Long-Range Target Acquisition and Visual Servoing with a UAV

Athanasios Tsoukalas , Nikolaos Evangeliou , Anthony Tzes *

Posted Date: 3 April 2026

doi: 10.20944/preprints202604.0178.v1

Keywords: UAV; visual servoing; YOLO; Gen6D; 3D object detection; robotics

 doi:10.20944/preprints202604.0178.v1

Article

Long-Range Target Acquisition and Visual Servoing with a UAV

Athanasios Tsoukalas [1], Nikolaos Evangeliou [1] and Anthony Tzes [1,2,*]

[1] Robotics and Intelligent Systems Control Lab, New York University Abu Dhabi (NYUAD), Abu Dhabi, UAE
[2] Center for Artificial Intelligence and Robotics (CAIR), NYUAD, UAE
* Correspondence: anthony.tzes@nyu.edu

Abstract

This article identifies, using a zero-shot method (Gen6d), the 3D-bounding box of a target far-distanced from a UAV. Furthermore, it infers the attached camera's pose to the drone, based on the underlying training on the visual data. These visual data are used in a YOLO-framework to identify targets belonging to a class. The vertices of the orthogonal 3D-box are used in a visual-servoing scheme on the attached gimbal on UAV. The camera has a varying focal length (zoom) and the indirect objective is to move the UAV close to the target while reducing the zoom factor. Initially, the UAV starts with a large zoom-factor (36×) at a far distance (100m) from the target. The UAV approaches the target using the visual servoing scheme, while reducing its zoom at discrete steps and maintaining its focus. Experimental results indicate the efficiency of the proposed method.

Keywords: UAV; visual servoing; YOLO; Gen6D; 3D object detection; robotics

1. Introduction

Object detection has been computationally demanding process, since it requires object classification and localization. Recent techniques include the extraction of the Region of Interest (RoI), feature extraction, and regression. Anchor-based methods include R-CNN, YOLO [1] and SSD for prediction of the RoI. Anchor-point based techniques have evolved resulting in very accurate and fast algorithms. Of paramount importance is the real-time implementation of the suggested algorithms, which trains systems to detect novel objects on various benchmarks (Pascal VOC) [2].

Most of the suggested works include 2D-object detection, while moving towards the 3D-case typically requires the use of various sensors like RGB-D cameras to infer the point cloud data for describing the object.

To overcome the need for additional sensors, visual-only techniques have been employed, ranging from monocular [3]to stereo imaging [4], or hybrid [5] methods using tight oriented 3D-cuboids, ending to multi-view based techniques [6]. Despite its heavy computational load, photogrammetry can be used for the 3D reconstruction problem [7]. Commercial photogrammetry pipelines have been developed and with powerful GPUs, the typical achieved rate for 3D-reconstruction from multiple images is close to 15 FpS [8]. Furthermore there it is extremely difficult to label specific objects and semantic photogrammetry [9] needs further exploitation prior to its adoption.

Relying on the object properties and other geometrical constraints, deep convolutional neural networks have been employed in [10] To overcome the computational burden, the classified object is enhanced by a mechanism relying on YOLO [11]. Datasets using RGB-D video streams [12] have been generated to facilitate the testing of 3D-object detection, reconstruction and segmentation.

Rather than directly using the point cloud, low level features using various information techniques have been suggested in [13]. The difficulty in determining 3D-object rotation classification was addressed in [14] by employing distinct local coordinates, bounding box scales and orientations.

 doi:10.20944/preprints202604.0178.v1

Similarly, Cross Potential Radar Point Clouds have been employed in [15] solving the problem of noise and sparsity, while the utilized 3D-bounding box can be used in the sequel for grasping purposes of the 3D-object by robot-grippers [16].

The presented method is an extension to [17] given that the camera's pose is a priori known within a certain degree of certainty. The mathematics behind the suggested algorithm relies on intersection of pyramids with convex sets with subsequent convex polyhedra meshing [18]. The assumption is that the object is recognized with certain certainty within the camera's Field of View (FoV). The object is classified using Yolo-v8, a convolution-based feed forward neural network where its input layer is followed by multiple convolutional layers to acquire feature maps. These feature maps are transformed to one-dimensional feature vectors before being used as input to the fully connected layer(s).Yolo-v8 considers object detection as a single pass regression problem, and was introduced in 2023 by Ultralytics as being faster and more accurate than its previous versions, making it ideal for object detection and image classification.

