Multiview Object Detection and Tracking

K. Makroleivaditis, I. Karakostas, I. Mademlis, Prof. Ioannis Pitas
Aristotle University of Thessaloniki
pitas@csd.auth.gr
www.aiia.csd.auth.gr
Version 3.0

Artificial Intelligence & Information Analysis Lab

Multiview Object Detection and Tracking

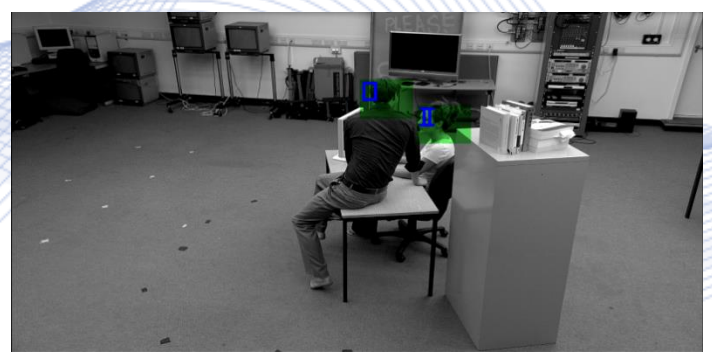
- **Multiview Human Detection**

- Multiview Object tracking

Multiview Human Detection

- Problem statement: Use information from multiple cameras to detect bodies or body parts, e.g. head.
- Applications:
  - Human detection/localization in postproduction.
  - Matting/segmentation initialization.

Camera 4



Camera 6

Multiview Human Detection

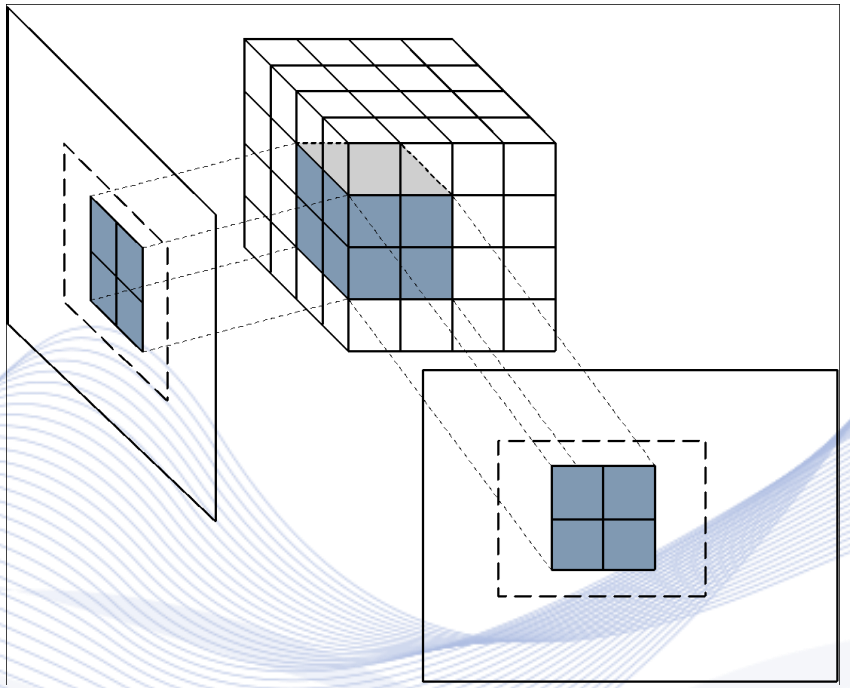
- ***Region-Of-Interests*** (ROIs) are typically bounding boxes. They are determined at a specific time t by the upper left and lower right rectangle coordinates $[x_l, y_l, x_r, y_r]$.
- Object ROI center: $\mathbf{c} = [x_c, y_c]^T$

Multiview Human Detection

- Head or body detection in two stages:
  - Use a head/face/body detector to derive ROIs in each view separately.
  - Insert these ROIs to an algorithm utilizing 3D information.
- Use of camera calibration parameters.
- Output: a rectified set of ROIs for each view that contains:
  - fewer false negatives
    - especially those due to occlusion are eliminated;
  - associations across views
    - all ROIs corresponding to the same human head/body are associated.

