VIRTUALOT - A FRAMEWORK ENABLING REAL-TIME COORDINATE TRANSFORMATION & OCCLUSION SENSITIVE TRACKING USING UAS PRODUCTS, DEEP LEARNING OBJECT DETECTION & TRADITIONAL OBJECT TRACKING TECHNIQUES

Bradley J. Koskowich, Maryam Rahnemoonfar, Michael Starek

Texas A&M University - Corpus Christi
College of Science and Engineering
6300 Ocean Drive, Corpus Christi, TX, USA 78412

ABSTRACT

In this work we explore a combination of methods that allow us to analyze and study hyper-local environmental phenomena. Developing a unique application of monoplotting enables visualization of the results of deep-learning object detection and traditional object tracking processes applied to a perspective view of a parking lot on aerial imagery in real-time. Additionally, we propose a general algorithm to extract some scene understanding by inverting the monoplotting process and applying it to digital elevation models. This allows us to derive estimations of perspective image areas causing object occlusions. Connecting the real world and perspective spaces, we can create a resilient object tracking environment using both coordinate spaces to adapt tracking methods when objects encounter occlusions. We submit that this novel composite of techniques opens avenues for more intelligent, robust object tracking and detailed environment analysis using GIS in complex spatial domains provided video footage and UAS products.

Index Terms— photogrammetry, homography, computer vision, object detection, object tracking

1. INTRODUCTION

Hyper-local environments are complex, spatially constrained areas. A single floor of a building, a building itself, or a university campus could each be considered a hyper-local environment. Within these areas, any number of spatially relevant phenomena, such as building evacuations, can occur. Our goal is to capture and visualize these phenomena on an aerial image for enhanced situational awareness. Typically, awareness of this kind is achieved using security cameras. However, raw video footage usually requires human interpretation to understand its content and any effects beyond its immediate scope.

To reduce video information complexity to a model-able form requires us to combine several established methods. We build upon the concepts of Bozzini et al's monoplotting work [1] used for visualizing land cover changes from perspective imagery as GIS polygon geometry. We extend that interface by replacing the perspective imagery with video. Deep-learning object detection applied to said video is combined with an evaluation of multiple traditional tracking methods, whose outputs are recorded as points and lines. The transformed outputs can be used to visualize movement behavior on aerial imagery in real-time. This can provide first responders and resource management personnel a simulated birds-eye view of day-to-day operations or disasters as they unfold, increasing their situational awareness.

Object occlusion remains a barrier for our application. To that end, we compose a novel algorithm using monoplotting inputs to estimate occlusions explicitly in accompanying perspective video. This allows us the option to extend our detection and tracking platform by defining additional behaviors to follow when an object encounters an occluded region.

Our ongoing tests occur at Texas A&M University Corpus Christi (TAMUCC), USA, in a long-term effort to understand how the layout of road travel directions, temporary barriers, and crosswalks affect pedestrian and vehicular traffic. We believe this initial work is a novel combination of remote sensing, computer vision, and GIS principles which we can expand in the future to accomplish this ultimate goal.

2. RELATED WORKS

2.1. Monoplotting

The foundation for this project is the concept of monoplotting, the less well-known cousin of stereo-photogrammetry. This single-image photogrammetry method has been typically used for plotting historical images onto current orthometric maps or aerial orthophotos to visualize topographical changes over time, such as environmental phenomena like the growth of a forest or terrain elevation changes [1, 2]. Traditional monoplotting normally requires a recorded cam-

978-1-5386-7150-4/18/$31.00 ©2018 IEEE

era POSE, at least one calibrated camera image (though two are preferable), an aerial orthophoto, and an accompanying digital elevation model (DEM)[1].

There are several well defined relationships that can be leveraged to reconstruct complete or partial POSE parameters [3] given a series of known points in a pair of images. Computing an image homography also provides us the parameters of relative camera POSE between aerial and perspective images. Homographies can be adjusted for imprecision error via iterative least squares adjustment [1]. Incorporating the DEM with the aerial orthophoto, we can also derive transformation parameters from 3D-2D space, where the 2D perspective image maps back into 3D world space coordinates [1].

2.2. Object Detection & Tracking

Often issues in object variation (lighting, scale, deformation, etc.) preclude perfectly accurate object detection and tracking. Incorporating the YOLO framework [4] handles most variable presentation aspects. YOLO's flexibility on input size and speed makes it a natural choice over the RCNN family, especially over an eclectic mix of smaller input images. Subsequently, we explore the efficacy of several traditional tracking algorithms: Track-Learn-Detect (TLD), Kernalized Correlation Filters (KCF), and Multiple Instance Learning (MIL) [5, 6, 7] on the outputs of the YOLO network.