YOLO provides the 2D-bounding box encapsulating the object, which, due to lack of depth, corresponds to a pyramid-uncertainty protruded from the base-vertex of the camera. Intersection of pyramids from successive images form the 3D-bounding box of the object, which corresponds to a convex set of intersected pyramids. If there is translational and positional uncertainty regarding the camera's pose, then a convex shape is substituted (instead of the pyramids). The volume of the 3D-bounding box is non-increasing and results in computing the convex-hull of the object. Simulation studies using a pre-trained YOLO from ImageNet indicate that the 3D-bounding box converges to that of the object's convex hull.

The contribution of the work is in computing the 3D-bounding box of the object using a zero-shot visual method. The convex hull of the object can also be computed, yet results in a considerable computational load, which prohibits its real-time implementation in grasping maneuvers. However the vertices of the classified object bounding box can be used in a constrained visual-servoing scheme at a rate of 4 Frames per Second (FpS). This is particularly useful for servoing towards objects using UAVs, classified and detected from large distances. Rather than relying on the object's 2D-bounding box, computed via YOLO-v8, the developed technique provides an estimate of the camera pose, thus allowing its 3D-box computation. The visual servoing scheme uses varying zoom and manual focus dependent on the computed distance between the object and the camera, and proper modifications to increase the system's robustness in defining the camera's pose are provided. Furthermore,the visual servoing algorithm is modified in order to account for any translation and yaw-orientation only. The developed framework appears in Figure 1.

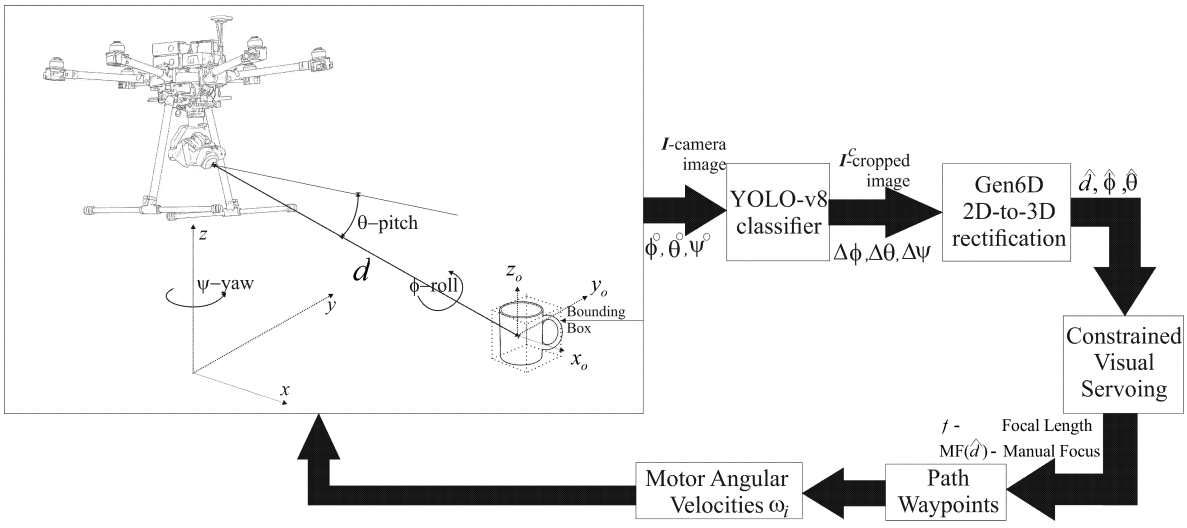

Figure 1. Visual Servoing Framework.

 doi:10.20944/preprints202604.0178.v1

This paper is structured in the following manner. Section 2 defines the visual target acquisition problem. Section 3 describes the necessary modifications for providing the 3D-bounding box of the object, using the zero-shot method Gen6D [10]. The constrained visual servoing problem with varying zoom and manual focus (distance dependent) is described in Section 4. Experimental and Simulation[1] studies are offered in Section 5 following by concluding remarks in Section 6.