Multiview Human Detection

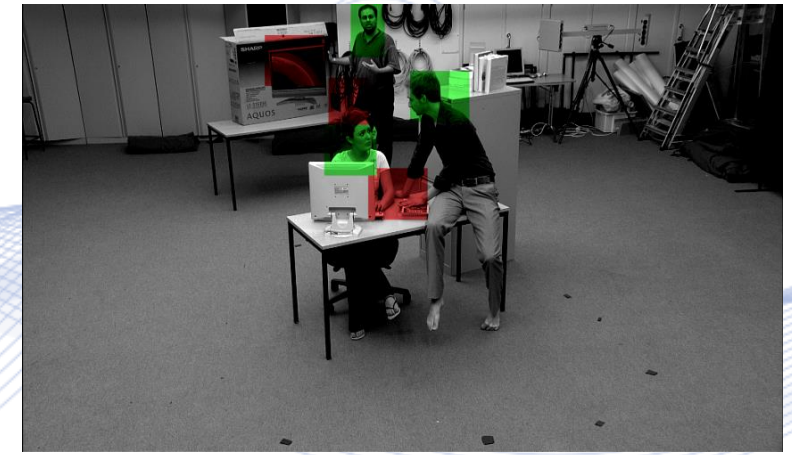
- Detected ROIs are projected back in the 3D space.
- A “probability volume” is created collecting “votes” from individual ROIs.
- High probability voxels correspond to the most probable head/body ROIs.



Multiview Human Detection

- The retained voxels are projected to all views.
- For every view we reject ROIs that have small overlap with the regions resulting from the projection.

