames to be judged "similar," especially by the intersection of their respective line-of-sight and field-of-view with an SfM-derived 3D model. This notion of "similarity" is a static judgment, based on an instantaneous point in time. In reality, we are interested in the similarity of a pair of videos, across multiple frames, for some synchronized time interval. This requires a further understanding of what it means for a pair of videos to be "similar," above and beyond the similarity of their constituent frames. We will assume that a pair of "similar" video streams need not both track the same spot consistently. Instead, it is only required that they should both move in a correlated way, consistently capturing the same physical subject, at the same time. Simply, both streams should maintain (instantaneous) similarity with each other across time, but not necessarily have self-similarity from beginning to end. This seems reasonable in the case of a soccer game: some videos will follow the ball, some will follow a favored player, and others will capture the excitement of the crowd or changes to the scoreboard. These "logical clusters" should map as neatly as possible to FOCUS' groupings.

Spatiotemporal Similarity. To capture the mutual correspondence in a set of N videos with time, we apply our notion of an $N \times N$ spatial similarity matrix across T points in time. For every instant $t \in T$ in a synchronized time interval, we find the corresponding spatial matrix S_t and apply clustering, finding some set of groupings G_t from line-of-sight and field-of-view at time t. Next, we aggregate these spatial results into an $M = N \times N$ *spatialtemporal* similarity matrix. Let $\delta_g(i, j) = 1$ if streams i and j are both placed into the same spatial cluster $g \in G_t$. $\delta_g(i, j) = 0$, otherwise.

$$M_{ij} = \sum_{\forall t \in T} \sum_{\forall g \in G_t} \delta_g(i, j)$$

Finally, we apply clustering again, on M, providing groups of videos matching our notion of *spatiotemporal* similarity. Next, we elaborate on our choice of clustering heuristics.

The "Right" (and Right Number) of Clusters. Several clustering approaches require some parameterization of how many clusters are desired (e.g., the k value in k-means clustering). By comparison, community identification via *modularity maximization* has the appealing property that community boundaries are a function of their *modularity*, that is, a

mathematical measure of network division. A network with high modularity implies that it has high correlation among the members of a cluster and minor correlation with the members of other clusters. For FOCUS, we apply a weighted modularity maximization algorithm [13]. As input, FOCUS provides an $N \times N$ matrix of "similarity" weights — either that of spatial or spatiotemporal similarity values. Modularity maximization returns a set of clusters, each a group of videos, matching our notions of content similarity.

3.4 Optimizing for Realtime Operation

A key motivation for FOCUS is to provide content analysis for streaming realtime content *in realtime*. Excess computational latency cannot be tolerated as with it increases (1) the delay before content consumers may be presented with clustered videos and (2) the monetary costs of deploying the FOCUS Hadoop-on-cloud prototype. Here, FOCUS makes two contributions. First, as previously discussed, FOCUS leverages gyroscope-based dead reckoning as a lightweight proxy to fill gaps between SfM camera pose reconstruction, reducing the frequency of heavyweight computer vision (alignment of an image to an SfM-derived model) to once in 30 seconds. Second, FOCUS applies *application checkpointing* to combat startup latency for SfM alignment tasks.

As described in Section 3.1, FOCUS uses Bundler to align an image (estimate camera pose) relative to a precomputed 3D model. Bundler takes approximately 10 minutes to load the original model into memory before it can initiate the relatively quick alignment optimization process. To avoid this latency, FOCUS uses BLCR [15] to *checkpoint* the Bundler application (process) state to disk, just prior to image alignment. The process is later *restarted* with almost zero latency, each time substituting the appropriate image for alignment.

FOCUS Hadoop Prototype Cluster. FOCUS exists as a set of cloud virtual machine instances configured with Apache Hadoop for MapReduce processing. FOCUS informs virtual machine elastic cloud scale-up/down behavior using the Hadoop queue size (prototype FOCUS elasticity manager currently in development). For FOCUS, there are several types of MapReduce task: (1) base video processing, to include decoding a live video feed and sampling frames for further image-based processing; (2) image feature extraction, computation of feature descriptors for each keypoint, alignment to an SfM model, and output of a bitmap enumerating the set of visible model points; (3) pairwise image feature matching, used when building an initial 3D SfM model; and (4) clustering of similar video feeds. Tasks of multiple types may be active simultaneously.

3.5 Failover to Sensing-only Analysis

In certain circumstances, it may be undesirable or infeasible to use SfM-based line-of-sight estimation. For example, in an "iReport" scenario, video may be captured and shared from locations where no SfM model has been previously built. Further, users may choose to upload video only if very few peers are capturing the same video subject — saving battery life and bandwidth for the user. A lightweight clustering technique, without requiring upload of the video stream, could be used to pre-filter uploads of redundant streams. FOCUS provides a sensing-only alternative (using GPS and compass), clustering streams without requiring computer vision processing or even access to video sources. Through iterative design and testing, we have refined our technique to be relatively insensitive to compass error. By considering the wide camera field-of-view angle in the direction of line-of-sight, our metric is not substantially impacted by compass errors of comparable angular size.