2. Visual Target Acquisition

Let the camera's pinhole point be placed at $l^{\circ}_{0,3} = [x_c, y_c, z_c]^{\top}$ and its pose $\mathcal{P}^{\circ} = \left[l^{\circ,\top}_{0,3}, \phi^{\circ}, \theta^{\circ}, \psi^{\circ}\right]^{\top}$, when pointing to the center of its pixel-array $(\frac{W}{2}, \frac{H}{2})$, as shown in Figure 2.

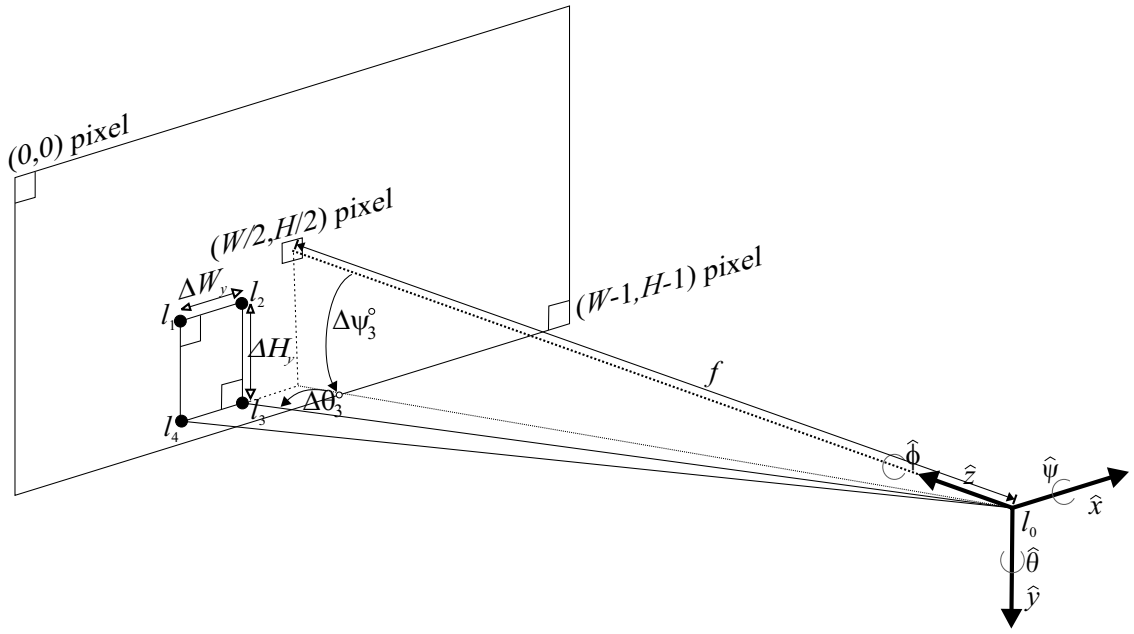

Figure 2. YOLO induced pyramid.

Every pixel has dimensions $(\rho_w \times \rho_h)$m (width × height) and a focal length f; resulting in the camera's Field of View (FoV) $\left[\tan^{-1}\left(\frac{W\rho_w}{f}\right) \times \tan^{-1}\left(\frac{H\rho_h}{f}\right)\right]$. Thus, the camera's intrinsic parameter vector is $\mathcal{C} = [W, H, \rho_w, \rho_h, f]^{\top}$.