Camera 2



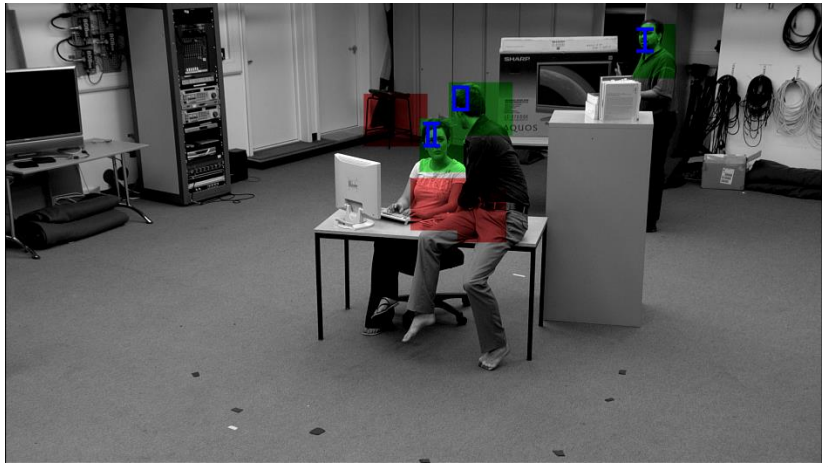
Camera 4

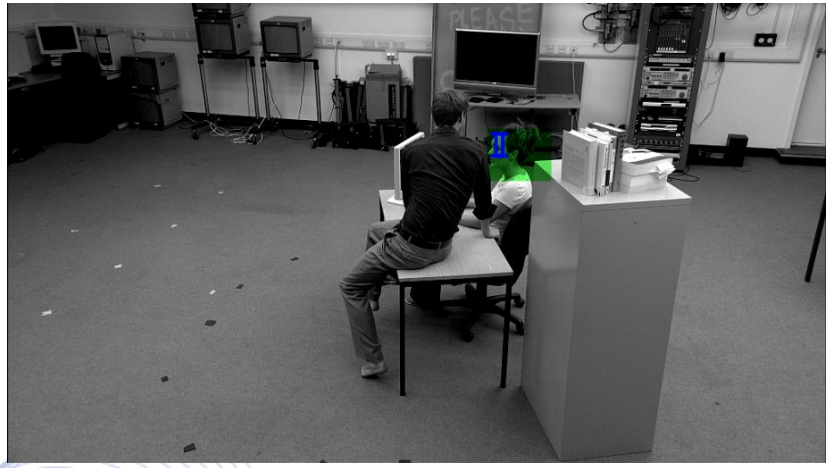
Multiview Human Detection

- ROI association across different views:
  - A voting scheme is used to find ROIs across views containing projections of the same voxels.
  - These ROIs are associated across views.
  - ROIs that are not associated are rejected.
  - Further elimination of false positives may be achieved.
- ROI rectification:
  - Using 3D information we create ROIs for a certain head/body in views lacking in them.
  - False negatives elimination.

Multiview Human Detection

Camera 4

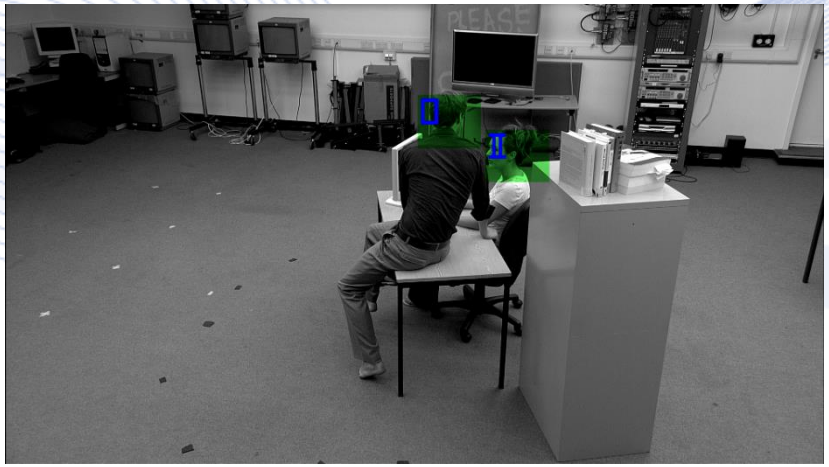




Camera 6

After ROI Rectification

Camera 4



Camera 6

Multiview Human Detection

- Multi-view human tracking can be formulated in a probabilistic framework.
- Let K determine the number of trajectories, V_i the detected human boxes, $c_i \in [0, K]$ the trajectory index ($c_i = 0$ is equivalent to a false alarm of box i).
- Let $V = \{(V_i, c_i)\}$ denote the set of boxes and $\tau = \{(\tau_k, t_k^s, t_k^e)\}$ trajectories where t_k^s is the starting frame and t_k^e is the ending frame.
- Formula for 3D localization and cross-view human tracking:
  - $W = (K, V, \tau)$

Multiview Human Detection

- Information extracted for scene modeling:
  - Ground-plane region in images
  - Cross-view homograph matrix between views, in the form of $\mathbf{H}^{u,v} \in R^{3\times3}$, where *u, v* are the camera indexes
  - Projection matrix that transforms a 3D coordinate into a view, in the form of $\mathbf{M}^{u}$
  - View-to-map homograph matrix between each view and the scene map, in the form of $\mathbf{H}^{u}$

Multiview Human Detection

- Appearance energy term (E^{app}) : utilized to maximize similarities of detected human boxes of the same trajectory.
- For a view u, let i index the detected boxes and c_i index the trajectories. Then the appearance energy term takes the form:

$$E_u^{app} = -\sum_{i,j} log \frac{P(c_i = c_j, \| f_i - f_j \|)}{P(c_i \neq c_j, \| f_i - f_j \|)}$$

- $\| f_i - f_j \|$ determines the norm of feature distance between f_i and f_j
- $P(c_i = c_j, \| f_i - f_j \|)$ determines the probability of $c_i = c_j$ for f_i appearance feature (e.g. color).

Multiview Object Detection and Tracking

- Multiview Human Detection

- **Multiview Object tracking**

Multiview Object tracking

- Motion blurs, partial occlusions, background clutter and image motion are the main issues that an object tracker has to overcome, in order to precisely track a moving object.