For each latitude/longitude/compass tuple, FOCUS converts the latitude/longitude coordinates to the rectangular Universal Transverse Mercator (UTM) coordinate system, taking the EASTING and NORTHING value as an $< x, y >$ camera coordinate. From the camera, the compass angle is projected to find a 2D line-of-sight ray. Next, two additional rays are constructed, symmetric to and in the same direction as the line of sight ray, and separated by the horizontal camera field-of-view ($hAngle$). This construction can be visualized as a triangle emerging from the GPS location of a camera and expanding outward to infinity (with an angle equal to the camera's horizontal field-of-view). A metric for view similarity is computed as the area bounded by intersecting two such regions. Since this area can be infinite, we impose an additional bounding box constraint. The resulting metric values are used to populate the spatiotemporal similarity matrix. Clustering proceeds as for SfM-based similarity. To reduce the potential for compass error, gyroscope informs a hysteresis across multiple compass line-of-sight estimates.

To compute the area of intersection (and thus our metric), we find the intersection of the constraining rays with each other and with the bounding box, forming the vertices of a *simple* (not-self-intersecting) polygon. We may order the vertices according to *positive orientation* (clockwise) by conversion to polar coordinates and sorting by angle. Next, the polygon area is found by applying the "Surveyor's Formula."

4. EVALUATION

Through controlled experimentation[2] *in two realistic scenarios and comparison of FOCUS' output with human efforts,*[3] *we endeavor to answer the following key questions:*

1. How accurate is line-of-sight for identifying unique subject locations? *(Figs. 10, 12)* Indoors? *(Fig. 14)* For objects only a few meters apart? *(Figs. 11b, 14b)*
2. How does GPS/compass-based line-of-sight estimation compare with SfM/gyroscope? *(Figures 3, 11, 17)*
3. When video streams are misclassified, are incorrectly clustered videos placed in a reasonable alternative? Are SfM processing errors silent or overt (enabling our gyroscope-based hysteresis/failover)? *(Figs. 12, 13)*
4. It our spatiotemporal similarity matrix construction robust to videos with dynamic, moving content, tracking spatially-diverse subjects with time? *(Figure 15)*
5. What is the latency of vision-based analysis? *(Fig. 16)*
6. Is FOCUS' sensing-only failover approach tolerant to large compass errors? Can GPS/compass provide a reasonable accuracy when SfM/gyroscope is undesirable, infeasible, or as temporary failover? *(Figures 3, 17)*

[2]Constrained by (a) privacy for video-recording human subjects and (b) copyright ownership for NCAA athletic events.
[3]The procedures for this study were vetted and approved in advance by our institution's ethics and legal review board.

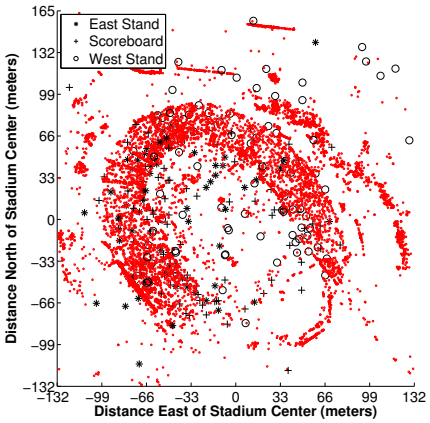


Figure 9: Experimental locations in/around the stadium. Symbols denote each video's focal subject: (∗) East Stand; (+) Scoreboard; and (○) West Stand.

7. How do FOCUS's clusters compare with those created by human volunteers? *(Figure 20)*

Methodology. Our FOCUS evaluation takes the perspective of a likely use case: user video streams in a collegiate football stadium and an indoor basketball arena. With 33K seats and a maximum attendance of 56K, by sheer numbers, it is likely that multiple (even many) visitors to our stadium would choose to stream video simultaneously. To exercise FOCUS across dynamic lighting by time of day, a range of cloud cover, variations in attendance, a span of videographer skill, and transient occlusions,[4] we collected short video clips from the stadium over a period of two months.

To build the stadium SfM model, containing 190K points and shown in Figure 6, we took 412 2896x1944 resolution photographs with a Nikon D3000 SLR camera. All videos were taken in 1920x1088 resolution at 15 FPS, using our Android app, on a Samsung Galaxy Nexus or Galaxy S3. In line-of-sight estimation results, short clips (20-30 seconds each) were used. For longer motion tracking results, videographers were asked to keep the running "athlete" centered in view as consistently as possible, over periods of minutes.

Line-of-sight Estimation Accuracy. Using our app, three large, preselected points-of-interest in the stadium were filmed: (1) bleachers where the "pep" band sits, centered in the the EAST STAND; (2) the press box above mid-field in the WEST STAND; and (3) the SCOREBOARD.[5] Videographers walked around arbitrarily, capturing each of the intended subjects from 325 locations (shown in Figure 9, with locations from GPS, marked by intended subject). Figures 10 (a,b,c) show visualizations of SfM/gyroscope-based line-of-sight estimation accuracy, for each subject. Dark lines show estimated line-of-sight. Rough convergence through the designated subject in the EAST STAND, SCOREBOARD, or WEST STAND, respectively, visually suggests that typically SfM frame-to-model alignment is highly accurate. Figure 11 (a,b) plot CDFs, confirming this accuracy. Further, 11 (a,c) confirm the inferior accuracy of line-of-sight estimation with GPS/compass (only). Note that in both Figures 10 and 11, a substantial portion of angular "error" is attributable to inaccuracy in filming a designated subjects. Visible outliers in Figure 10 (b) are attributable to poor SfM alignment, typically due to difficult viewing angles (e.g,. closeups).

[4]For example, an opaque protective tarp not captured by our SfM model was typically present, covering the entire field.