YOLO classifies and identifies the 2D-bounding box of the object (within the camera's FoV) defined by its four 2D-pixels at the image plane $l_i = [x_i, y_i]^{\top}, i = 1, \ldots, 4$. The remaining 3D-vertices of the pyramid Pyr°_k, $l^{\circ}_{i,3}, i = 1, \ldots, 4$ (excluding its apex $l^{\circ}_{0,3}$) centered at the camera's coordinate system $C^{\circ}_{0,3} = [\hat{x}, \hat{y}, \hat{z}]^{\top}$ are $l^{\circ}_{i,3} = \begin{bmatrix} \left(x_i - \frac{W}{2}\right)\rho_w \\ \left(y_i - \frac{H}{2}\right)\rho_h \\ \hline f \end{bmatrix}$. The YOLO-captured 3D-parameters are labeled as $\mathcal{Y}^{\circ} = \left[l^{\circ}_{i,3},\ i = 1, \ldots, 4\right]^{\top}$, and the object resides within the noted pyramid Pyr°_k at the kth-instant.

The detected object's FoV is $\Delta^{\circ}_{\theta} \times \Delta\psi^{\circ}_i = (|\Delta\theta^{\circ}_1 - \Delta\theta^{\circ}_2| \times |\Delta\psi^{\circ}_2 - \Delta\psi^{\circ}_3|)$, where $\Delta\theta^{\circ}_i = \tan^{-1}\frac{\left(l^{\circ,1}_{i,3} - l^{\circ,1}_{0,3}\right)}{f}$, $\Delta\psi^{\circ}_i = \tan^{-1}\frac{\left(l^{\circ,2}_{i,3} - l^{\circ,2}_{0,3}\right)}{f}$.

The nominal (noiseless) pyramid at time k is defined as $\text{Pyr}^{\circ}_k\left[R_p, \mathcal{C}, \mathcal{Y}^{\circ}, \mathcal{P}^{\circ}\right]^{\top}$, and the intersection of pyramids of successive images provides the 3D-convex hull of the classified object.

The convex hull [18] is formed through intersections on convex pyramids and encompasses the classified object having a monotonically decreasing volume $\text{vol}\bigcap_{i=1}^{k}\text{Pyr}^{\circ}_i \leq \text{vol}\bigcap_{i=1}^{k-1}\text{Pyr}^{\circ}_i$. This

[1] Due to the state-of-war in the Gulf region, we were unable to cross-verify our derived experimental results. For this reason, sim-to-real studies are offered inhere.

 doi:10.20944/preprints202604.0178.v1

convex hull of the object, $\mathrm{Co}(O) \subset \bigcap_{i=1}^{k} \mathrm{Pyr}_i^{\circ}$ as long as YOLO's bounding box $\mathrm{B}(W_y^c, H_y^c) \supseteq Co(O)$ encapsulates the object classified with some confidence y^c to the right class.

The inherent assumption is that the camera's pose is known; if this is inaccurate, or $\mathcal{P} = \mathcal{P}^{\circ} + \Delta\mathcal{P}$, the algorithm is modified to account for this uncertainty.

Assuming perfect orientation measurements, or $\mathcal{P} = \mathcal{P}^{\circ} + \Delta\mathcal{P}_t$, where $\| \Delta\mathcal{P}_t \| \leq [[U_x, U_y, U_z], 0^{\circ}, 0^{\circ}, 0^{\circ}]^{\top}$, where $U_x(U_y)[U_z]$ is the uncertainty along the $x(y)[z]$ axis of the world coordinate frame.

This orthogonal parallelepiped uncertainty, $\mathcal{U}$, at the pyramid's vertices $l_{i,3} = l_{i,3}^{\circ} \pm \left[U_x, U_y, U_z\right]^{\top}$ is parallel to the world coordinate frame with lengths $2\left[U_x, U_y, U_z\right]^{\top}$, defined as $\mathrm{Par}\left[\mathcal{P}^{\circ}_{\mathrm{vertex}_i}, \mathcal{U}\right]$, where vertex_i is the ith vertex of $\mathrm{Pyr}_{k,t}^{\circ}\left[R_p, \mathcal{C}, \mathcal{Y}^{\circ}, \mathcal{P}^{\circ}\right]^{\top}$ placed at $l_{i,4}^{\circ}$, and $_{k,t}$ corresponds to the time k while t reflects the translational uncertainty.