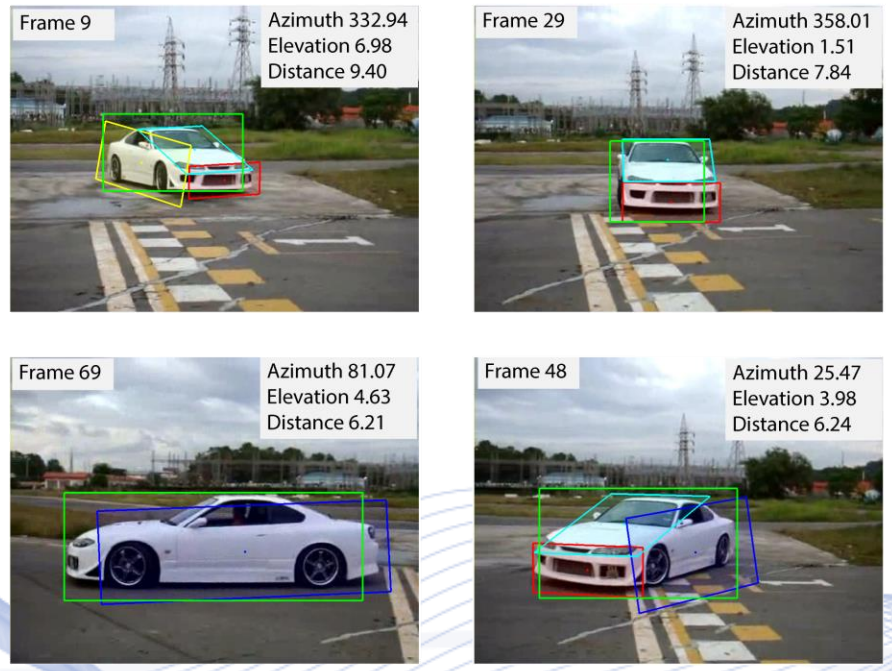
- Tracking a detected object in frame $t+1$.
  - Predicted object position $[x, y]^T(t+1)$
  - Compute ROI parameter vector $\widehat{\mathbf{y_1}}(t+1) = [x, y, w, h]^T$ within a *search region* on frame $t+1$
  - Retain object ID $\mathrm{J}(t+1) = \mathrm{J}(t)$

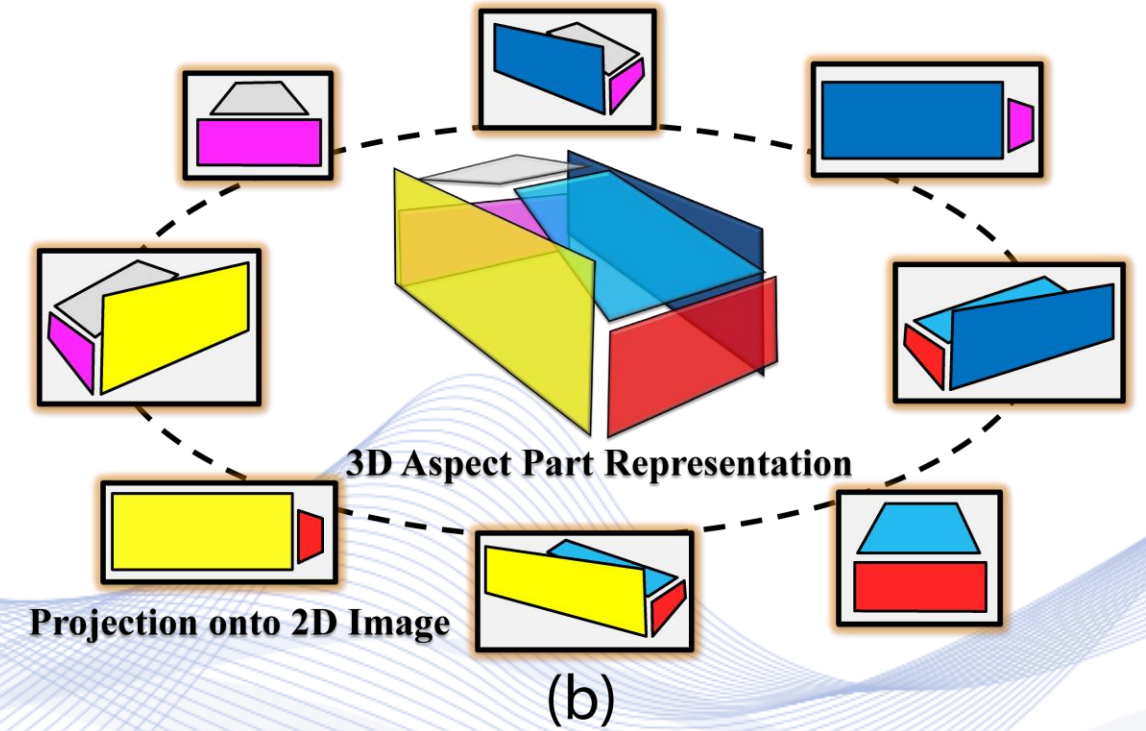
- Tracking failures can occur due to occlusions to the background and in those cases ***object-redetection*** techniques are implemented.