[5]Large subject areas chosen to reduce impact of human filming error. *Figure 11(b) confirms applicability to small subjects.*

Spatial Clustering Accuracy. Figure 12 summarizes the accuracy of FOCUS spatial clustering using SfM/gyroscope in a challenging scenario: clustering on 325 video streams simultaneously (using sequentially-collected 20-30 second video clips as a proxy for a large deployment), from the diverse set of locations shown in Figure 9. FOCUS placed each "stream" (video clip) in one of three spatial clusters, or marked the stream as a processing failure (no SfM alignment, e.g., due to blur). For every assigned member of every spatial cluster, we compared the video's intended subject to the geographic centroid of each output cluster. If the assigned cluster was the closest (geographically) to the intended subject, it was considered a "true positive" result (placed in the most correct cluster). Otherwise, it was considered both a "false negative" for the intended subject and "false positive" for its actual placement. Note that we consider false positives/negatives less desirable than processing failures, as they represent a silent failure. Sanity checking (by GPS, compass, and accelerometer) was not applied. Further, though gyroscope-based hysteresis was used in this experiment, correcting several poor SfM alignments, the short video

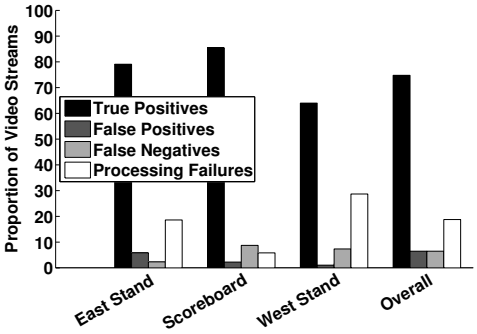


Figure 12: Stadium spatial clustering accuracy. For each True Positive, FOCUS correctly aligned a video clip and placed it into a cluster with others predominantly capturing the same intended subject.

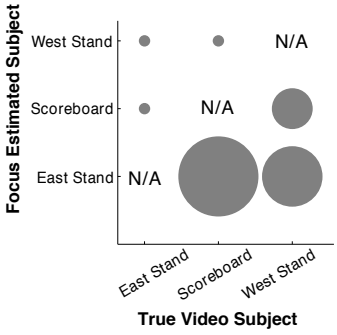


Figure 13: Stadium spatial clustering confusion matrix. For each intended subject along the X axis, the size of the circle reflects the proportion of video clips misplaced in the corresponding Y-axis cluster.

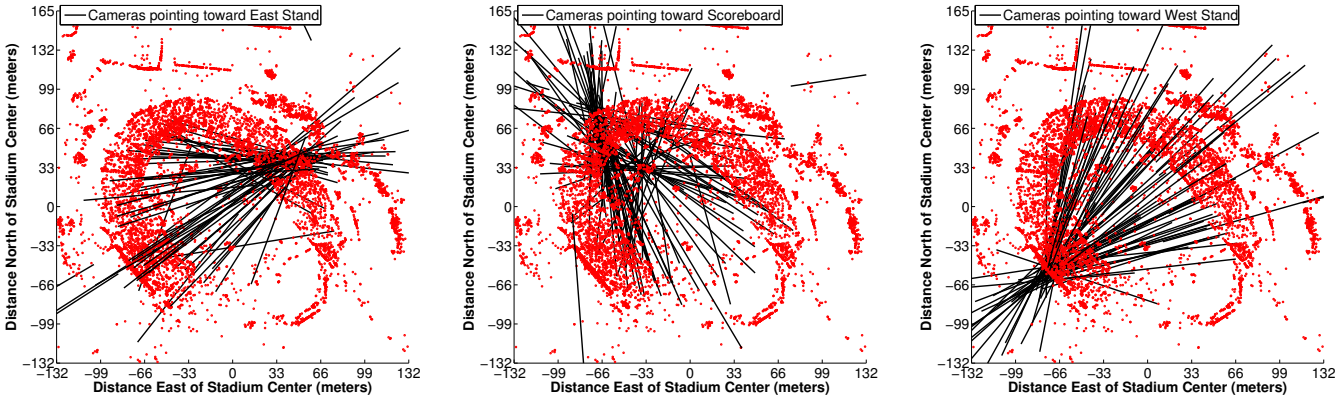


Figure 10: Estimated line-of-sight rays on stadium model with true focal subject: (a) in the East Stand; (b) on the Scoreboard; and (c) in the West Stand. Converging lines reflect precision of SfM line-of-sight estimation.

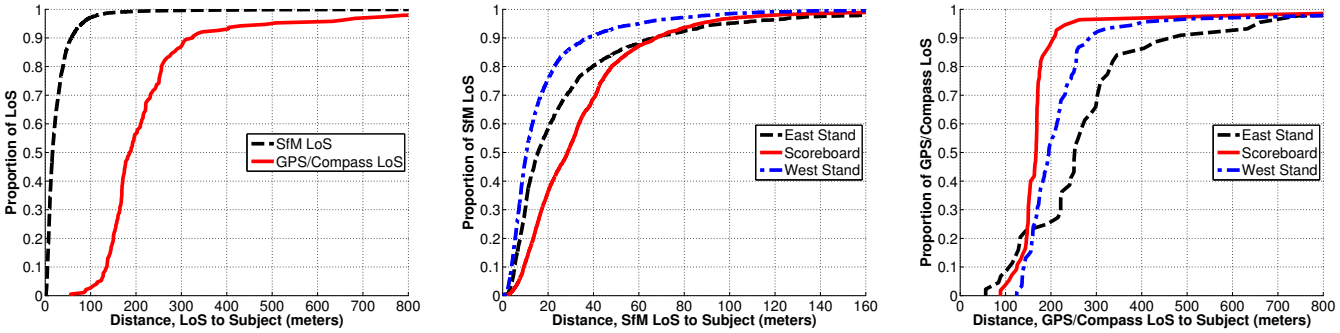


Figure 11: CDFs of alignment "error." Y-axis shows the shortest distance from the estimated line-of-sight ray to the intended subject: (a) SfM/gyroscope versus GPS/compass overall; (b) SfM/gyroscope by subject; (c) GPS/compass by subject. Note that a nontrivial error proportion is attributable to videographer imprecision.