Similarly , assume perfect translational measurements and inaccurate rotational uncertainty, or $\mathcal{P} = \mathcal{P}^{\circ} + \Delta\mathcal{P}_o$, where $\| \Delta\mathcal{P}_t \| \in \left[0, 0, 0, U_{\psi}^{+}, U_{\theta}^{+}, U_{\phi}^{+}\right]^{\top}$, where $U_{\psi}^{+}, (U_{\theta}^{+}), [U_{\phi}^{+}]$ is the maximum rotational uncertainty around $x(y)[z]$ axis of the world coordinate frame; or $\bar{U}_{\phi} \in \left[-U_{\phi}^{+}, U_{\phi}^{+}\right]$, $\bar{U}_{\theta} \in \left[-U_{\theta}^{+}, U_{\theta}^{+}\right]$, $\bar{U}_{\psi} \in \left[-U_{\psi}^{+}, U_{\psi}^{+}\right]$. For small rotational uncertainty $\bar{U}_{\psi} \simeq \bar{U}_{\theta} \simeq \bar{U}_{\phi} \simeq 0^{\circ}$, the uncertain rotational component of $\mathrm{Rot}(z, \bar{U}_{\phi})\mathrm{Rot}(y, \bar{U}_{\theta})\mathrm{Rot}(x, \bar{U}_{\psi})$ becomes $\begin{bmatrix} 1 & -\bar{U}_{\phi} & \bar{U}_{\theta} \\ \bar{U}_{\phi} & 1 & -\bar{U}_{\psi} \\ -\bar{U}_{\theta} & \bar{U}_{\psi} & 1 \end{bmatrix}$ and the YOLO's four vertices $l_{i,3}^{r,\phi\theta\psi}, i = 1, \ldots, 4$ lay within

$$l_{i,3}^{r,\phi\theta\psi} \in l_{i,3}^{\circ} + \begin{bmatrix} -\bar{U}_{\phi} l_{i,3}^{\circ,2} \\ \bar{U}_{\phi} l_{i,3}^{\circ,1} \\ 0 \end{bmatrix} + \begin{bmatrix} \bar{U}_{\theta} l_{i,3}^{\circ,3} \\ -\bar{U}_{\psi} l_{i,3}^{\circ,3} \\ \bar{U}_{\psi} l_{i,3}^{\circ,2} - \bar{U}_{\theta} l_{i,3}^{\circ,1} \end{bmatrix}. \tag{1}$$

The camera's pose uncertainty or given that $\| \Delta\mathcal{P}_t \| \leq \left[U_x, U_y, U_z, U_{\psi}^{+}, U_{\theta}^{+}, U_{\phi}^{+}\right]^{\top}$ is computed by:

1: Handle $(\psi \times \theta)$-uncertainty by computing $\mathrm{Pyr}_k^{\psi\theta}\left[R_p, \mathcal{C}, \mathcal{Y}^{\circ}, \mathcal{P}^{\circ}, (U_{\theta}^{+}, U_{\psi}^{+})\right]^{\top}$.

2: Handle (ϕ)-uncertainty by computing $\widetilde{\mathrm{Pyr}}_k^{\phi}\left[R_p, \mathcal{C}, \mathcal{Y}^{r,\psi\theta}, \mathcal{P}^{\circ}, U_{\phi}^{+}\right]^{\top} = \widetilde{\mathrm{Pyr}}_k^{\psi\theta\phi}$

3: Handle translational uncertainty by computing $\mathrm{Tran}_k\left[\widetilde{\mathrm{Pyr}}_k^{\psi\theta\phi}, \mathcal{U}\right]^{\top}$

The rotational uncertainty (Steps 1 through 2) along with the translational uncertainty (Step 3) demand the computation of the convex hull of a widened pyramid with: a) one parallelepiped at its vertex-0, and b) four parallelepipeds at its other vertices for every ϕ-angle. If Φ distinct cases are assumed for the ϕ-uncertainty range, then computing the convex hull of $((\Phi \times 4 \times 8) + 8)$ points is required. This task has minimal computational burden. Similarly, the facets of the FoV-induced uncertainty appear in the right part of Figure 3, where the departure from the typical pyramid is apparent.