Multiview Object tracking

- Different and multiple viewpoints pose a challenge to traditional object tracking methods.
- Variations on topological appearance are dealt with by viewpoint transformations; representing objects with 3D aspect parts and modeling the connection in part-based particle filtering frameworks.
- The tracking framework can handle the appearance alternation and accurately predict the visibility and shape of a part.

Multiview Object tracking





(a): Example output for the tracking framework; (b): 3D aspect part representation of the car, and its projections from different viewpoints

Multiview Object tracking

- A multicamera based tracking method can be employed to shooting missions that use multiple camera-equipped UAVs.
- The proposed approach relies on fusing the results of separate 2D visual trackers running on-board each drone.
- This can handle target occlusions due to obstacles, since there is almost always at least one drone with an unoccluded view of the target.

Multiview Object tracking

- The method is based on locating the 3D target position in the world by back-projecting detected/tracked ROIs from the *K* UAV-mounted cameras. Camera parameters are considered known.
- The reliability of each tracked ROI (determined by the tracker response map) is exploited for weighting the per-drone results during centralized fusion.
- The 3D target position is then separately projected back to the 2D image plane of each UAV/camera.

Multiview Object tracking

- Likelihood $P(Z_t|X_t, V_t)$ estimates the compatibility between the state of the target (X_t, V_t) with the observation Z_t at time t.

- The overall likelihood of the object is decomposed as the product of the likelihoods of the 3D aspect parts:

  - $P(Z_t|X_t, V_t) = \prod_{i=1}^{n} P(Z_t|X_{it}, V_t)$, where

    - $P(Z_t|X_{it}, V_t)$ determines the appearance likelihood of part i.

Multiview Object tracking

- Motion prior $P(X_t, V_t | X_{t-1}, V_{t-1})$ predicts the state of the target based on its previous state.
- Motion prior decomposition according to part location and viewpoint:
  - $P(X_t, V_t | X_{t-1}, V_{t-1}) = P(X_t | X_{t-1}, V_{t-1}, V_t) P(V_t | X_{t-1}, V_{t-1}) = P(X_t | X_{t-1}, V_t) P(V_t | V_{t-1})$, where
- $P(X_t | X_{t-1}, V_t)$ models the change in location
- $P(V_t | V_{t-1})$ is the viewpoint motion

Multiview Object tracking

- Change in location can be modeled by implementing a *Markov Random Filed* to capture the relationship between parts:
  - $P(X_t|X_{t-1}, V_t) \propto \prod_{i=1}^{n} P\left(X_{it}\middle|X_{i(t-1)}\right) \prod_{(i,j)} \Lambda\left(X_{it}, X_{jt}, V_t\right)$, where
- $P(X_{it}|X_{i(t-1)})$ is the motion model for part i
- $\Lambda(X_{it}, X_{jt}, V_t)$ is the pairwise potential that constrains the relative location of two parts according to the 3D aspect part representation