3. DATASETS

Perspective video was provided by the TAMUCC University Police Department. For this initial work we analyzed a 1280x720 video file at 20 frames per second from an AXIS Q6044-E PTZ Dome Network Camera. The camera was held mostly in a static POSE. Aerial imagery products of campus provided by the Measurement Analytics Lab at TAMUCC were generated from a fixed-wing UAV (Sensefly eBee) platform with an RGB camera. The orthophoto and DEM were dervived from a point cloud generated using Structure-from-Motion over a masked area of the parking lot with a ground sample distance of 2.79cm.

The applied YOLO network was trained on a subset of PASCAL VOC 2007 & 2012 data, specifically on the classes of people, cars, and motorbikes. This allows us to reduce some model overhead for increased performance.

4. METHODS

We stretch the limits of prior monoplotting work by starting with the most complex case of input sources available to us: a perspective view, wall-mounted video camera with no known world-space coordinates or calibration data. We were able to avoid using external measurements as inputs during the registration of perspective imagery with aerial orthophoto, allowing us to validate our results using a total station and avoiding some measurement bias.

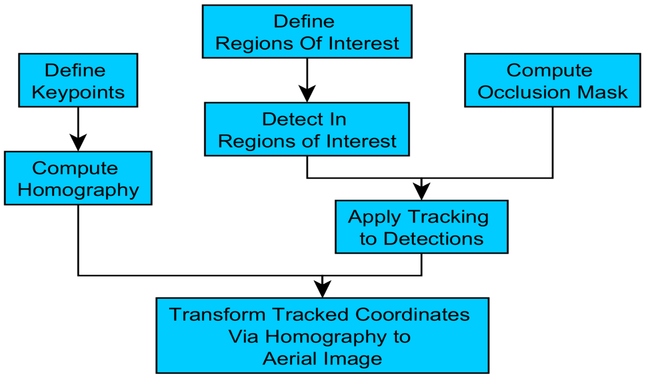

Define Regions Of Interest
Define Keypoints
Compute Occlusion Mask
Detect In Regions of Interest
Compute Homography
Apply Tracking to Detections
Transform Tracked Coordinates Via Homography to Aerial Image

Fig. 1. Processing Methodology

4.1. Required Inputs

Several inputs are necessary in combination with the perspective video footage, and the processing flow is visualized in Figure 1:

- Keypoints: 400 point pairs used to compute image homography collected by hand as our results applying Shi-Tomasi corner detection [8] were surprisingly sparse.
- Registrations: homography parameters are computed individually from perspective to aerial and vice versa (inverted) using the keypoints.
- Working Area Geometry: used to check the accuracy of a registration and create a mask of the working area in the perspective image containing physical occlusions.
- Regions Of Interest (ROIs): areas around the perspective view periphery where track-able objects are likely to pass entering or exiting the frame.

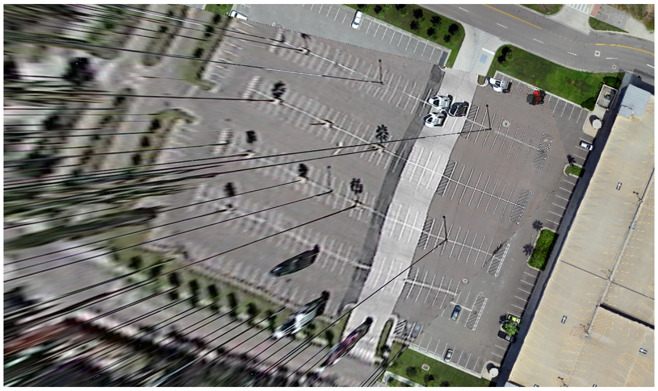


Fig. 2. Registration Overlay: The perspective image transformed onto the the aerial image.

4.2. Image Transformations

The process of deriving image transformations applies image homography and is iterated upon with several collections of keypoints, increasing in volume starting with 8 keypoints and doubling until we encompass all 400 points. The efficacy of each transformation is evaluated by measuring the difference of known points in the image with where they should align on the aerial orthophoto, an example of which is shown in Figure 2.