Assume that Ψ (Θ) distinct cases are considered for $\psi \in \left[-U_{\psi}^{+}, U_{\psi}^{+}\right], (\theta \in \left[-U_{\theta}^{+}, U_{\theta}^{+}\right]$; the object is within the convex hull of all rotational uncertain points $4 \times \Phi \times \Theta \times \Psi$ with an orthogonal parallelepiped (8 vertices) applied in each one of these points. Thus at every step rather than computing the aforementioned three steps, the algorithm can be formed by computing the 3D-convex hull of: a) $(4 \times \Phi \times \Theta \times \Psi \times 8)$ 3D-points, plus b) the eight 3D-vertices of the parallelepiped for the base $\mathrm{Pyr}^{\circ}_{\mathrm{vertex}_0}$.

 doi:10.20944/preprints202604.0178.v1

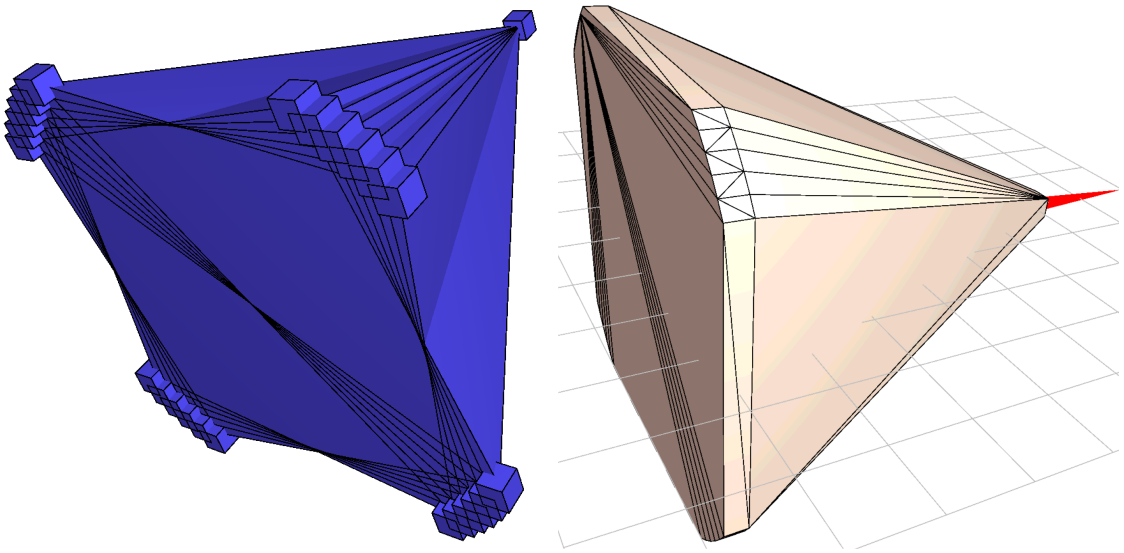

Figure 3. Camera pose uncertainty primitive elements and facets.

3. Gen6D Zero-Shot Target Acquisition

Interacting with the environment requires the relative pose of the identified object and the RGB monocular camera observing this object[19]. Essentially the relative 6D position and orientationis sought using methods related to the object's features [10,20,21]

For the 3D bounding box identification, the Gen6D [10] method is utilized. YOLOv8 identifies and classifies the selected object, and based on the YOLO bounding box the area around the object is cropped and then Gen6D operates on the appropriate focused remaining region of the image that includes the object. Gen6D provides an estimate of the distance between the camera and the object by normalizing the pixel count with the nominal pixel counts (for a given zoom-factor) of the identified object (during the training phase). YOLO has been used extensively for UAV identification [22]and successfully classified the object with a 2D-box of approximately as low as 60×60 pixels.

The Gen6D method is trained with the identified object, using a point cloud (object and surroundings) and the COLMAP-library[23]. Subsequently, the object points are isolated and used as input to the estimator. The library also produces a set of estimated camera positions in relation to the point cloud axis system with their equivalent images of the item that correspond to the camera pose from the training images. This information is also used in the initial experimentation phase, where the initial camera pose estimation is set as the closest one matching the object. It is worth noting that the Gen6D method can estimate poses for unseen objects without requiring a retraining.