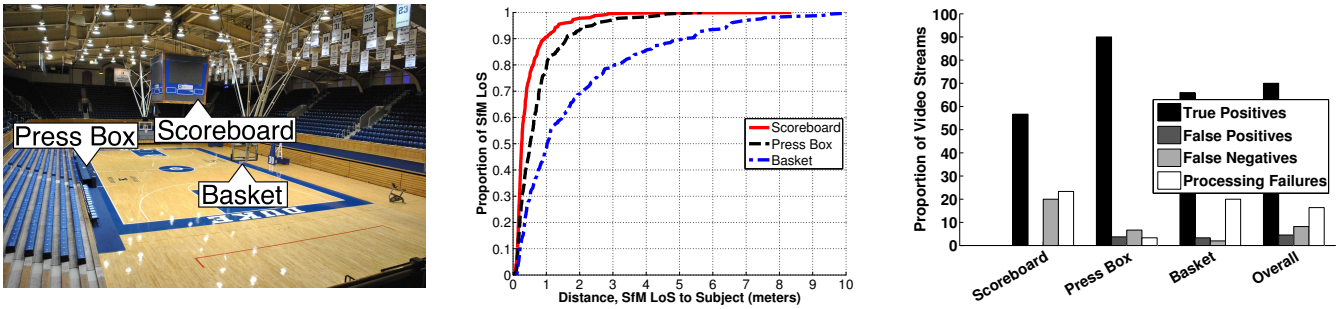


Figure 14: Indoor performance in a collegiate basketball arena: (a) example image from SfM model generation; (b) SfM/gyroscope line-of-sight estimation accuracy (110 locations); (c) spatial clustering accuracy.

clip time interval limited opportunities for dead reckoning. Figure 13 presents a confusion matrix showing how false positive/negative cluster assignments were distributed. As expected, nearer subjects were more frequently confused. Overall results achieve 75% "true positive" success. 19% are processing failures, in which case FOCUS would invoke GPS/compass failover (we disabled FOCUS' failover for this experiment to restrict results to SfM/gyroscope exclusively).

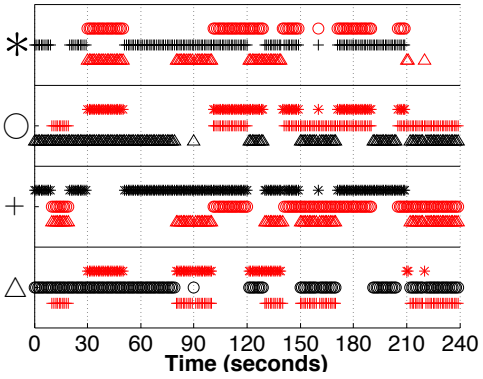
Indoor, Low-light Performance. We tested in a 10K-seat basketball arena. Strong performance in this challenging indoor environment, where GPS/compass-based techniques do not apply and under relatively low light, demonstrates generic applicability across environments. Figure 14 shows: (a) an example photo from our SfM model; (b) line-of-sight estimation accuracy; and (c) spatial clustering accuracy.

Spatiotemporal Clustering. Figure 15 presents a dynamic scenario for FOCUS, tracking four video streams across a period of minutes. Each stream is tracking one of two "athletes" (fit volunteers) on the stadium running track. The athletes ran around the track in opposite directions (clockwise and counterclockwise), crossing paths multiple times. In the figure, each stream is represented by a marker ($\triangle$, $\circ$, $+$, $*$). Each grouping along the Y-axis denotes a particular video stream. Along the X-axis, we show markers at every second of a four-minute video. A dark (black) marker denotes a true positive result: a stream, for the corresponding one-second interval,

was placed in the same spatial cluster as was the true videographer intent (same as that of symbol along Y-axis). A light (red) marker denotes a false positive result: a stream, for the corresponding one-second interval, was placed in the same spatial cluster as that of symbol along the Y-axis, but contradictory to to the true videographer intent. The figure illustrates that, typically, FOCUS is able to place a pair of videos with matching content into a matching cluster. It also captures all the data required to construct the spatiotemporal matrix of content similarity for these clips. The final, overall spatiotemporal clustering successfully outputs the correct stream-to-athlete matching ($\triangle$ and $\circ$ grouped for athlete A subject, $+$ and $*$ grouped for athlete B subject). This result was successful even though stream $\triangle$ suffered an extended 80-second period during which SfM alignment failed (due to deficiencies in our stadium model), leveraging our gyroscope-based dead-reckoning for seamless tracking.

Computer Vision Computational Latency. FOCUS' end-to-end performance is largely a function of vision processing, especially the dominating subtask: alignment of extracted features to a precomputed SfM model. Figure 16 plots single-threaded performance. In ongoing work, we are investigating ways to further expedite processing leveraging state-of-the-art techniques for fast alignment [14, 28, 41].