Multiview Object tracking

- Location and viewpoint motions are modeled with Gaussian distributions centered on the previous location and viewpoint respectively as :
  - $P(X_{it}|X_{i(t-1)}) \sim \mathrm{N}(X_{i(t-1)}, \sigma_x^2 \sigma_y^2)$
  - $P(V_t|V_{t-1}) \sim \mathrm{N}(V_{t-1}, \sigma_\alpha^2 \sigma_e^2 \sigma_d^2)$, where
- σ_x^2, σ_y^2 are the variances of the Gaussian distributions for 2D part center coordinates
- $\sigma_\alpha^2, \sigma_e^2, \sigma_d^2$ are the variances of the Gaussian distributions for azimuth, elevation and distance respectively

Multiview Object tracking

- We define an orthonormal, right-handed World Coordinate System (WCS) and a time-varying, orthonormal, right-handed UAV/camera-centered Coordinate System (UCS) for each UAV.

- At each time instance, a homogeneous transformation matrix $\mathbf{T}_j$, encoding rotation and translation, transforms between them for the j-th UAV.

Multiview Object tracking

- At current time instance i:
  - For the 2D ROI coming from the *j*-th UAV, its center (provided in 2D pixel coordinates by the *j*-th tracker) is transformed first into continuous 3D UCS coordinates and then into WCS, using $\mathbf{T}_j$.
  - The 3D coordinates of the *K* centers of projection (COPs) are known in WCS.
  - Thus, two *K* x 3 matrices are constructed:
    - **M**, containing the 3D WCS coordinates of the *K* ROI centers.
    - **N**, containing the 3D WCS coordinates of the *K* COPs.
  - **M** and **N** are exploited for finding the 3D target location in WCS, by searching for the intersection point of the 3D lines defined by the ROI center and the COP of each UAV.
  - $\mathbf{B} = \frac{\mathbf{N}-\mathbf{M}}{\sqrt{\mathbf{a}}\,\mathbf{1}}$, where . denotes element-wise division, **1** is a 1 x 3 row matrix and $\boldsymbol{a}$ is: $a_i = \sum_{j=1}^{3}\left(N_{ij} - M_{ij}\right)^2$.

Multiview Object tracking

- At current time instance i (continued):
  - $\mathbf{S} = \mathbf{B}^T\mathbf{B} - K\mathbf{I}_{3x3}$
  - $\mathbf{c} = \sum_{j=1}^{K} \mathbf{C}_1 + \mathbf{C}_2 + \mathbf{C}_3$

    $\mathbf{C}_1 = \mathbf{M} \odot (\mathbf{B} \odot \mathbf{B} - 1)$,
  - $\mathbf{C}_2 = \mathbf{M} \odot (\mathbf{B} \odot \mathbf{B}\mathbf{A}_1 - 1)\mathbf{A}_2$,
    $\mathbf{C}_3 = \mathbf{M} \odot (\mathbf{B} \odot \mathbf{B}\mathbf{A}_2 - 1)\mathbf{A}_1$.
  - In the above, the symbol $\odot$ denotes the Hadamard product, while the row permutation matrices $\mathbf{A}_1$ and $\mathbf{A}_2$ switch the rows of matrix $\mathbf{B}$ being multiplied with them as $[\mathbf{b}_3, \mathbf{b}_1, \mathbf{b}_2]^T$ and $[\mathbf{b}_2, \mathbf{b}_3, \mathbf{b}_1]^T$, respectively.
  - The intersection point is given in 3D WCS by $\mathbf{p}_m = [\mathbf{c}/\mathbf{S}]^T$.
  - $\mathbf{p}_m$ is then projected to each of the K cameras using the known camera parameters.

Multiview Object tracking

- In order to evaluate the presented multicamera tracking method, a realistic simulated video dataset was generated.
- The AUTH-developed, AirSim-based simulator was employed.
- A bicycle race scenario with multiple cyclists was implemented in this simulation environment. Each sequence may include up to 10 cyclists, differing only in the color of their jerseys.