Problem Formulation

Given an RGB image I, the goal is to estimate its pose $\mathbf{T} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix}$, where $\mathbf{R} \in SO(3)$ is the rotation matrix, and $\mathbf{t} \in \mathbb{R}^3$ is the translation vector. The classified object is represented by its N-3D-point cloud $\mathcal{X} = \{\mathbf{x}_i \in \mathbb{R}^3, i = 1, \ldots, N\}$.

The projection of each 3D-point onto the image plane is $\mathbf{u}_i = \mathbf{K}(\mathbf{R}\mathbf{x}_i + \mathbf{t})$, where $\mathbf{K}$ is the camera intrinsic matrix.

Gen6D initially defines its 2D-Region of Interest $\mathbf{b} = (x^\circ, y^\circ, \Delta W, \Delta H)$, while a neural network assigns to each pixel $\mathbf{u}$ a canonical coordinate, $\mathbf{c}$, $f_\theta(I, \mathbf{u}) \rightarrow \mathbf{c} \in \mathbb{R}^3$, thus establishing a mapping $\mathbf{u} \leftrightarrow \mathbf{x}$.

Hence, given correspondences for all N-pixels corresponding to the object, $\{(\mathbf{u}_i, \mathbf{x}_i)\}_{i=1}^N\}$ the object's pose is estimated by minimizing the re-projection error

$$\min_{\mathbf{R},\mathbf{t}} \sum_{i=1}^{N} \|\mathbf{u}_i - \pi(\mathbf{R}\mathbf{x}_i + \mathbf{t})\|^2. \tag{2}$$

 doi:10.20944/preprints202604.0178.v1

This is typically solved using a Perspective-n-Point (PnP) algorithm[24], or $(\mathbf{R}, \mathbf{t}) = \text{PnP}(\{\mathbf{x}_i, \mathbf{u}_i\}, \mathbf{K})$.

The eight 3D-bounding box corners $\mathbf{X}_{\text{3D-box}} = \{\mathbf{x}_j^c\}_{j=1}^8$ are transformed into camera coordinates $\mathbf{X}_j^{\text{cam}} = \mathbf{R}\mathbf{x}_j^c + \mathbf{t}$ followed by their projection $\mathbf{u}_j = \pi(\mathbf{X}_j^{\text{cam}})$.

During the Gen6D learning process, it minimizes: a) the coorespondence loss $\mathcal{L}_{\text{corr}} = \sum_{i=1}^N \|\hat{\mathbf{c}}_i - \mathbf{c}_i^*\|_1$, b) the reprojection Loss $\mathcal{L}_{\text{reproj}} = \sum_{i=1}^N \left\|\mathbf{u}_i - \pi(\hat{\mathbf{R}}\mathbf{x}_i + \hat{\mathbf{t}})\right\|_1$, and c) the Pose-loss $\mathcal{L}_{\text{pose}} = \|\hat{\mathbf{R}} - \mathbf{R}^*\|_F + \|\hat{\mathbf{t}} - \mathbf{t}^*\|_2$.

The Gen6D method is robust to: i) object's shape, ii) canonical coordinate embeddings and iii) viewpoint-invariant features, using synthetic data, augmentation, and feature learning strategies.

4. Constrained Visual Servoing

The purpose of the visual servoing concept is to get the UAV closer to the eight vertices related to the object's 3D-bounding box, subject to any constraints related to its motion.

We assume a camera mounted on a 3-axis gimbal, which is subsequently mounted on a UAV. It is also assumed that the attached gimbal will not be used for any yaw maneuvers. The standard Robot Operating system's REP-103 conventions are used, whereby in the initiation phase of every experiment the camera's forward facing z-axis is being originally aligned with the UAV's body frame forward x-axis.