Tolerance of Sensing-only Failover to Compass Error. FOCUS' sensing-only failover (GPS/compass-based line-of-sight estimation) must be tolerant to substantial compass error. We tested failover by introducing random Gaussian errors to ground truth compass angles (GPS locations were used unmodified from experimental locations). Figure 17 shows diminishing clustering accuracy with extreme compass error, but solid performance with angular error having a standard deviation of 20 degrees, clustering at 85% accuracy.



Video	$\triangle$	$+$	$\circ$	$*$
$\triangle$	-	88	151	63
$+$		-	115	149
$\circ$			-	86
$*$				-

Figure 15: (a) Y-axis shows four videos, ($\triangle$) and ($\circ$) capture athlete A, ($+$) and ($*$) capture B. Markers show streams placed into the same cluster, by time. Dark markers are matching subject. (b) Spatiotemporal matrix M: M_{ij} denotes the number of spatial clusters where stream i and j were mutually placed.

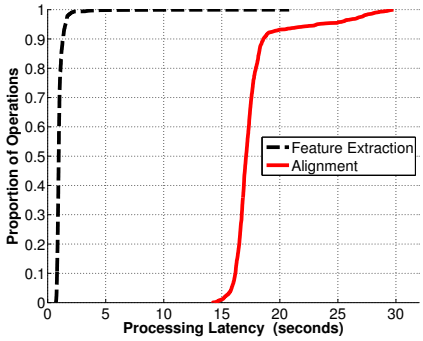


Figure 16: CDF of processing latency: video frame feature extraction, SfM frame-to-model alignment.

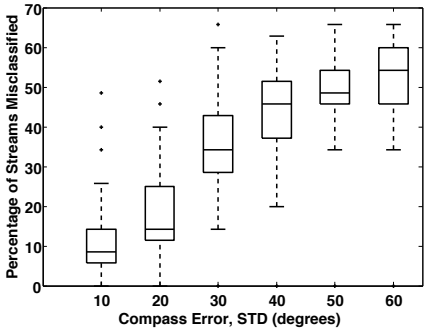


Figure 17: Box-and-whisker plots showing performance of GPS/compass-only failover by compass accuracy. X-axis shows standard deviation of introduced Gaussian compass errors (angles in degrees).

User Study: Comparison to Human-created Clusters. We recruited 70 volunteers (demographics in Figure 18) to compare FOCUS' clusters to human groupings. Participants manually formed clusters from 10 randomly-selected videos (either from the football stadium or basketball arena datasets) by "dragging-and-dropping" them into bins, as shown by the screenshot in Figure 19. We adapt metrics from information retrieval to quantify the correctness of both human and FOCUS-created clusters: *precision*, *recall*, and *fallout*, using predefined labels of videographer intent.

For understanding, precision roughly captures how consistently a volunteer or FOCUS is able to create groupings where each member of the group is a video of the same intended subject as all other members. Recall captures how completely a group includes all videos of the same intended subject. Fallout captures how often a video is placed in the same group as another video that does not capture the same intended subject (lower values are better). More precisely:

Let $V = \{v_1, v_2, \ldots, v_n\}$ be a set of videos under test. Let $C(v_i, v_j) = 1$ if v_i and v_j are placed in the same cluster, 0 otherwise. Let $G(v_i, v_j) = 1$ if v_i and v_j *should* be placed in the same cluster, according to ground truth, 0 otherwise.

$$\text{PRECISION} = \frac{|\{\forall v_i, v_j \in V \text{ s.t. } G(v_i, v_j) \wedge C(v_i, v_j)\}|}{|\{\forall v_i, v_j \in V \text{ s.t. } C(v_i, v_j)\}|}$$

$$\text{RECALL} = \frac{|\{\forall v_i, v_j \in V \text{ s.t. } G(v_i, v_j) \wedge C(v_i, v_j)\}|}{|\{\forall v_i, v_j \in V \text{ s.t. } G(v_i, v_j)\}|}$$

$$\text{FALLOUT} = \frac{|\{\forall v_i, v_j \in V \text{ s.t. } \neg G(v_i, v_j) \wedge C(v_i, v_j)\}|}{|\{\forall v_i, v_j \in V \text{ s.t. } \neg G(v_i, v_j)\}|}$$

As shown by Figure 20, FOCUS' precision, recall, and fallout percentages compare favorably to that of volunteers. Note that the performance of FOCUS occasionally (though not typically) exceeds that of our volunteers. We found that for humans, as well as for FOCUS, videographer intent is subjective and can be ambiguous to a third party. In some (randomized) cases, the volunteers were asked to cluster images which look, subjectively, quite similar but actually capture different physical locations. FOCUS was able to find small details to distinguish the locations that were not obvious to our volunteers. We observed a few trial participants and found them surprised when we debriefed them of errors in their clustering choices.