Multiview Object tracking

- The sequences where generated by simulating 3 camera equipped UAVs flying simultaneously under specific UAV/camera motion types (CMTs) and framing shot types (FSTs).
  - 3-UAV Orbit
  - 2-UAV Chase plus 1-UAV VTS
  - 3-UAV Track setups
- The evaluation dataset contains more than 90000 video frames, at a resolution of 640 x 360 pixels and a framerate of 25 FPS, while each video is more than 6.5 minutes long.
- KCF was used as the baseline 2D visual tracker.
  - Fast correlation filter able to run in real time on embedded AI computing platforms.

Multiview Object tracking

Multiview 3-UAV ORBIT

(a) Video frame from UAV 0.

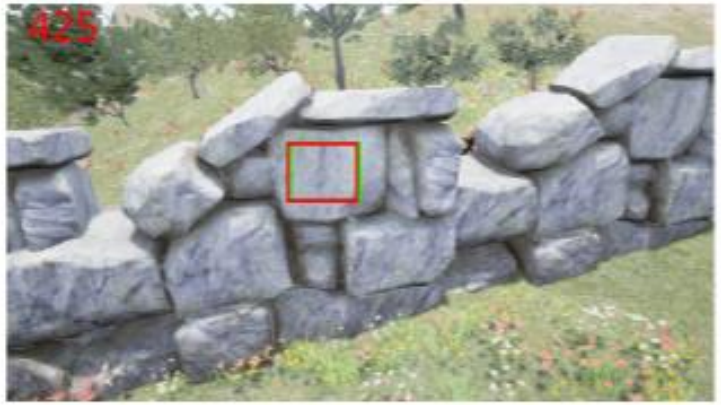
(b) Video frame from UAV 1.

(c) Video frame from UAV 2.

Multiview Object tracking

Multiview 3-UAV TRACK

(a) Video frame from UAV 0, following the desired target with an LTS CMT.



(b) Video frame from UAV 1, following the desired target with an LTS CMT.

(c) Video frame from UAV 2, following the desired target with a VTS CMT.

Multiview Object tracking

Multiview Object tracking

Baseline 3-UAV TRACK (without multiview fusion)

(a) Frame from UAV 0 following the desired target with an LTS CMT.

(b) Frame from UAV 1 following the desired target with an LTS CMT.

(c) Frame from UAV 2 following the desired target with an VTS CMT.

Multiview Object tracking

- Comparison in terms of tracking precision between:
  - Single-view KCF tracker
  - Proposed multiview KCF tracker with known ground-truth camera parameters
  - Proposed multiview KCF tracker with noisy camera parameters simulating RTK GPS accuracy

		Tracking Method		
		Baseline	**Proposed**	**Proposed (RTK)**
UAV 0	Precision	0.102	0.711	0.695
UAV 1	precision	0.070	0.752	0.698
UAV 2	precision	0.130	0.714	0.676

Multiview Object tracking

- The proposed multiview method depends on the following information being available:
  - Camera parameters
  - 2D target ROIs from single-view tracker running independently on each UAV
- Additional sources of information that could be exploited for increased accuracy include:
  - Target 3D position (e.g., with noisy GPS measurements)
  - LIDAR measurements from optical sensors mounted on each UAV
- All the above information must be temporally synchronized and updated with the same frequency (e.g., per video frame, for real-time operation)

Multiview Object tracking

- The double need for temporal synchronization and common update frequencies arise both in on-line and off-line/a-posteriori multi-view tracking.
- Any problems of update frequency discrepancy between different information sources can be easily solved via interpolation.
- However, temporal synchronization can be a much more challenging issue.