4.3. Occlusion Masking

To account for occlusions during object tracking, we estimate occlusion locations in the perspective view by transforming the DEM with an inverted image homography. This allows us to overlay the elevation values present in the DEM onto the perspective plane accurately. Thresholding elevation post-transformation in Figure 3 provides us a lower limit of occluded areas. The upper limit can be determined by computing the offset from the lower limit based on the elevation value. In the widest bounding envelope including these limits, a Hough Transform generates lines filtered by the best pair matched by angle relative to vertical and proximity. Point intersections of all boundary lines form polygons which are clipped by the defined working area to eliminate extraneous geometry, shown in Figure 4a.

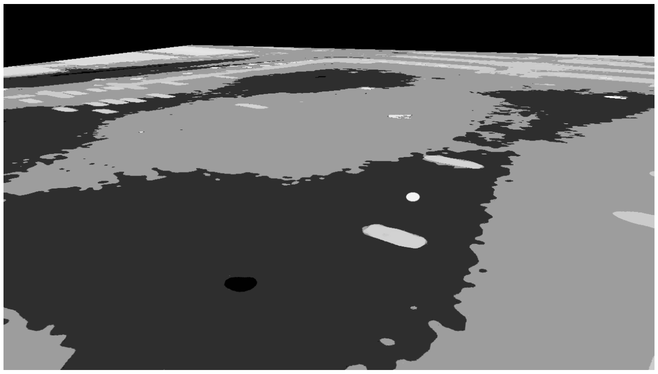


Fig. 3. The transformation of the DEM into the perspective image plane.

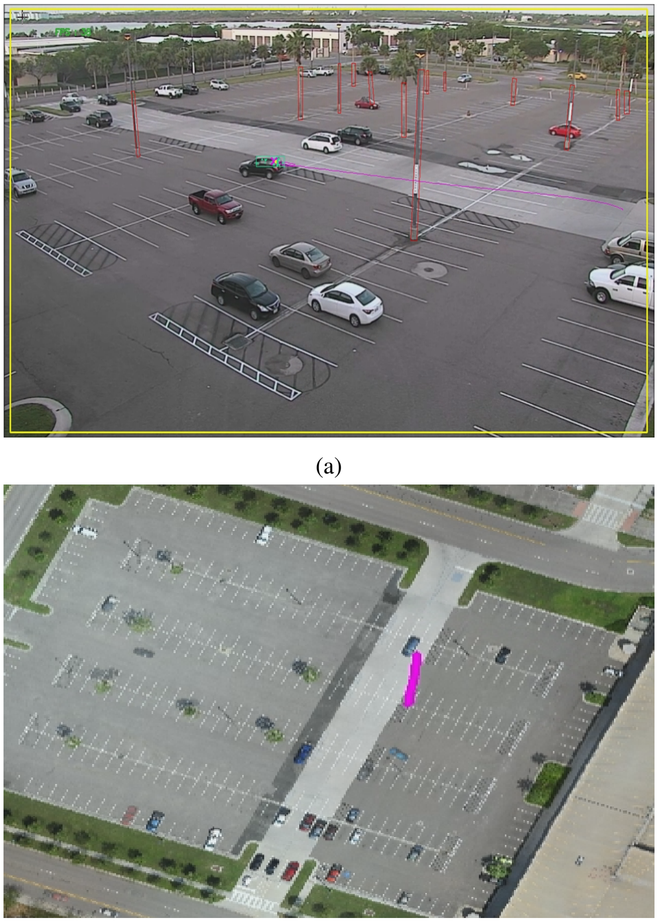
5. RESULTS

5.1. Registration Accuracy

We compute registration accuracy as the maximum value of deviation between identifiable points using the standard distance equation:

$$D = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Registration points nearest the origin are such that their deviation is negligible. As distance from the camera increases, registration accuracy decreases, visible in Figure 2 as crooked lines and disjoint connections. The highest recorded deviation across all registration iterations was ∼3.5m at the farthest ends of the 118×93m parking lot from the camera origin, ∼3% transformation error at worst. Notably, deviations are not linear. We theorize their cause is image imperfections due to the intentional lack of camera calibration and/or distortion caused by the weather dome.