5. LIMITATIONS AND DISCUSSION

Difficult Scenarios for Structure from Motion. FOCUS is designed to work in locations where it is feasible to visually reconstruct a sound 3D model of the physical space. Reconstruction efficacy is subject to the properties of the feature detection algorithm, algorithms to identify edges or corners of a rigid structure. Stadiums and open areas between buildings, for example, are compliant environments as they contain large rigid structures, likely to produce many consistent keypoints across multiple images. Even when filled with spectators, the rigid structure of a stadium grandstand is still preserved (and thus are so many keypoints in the SfM model, see Figure 7). However, in other contexts, environmental dynamism may hide rigid structures, such as in a parade with large floats. Further, open fields, areas heavily occluded with trees, and tight indoor spaces will present a challenge to SfM, yielding poor results with FOCUS. We have not systematically explored the performance of FOCUS under heavy occlusions. In such cases, we assume SfM to fail and expect to failover to sensing-only techniques. Unsurprisingly, the efficacy of SfM is also dependent on lighting conditions. Dimly lit environments and outdoor environments at dawn or dusk yielding sun glare are especially challenging. For all such extreme cases, overall accuracy will roughly equate to sensing-only performance, as SfM alignment rarely results in silent failures.

Average Reported Age	45
% Male / % Female	79% / 21%
% own a smartphone	82%
% use smartphone for taking photos	88%
% use smartphone for taking video	60%
% share multimedia on social networks	41%

Figure 18: User study volunteer demographics.

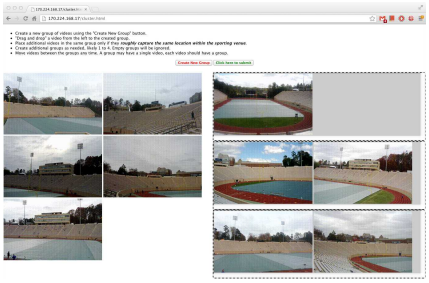


Figure 19: Screenshot from the user study interface. Volunteers "drag-and-drop" 10 animated GIF images (sampled from short video clips) from the left to define one or more "clusters" with boxes on the right.

Optimized Frame Selection for Alignment. Predictably, not all video frames are equally amenable to SfM alignment. Leveraging sensory inputs or lightweight computer vision, it is possible to identify which video frames are likely to be the "best" for further processing. For example, any of accelerometer, gyroscope, or visual inputs can be applied to identify shaken and blurred views. Similarly, simple visual metrics, such a color histogram or spatiograms [10], would be useful in detecting a change in the presence of occlusions. Gyroscope-based dead reckoning is again useful here, making it easy to select, align, and leverage a compliant frame.

Reconstructing 3D Models "on the Fly". In our evaluation, we have considered environments where it is feasible to precompute the SfM 3D model. In a stadium for professional or collegiate sports, for example, we imagine an employee or fan taking photographs of the arena, in advance, as input to our Hadoop-based model generation pipeline. However, with a goal to streamline practical deployment of the FOCUS system, we believe it is also possible to construct the model dynamically, using the content of the video streams themselves. Fundamentally, SfM is able to build a model by leveraging a diversity of perspective across multiple views of the same object: normally achieved by taking photos while moving through a space. A similar diversity is inherently available across the various feeds coming into system.[6] During the bootstrap period, during which sufficient images are gathered and the model is constructed, FOCUS would leverage its sensing-only failover technique, operating at a reduced accuracy until the model is complete. We tested model generation from streams as part of our indoor arena evaluation — from 60 video clips, we reconstructed an SfM model of comparable quality to that generated from still images.

Applications. We exclude a thorough treatment of how one might leverage (e.g., query) FOCUS' output video clusters. Section 4 evaluates in the context of sports, but many novel presentations are possible. Rather than domain-specific, our target is a generic construct to extract "app-enabling" metadata: logical pointers to videos capturing the same subject.

6. RELATED WORK

Structure from Motion and Related. FOCUS applies structure from motion (SfM) to generate 3D models from multiple images, leveraging Bundler [37]. [8] uses Bundler to develop a large-scale 3D model of Rome from a corpus of photographs available on image sharing websites. [17,18,42] generate SfM models from continuous video sequences. Such models can be directly used by FOCUS. [14, 28, 41] propose various methods for fast alignment of an image into an existing SfM 3D model using motion patterns and inertial sensing. These techniques are complementary to FOCUS, improving the speed of line-of-sight estimation. [21] uses LED markers to estimate camera pose. [38] develops an iPhone photography game for city-scale SfM. [29] applies SfM and sensing to estimate the location of a remote object.

Visual Similarity Analysis and Tracking. Video content matching and grouping is not unique to FOCUS, especially leveraging computer vision. [20] constructs graphs to connect related imagery in a large collection. [24] considers the

[6]Historical videos can also be used (e.g., from YouTube).

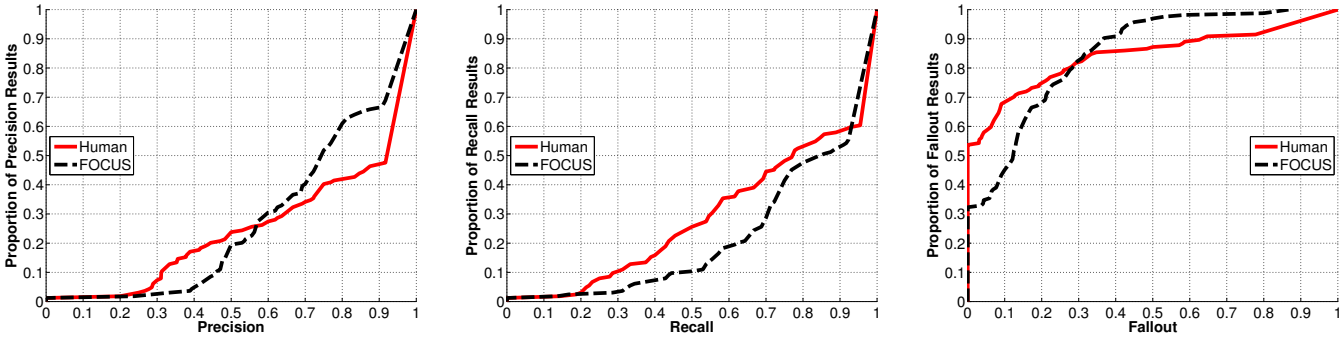


Figure 20: CDFs comparing FOCUS accuracy versus ground truth to human-generated clusters from the user study. FOCUS' clustering accuracy is comparable to that of volunteers; (a) Precision; (b) Recall; (c) Fallout.