Multiview Object tracking

- The overall problem can be formulated in a general manner, based on the reprojection error minimization problem of the Structure-from-Motion and Visual SLAM:

$$\underset{\mathbf{P}_i,\mathbf{X}_j}{\operatorname{argmin}} \sum_{i,j} \left(\left\| \mathbf{x}_{ij} - (\mathbf{P}_i \mathbf{X}_j) \right\|^2 \right)$$

  - $\mathbf{P}_i$ is the perspective projection matrix of the $i-th$ time instance/video frame (encoding both extrinsic and intrinsic camera parameters)
  - $\mathbf{X}_j$ is the $j-th$ 3D scene point world coordinates
  - $\mathbf{x}_{ij}$ is the detected/tracked 2D on-frame position of $\mathbf{X}_j$ at the $i-th$ time instance/video frame, in pixel coordinates

Multiview Object tracking

- A single, unknown target 3D point $\mathbf{X}^t$ exists at time instance *t*.
- $x_i^t, 1 \leq i \leq K$: 2D on-frame projection of $\mathbf{X}^t$ on the $i-\text{th}$ UAV camera sensor at time *t,* estimated by each UAV independently.
- $\mathbf{P}_i^{t+\Delta_{t_i}}$ is the known camera projection matrix for the $i-\text{th}$ UAV at time instance *t*. Δ_{t_i} is unknown, hence the synchronization issue.
- In a similar manner, $\mathbf{X}_{GPS}^{t+\Delta_{t'}}$ and $\mathbf{X}_{LIDARi}^{t+\Delta_{ti''}}$ are the known, noisy and unsynchronized measurements of $\mathbf{X}^t$ derived by on-target GPS and on-drone LIDAR sensor, respectively. $\Delta_{t_i'}$ and $\Delta_{t_i''}$ are unknown.

Multiview Object tracking

- Under this setup, the general problem can be formulated as follows:

$$\underset{\mathbf{X}^t,\Delta_{t_i},\Delta_{t_i}',\Delta_{t_i}''}{\arg\min} \sum_i \left\|\mathbf{x}_i^t - \mathbf{P}_i^{t+\Delta_{t_i}}\mathbf{X}^t\right\|^2 + \left\|\mathbf{X}^t - \mathbf{X}_{GPS}^{t+\Delta_{t'}}\right\|^2 + \left\|\mathbf{X}^t - \mathbf{X}_{LIDAR}^{t+\Delta_{t''}}\right\|^2$$

- The problem definition can be easily extended with additional terms, representing additional information sources.
- This formulation can serve as the basis for a unified definition of the multi-view tracking and synchronization problem.

Multiview Object tracking

- In the off-line/a-posteriori analysis scenario, there are independently available:
  - *K* timeseries, each one containing UAV camera projection matrices for a UAV
  - *K* timeseries, each one containing 2D tracker ROI centers from a UAV
  - *K* timeseries, each one containing 3D target world position measurement from a UAV (using LIDAR)
  - One timeseries containing 3D target world position measurements (using GPS)
- Synchronization of these timeseries would first require defining a temporal window of *T* video frames.
  - Only the subset of each timeseries falling within this window would be employed.

Multiview Object tracking

- Using this off-line setup, simple exhaustive search can be trivially employed for synchronizing on-drone camera parameters, on-target GPS and on-done LIDAR measurements with the on-drone tracker timeseries, for all K drones, across the entire temporal window of T instances. This would have a computational complexity of $O(TK^3)$.

- By employing the proposed problem definition as an optimization problem, and employing a suitable optimization algorithm for solving it, may potentially reduce this complexity to a significant degree.

Multiview Object tracking

- After acquiring the MULTIDRONE experimental media production data, multiview UAV-captured footage depicting a rowing boat race were successfully synchronized by AUTH using the exhaustive search approach.

Bibliography

[PIT2017] I. Pitas, “Digital video processing and analysis” , China Machine Press, 2017 (in Chinese).
[PIT2013] I. Pitas, “Digital Video and Television” , Createspace/Amazon, 2013.
[PIT2021] I. Pitas, “Computer vision”, Createspace/Amazon, in press.
[NIK2000] N. Nikolaidis and I. Pitas, “3D Image Processing Algorithms”, J. Wiley, 2000.
[PIT2000] I. Pitas, “Digital Image Processing Algorithms and Applications”, J. Wiley, 2000.

Q & A

Thank you very much for your attention!

More material in
http://icarus.csd.auth.gr/cvml-web-lecture-series/

Contact: Prof. I. Pitas
pitas@csd.auth.gr