(a)

(b)

Fig. 4. Detection and Tracking Results with Occlusion Mask. *a): Perspective View of Detection and Tracking Algorithm With Detected Occlusion Areas. b): Aerial View of Detection and Tracking Algorithm*

5.2. Occlusion Extraction

In our test case, our occlusion extraction method could isolate 12/13 vertical occlusion areas present in the image, at a 92% detection rate with ∼88% accurate fill rate. That is, the polygons drawn contain ∼88% occlusion pixels and ∼12% non-occluding pixels. However, being the only case in which this algorithm has been tested, we defer judgement regarding general efficacy.

5.3. Detection & Tracking Accuracy

In over six hours of reviewed video, YOLO detected every vehicle which passed through designated ROIs around the perspective view periphery. An optimization was made to detection operations by only calling them as movement-areas computed in ROIs began to decrease. This corresponded to a majority of instances where vehicles would present themselves best for detection. Tentatively, we would rate this application of YOLO as 99.99% accurate at object detection, whose outputs were passed to the tracking algorithms we evaluated.

Table 1 outlines performance of existing tracking methods available on OpenCV and our own variation on Lucas-Kanade optical-flow. Our variation performs k-means clustering on the detected features between frames and drops points which become stuck on similar features and exceed a distance limit. We define tracking accuracy as lock persistence on a set of 14 vehicles traveling radically different paths until they exit view. Global denotes tracking context in the entirety of the frame, while Patch denotes a moving window around tracked objects as a subset of the frame.

Table 1. Tracking Algorithm Performance*: At the time of this writing the TLD implmentation in OpenCV was not stable under Patch tracking. Value is a conservative estimate from what data could be recorded.

Tracker	**Global & Patch FPS with Coord. Transforms**	**Global & Patch Tracking Accuracy**
TLD	8, 16*	79%, 65%*
KCF	40, 70	58%, 58%
MIL	6, 12	65%, 65%
Optical-flow	70, 60	85%, 85%

6. CONCLUSIONS & FUTURE WORKS

We conclude that this is a valid approach for accomplishing our outlined goal of integrating video data with UAS products based on the relatively low degree of registration error over a comparatively large area. Theoretically it is possible to near-fully automate the processing workflow, however we are curious to investigate adapting alternative methods, such as Boerner's work on automatically computing camera POSE [9] to fully automate image registration. Similarly, regions of interest in perspective images combined with shapefiles of travel networks could isolate ROIs for vehicle detection algorithmically. We also plan to test the system with continually reduced image quality, in order to determine the minimum requirements where this system could operate with a negligible degree of uncertainty. As an alternative to traditional tracking mechanics, we also look to incorporate other in-progress work based on Bertinetto et al's [10] study of generic object tracking using Siamese networks.

7. REFERENCES

[1] C. Bozzini, M. Conedera, and P. Krebs, "A new monoplotting tool to extract georeferenced vector data and orthorectified raster data from oblique non-metric photographs," *International Journal of Heritage in the Digital Era*, vol. 1, no. 3, Swiss Federal Research Institute WSL, Insubric Ecosystem Research Group, CH-6500 Bellinzona, Switzerland, 2012.

[2] T. Produit and D. Tuia, "An open tool to register landscape oblique images and and generate their synthetic model," *REMOTE SENSING & SPATIAL ANALYSIS*, pp. 170–176, 2012. [Online]. Available: http://2012.ogrs-community.org/2012_papers/d3_2_produit_abstract.pdf

[3] D. A. Strausz Jr, "An application of photogrammetric techniques to the measurement of historic photographs," 2001. [Online]. Available: https://ir.library.oregonstate.edu/downloads/js956g515

[4] J. Redmon, A. Farhadi, U. of Washington, and A. I. for AI, "Yolo9000: Better, faster, stronger," in *Computer Vision and Pattern Recognition*. University of Washington; Allen Institute for AI, 2016. [Online]. Available: https://arxiv.org/pdf/1612.08242

[5] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, vol. 6, no. 1, Jan. 2010.

[6] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 3, pp. 583–596, mar 2015.

[7] B. Babenko, M.-H. Yang, and S. Belongie, "Visual tracking with online multiple instance learning," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, jun 2009.

[8] J. Shi and C. Tomasi, "Good features to track," *Computer Vision and Pattern Recognition*, 1994. [Online]. Available: http://www.ai.mit.edu/courses/6.891/handouts/shi94good.pdf

[9] R. Boerner and M. Krhnert, "Brute force matching between camera shots and synthetic images from point clouds," *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XLI-B5, pp. 771–777, jun 2016.

[10] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. S. Torr, "Fully-convolutional siamese networks for object tracking," *Computer Vision and Pattern Recognition*, 2016.