use of video sensors for cooperative tracking of surveilled subjects. [16, 30] find similar actions in sports games, identifying videos and the time of action. [25, 26, 34] apply image processing such as background subtraction, compositing, and increasing dynamic range to identify similar content in video and images. [19, 40] visually track a shared subject from multiple cameras in complex situations. [36] enables geo-referenced video search, similar in spirit to our GPS/Compass sensing-only failover approach.

Crowdsourcing and Localization. The power of crowdsourcing is widely confirmed. The ubiquity of mobile phones is helping to enable such systems on a large scale. [9, 11, 27, 31] propose various methods on collaborative and continuous sensing using smartphones, enabling multimedia tagging and organization by mining over contextual cues generated from inertial sensors and image processing. [32, 39] use crowdsourcing for image search over large collections of photographs. [22, 23, 29] apply sensing to localization.

Sports and Industries. [2] deploys/monitors laser cameras on commercial stadiums for tracking player movements. [35] is a connects collocated TV viewers to enhance a social viewing experience. [7,12] personalize sports multimedia feeds for different viewers. [3–6] enable users to create and share collaborative video experiences. [1] considers GPS and compass in image metadata to organize content from related angles.

7. CONCLUSION

The value of user-uploaded video is both immense and fragile. YouTube and other sites depend on a haphazard collection of manual tags and machine-mineable comments. Real-time content, prior to the availability of this crowdsourced context, is difficult to index. With the trends towards enhanced wireless data connectivity, improved smartphone battery life, and adoption of the cloud for low-cost, scalable computation, we envisage widespread distribution of user-uploaded real-time video streams from mobile phones. FOCUS is a system to analyze this live content, in realtime, finding groups of video streams with a shared subject. As an immediate high-value target, we have thoroughly evaluated FOCUS in a collegiate stadium environment. Once fully hardened, FOCUS can be deployed, for example, to enable a crowdsourced "instant replay," enabling the viewer to inspect multiple angles of a contentious play. More generally, we believe that FOCUS is broadly enabling for a variety of next-generation streaming multimedia applications. Nonetheless, FOCUS is a prototype, and a work-in-progress. Finally, we see further research opportunities in advancing a cloud platform for additional automated analyeses and distribution of live user video streams.

8. ACKNOWLEDGMENTS

We sincerely thank our many volunteers, David Chu our shepherd, as well the anonymous reviewers.

9. REFERENCES

[1] Crowdoptic. http://www.crowdoptic.com/.
[2] Exa-tech. http://www.exa-tech.com/.
[3] Streamweaver. http://streamweaver.com/.
[4] Stringwire. http://www.stringwire.com/.
[5] Switchcam. http://www.switchcam.com.
[6] Vyclone. http://vyclone.com/.
[7] Yinzcam. http://www.yinzcam.com/.
[8] S. Agarwal, N. Snavely, I. Simon, S. Seitz, and R. Szeliski. Building rome in a day. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 72–79. IEEE, 2009.
[9] X. Bao and R. Roy Choudhury. Movi: mobile phone based video highlights via collaborative sensing. In *Proceedings of the 8th international conference on Mobile systems, applications, and services*, pages 357–370. ACM, 2010.
[10] S. Birchfield and S. Rangarajan. Spatiograms versus histograms for region-based tracking. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 1158–1163. IEEE, 2005.
[11] Y. Chon, N. Lane, F. Li, H. Cha, and F. Zhao. Automatically characterizing places with opportunistic crowdsensing using smartphones. In *Proc. 14th Int. Conf. Ubiquitous Computing (UbiComp'12). ACM*, 2012.
[12] M. Chuang and P. Narasimhan. Automated viewer-centric personalized sports broadcast. *Procedia Engineering*, 2(2):3397–3403, 2010.
[13] A. Clauset, M. Newman, and C. Moore. Finding community structure in very large networks. *Physical review E*, 70(6):066111, 2004.
[14] D. Crandall, A. Owens, N. Snavely, and D. Huttenlocher. Discrete-continuous optimization for large-scale structure from motion. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3001–3008. IEEE, 2011.
[15] J. Duell, P. Hargrove, and E. Roman. *The design and implementation of Berkeley Lab's linuxcheckpoint/restart*.

Lawrence Berkeley National Laboratory, 2005.

[16] A. Efros, A. Berg, G. Mori, and J. Malik. Recognizing action at a distance. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 726–733. IEEE, 2003.

[17] A. Fitzgibbon, A. Zisserman, et al. Automatic 3d model acquisition and generation of new images from video sequences. In *Proceedings of European signal processing conference*, pages 1261–1269, 1998.

[18] R. Grzeszczuk, J. Kosecka, R. Vedantham, and H. Hile. Creating compact architectural models by geo-registering image collections. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pages 1718–1725. IEEE, 2009.

[19] A. Hampapur, L. Brown, J. Connell, A. Ekin, N. Haas, M. Lu, H. Merkl, and S. Pankanti. Smart video surveillance: exploring the concept of multiscale spatiotemporal tracking. *Signal Processing Magazine, IEEE*, 22(2):38–51, 2005.

[20] K. Heath, N. Gelfand, M. Ovsjanikov, M. Aanjaneya, and L. J. Guibas. Image webs: Computing and exploiting connectivity in image collections. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3432–3439. IEEE, 2010.

[21] F. Herranz, K. Muthukrishnan, and K. Langendoen. Camera pose estimation using particle filters. In *Int. Conf. on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–8, sep 2011.

[22] J. Hightower and G. Borriello. Location systems for ubiquitous computing. *Computer*, 34(8):57–66, 2001.

[23] H. Hile, R. Vedantham, G. Cuellar, A. Liu, N. Gelfand, R. Grzeszczuk, and G. Borriello. Landmark-based pedestrian navigation from collections of geotagged photos. In *Proceedings of the 7th International Conference on Mobile and Ubiquitous Multimedia*, pages 145–152. ACM, 2008.

[24] T. Kanade, R. Collins, A. Lipton, P. Burt, and L. Wixson. Advances in cooperative multi-sensor video surveillance. In *Proceedings of DARPA Image Understanding Workshop*, volume 1, page 2. Citeseer, 1998.

[25] C. Kim and B. Vasudev. Spatiotemporal sequence matching for efficient video copy detection. *Circuits and Systems for Video Technology, IEEE Transactions on*, 15(1):127–132, 2005.

[26] K. Kim, S. Yoon, and H. Cho. A faster color-based clustering method for summarizing photos in smartphone. In *Computer Science and Automation Engineering (CSAE), 2012 IEEE International Conference on*, volume 2, pages 561–565. IEEE, 2012.

[27] Y. Lee, Y. Ju, C. Min, S. Kang, I. Hwang, and J. Song. Comon: cooperative ambience monitoring platform with continuity and benefit awareness. In *Proceedings of the 10th international conference on Mobile systems, applications, and services*, pages 43–56. ACM, 2012.

[28] Y. Li, N. Snavely, D. Huttenlocher, and P. Fua. Worldwide pose estimation using 3d point clouds.

[29] J. Manweiler, P. Jain, and R. Roy Choudhury. Satellites in our pockets: an object positioning system using smartphones. In *Proceedings of the 10th international conference on Mobile systems, applications, and services*, pages 211–224. ACM, 2012.

[30] R. Mohan. Video sequence matching. In *Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on*, volume 6, pages 3697–3700. IEEE, 1998.

[31] C. Qin, X. Bao, R. Roy Choudhury, and S. Nelakuditi. Tagsense: a smartphone-based approach to automatic image tagging. In *Proceedings of the 9th international conference on Mobile systems, applications, and services*, pages 1–14. ACM, 2011.

[32] S. Reddy, A. Parker, J. Hyman, J. Burke, D. Estrin, and M. Hansen. Image browsing, processing, and clustering for participatory sensing: lessons from a dietsense prototype. In *Proceedings of the 4th workshop on Embedded networked sensors*, pages 13–17. ACM, 2007.

[33] E. Rosten and T. Drummond. Machine learning for high-speed corner detection. In *European Conference on Computer Vision (ECCV)*, pages 430–443, 2006.

[34] P. Sand and S. Teller. Video matching. In *ACM Transactions on Graphics (TOG)*, volume 23, pages 592–599. ACM, 2004.

[35] R. Schleicher, A. Shirazi, M. Rohs, S. Kratz, and A. Schmidt. Worldcupinion experiences with an android app for real-time opinion sharing during soccer world cup games. *International Journal of Mobile Human Computer Interaction (IJMHCI)*, 3(4):18–35, 2011.

[36] Z. Shen, S. Arslan Ay, S. H. Kim, and R. Zimmermann. Automatic tag generation and ranking for sensor-rich outdoor videos. In *MM*, pages 93–102. ACM, 2011.

[37] N. Snavely, S. Seitz, and R. Szeliski. Photo tourism: exploring photo collections in 3d. In *ACM Transactions on Graphics (TOG)*, volume 25, pages 835–846. ACM, 2006.

[38] K. Tuite, N. Snavely, D. Hsiao, N. Tabing, and Z. Popovic. Photocity: training experts at large-scale image acquisition through a competitive game. In *Proceedings of the 2011 annual conference on Human factors in computing systems*, pages 1383–1392. ACM, 2011.

[39] T. Yan, V. Kumar, and D. Ganesan. Crowdsearch: exploiting crowds for accurate real-time image search on mobile phones. In *Proceedings of the 8th international conference on Mobile systems, applications, and services*, pages 77–90. ACM, 2010.

[40] T. Zhao and R. Nevatia. Tracking multiple humans in complex situations. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(9):1208–1221, 2004.

[41] W. Zhao, D. Nister, and S. Hsu. Alignment of continuous video onto 3d point clouds. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(8):1305–1318, 2005.

[42] Z. Zhu, G. Xu, E. Riseman, and A. Hanson. Fast generation of dynamic and multi-resolution 360 panorama from video sequences. In *Multimedia Computing and Systems, 1999. IEEE International Conference on*, volume 1, pages 400–406. IEEE, 1999.