Post take-off, the drone at its hovering position is located at $[0, 0, z^\circ]^\top$. It should be noted that in order to reduce the measurement error GPS-RTK is used for the xy-components, while the drone's altitude is measured via a barometer. The center of the 3D-box is at $[x^b, y^b, z^b]^\top$ resulting in a distance from the drone $d_e = \sqrt{x^{b2} + y^{b2} + (z^\circ - z^b)^2}$; this information is used to adjust the zoom-factor (focal length) so that: a) the 3D-box remains within the FoV, and b) YOLO has enough resolution to classify the object. Essentially, the estimated distance, $\hat{d}_e$ is a function of the manual zoom $\hat{d}_e(f)$, while the adjusted manual focus also depends on it The drone initially adjusts its yaw, so that it points to the identified object, resulting in a yaw angle $\theta^\circ = \text{atan2}(y_b, x_b)$, while the gimbal is pitched at $\psi^\circ = \text{asin}(\frac{z^\circ - z^b}{d_e})$; the drone's roll is only used to account for any wind gusts affecting the drone. The gimbal's IMU negates any roll r^d and pitch p^d jittering of the frame owing to external disturbances' rejection as in $\psi^g = \psi^\circ - p^d$ and $\phi^g = -r^d$.

The UAV adjusts its yaw so that it is aligned with the far-object , with initial maximum zoom factor of 36x in our case and Manual Focus adjusted based on the computed distance d_e. Indicative target detections for various d_e-distances appear in Figure 4.

Figure 4. Detected object at d_e=10m (left) and $d_e = 90$m (right).

The feature points, $\mathbf{s}^\star$ correspond to the eight vertices of the 3D-bounding box provided by Gen6D projected into the image frame, or $\mathbf{s}^\star = [s_1^*, \ldots, s_8^*]_{8\times 2}^\top = \left[\begin{bmatrix} x_1^* \\ y_1^* \end{bmatrix}, \ldots, \begin{bmatrix} x_8^* \\ y_8^* \end{bmatrix}\right]^\top$. If the drone is aligned with the target, then the drone's forward velocity, v_z, is towards the object and minimizes $\sqrt{(x^b)^2 + (y^b)^2}$. The drone attempts to adjust its velocity $\mathbf{v} = \left[v_x^d, v_y^d, v_z^d, \omega_z^d\right]^\top$ so that the current points (in the image plane) $\mathbf{s} = [s_1, \ldots, s_8]^\top = \left[\begin{bmatrix} x_1 \\ y_1 \end{bmatrix}, \ldots, \begin{bmatrix} x_8 \\ y_8 \end{bmatrix}\right]^\top$, converge to the feature points $\mathbf{s}^\star$; the current points correspond to the vertices of a fictitious orthogonal box placed at a distance

 doi:10.20944/preprints202604.0178.v1

$d_f \ll d_e$; this fictitious box has the same normalized dimensions like the one provided by Gen6D and the same perspective. It should be noted that because of the initial rotation towards the object $\omega_z^d \simeq 0$, typically is included in the computations to account for any environmental disturbances in the drone's z-axis.

For every current point s_i there is a relationship between the UAV's translation and orientation speed $\mathbf{v}_{6\times1} = \begin{bmatrix} v_x & v_y & v_z & \omega_x & \omega_y & \omega_z \end{bmatrix}^\top$ and its velocity, as

$$\begin{bmatrix} \dot{x}_i \\ \dot{y}_i \end{bmatrix} = \begin{bmatrix} -\frac{1}{d_{e,i}} & 0 & \frac{x_i}{d_{e,i}} & x_i y_i & -(1+x_i^2) & y_i \\ 0 & -\frac{1}{d_{e,i}} & \frac{y_i}{d_{e,i}} & (1-y_i^2) & -x_i y_i & -x_i \end{bmatrix} \mathbf{v} = \mathbf{L}_{i(2\times6)}\mathbf{v}_{(6\times1)}, \tag{3}$$

where $d_{e,i}$ is the distance between the drone's camera and the ith feature point; this is under the assumption that the camera is looking at the object.

Assuming that the gimbal provides an opposite pitch and roll motion to the drone's movement, or $\omega_x \simeq \omega_y \simeq 0$, leading to their exclusion, then (3) is transformed to:

$$\begin{bmatrix} \dot{x}_i \\ \dot{y}_i \end{bmatrix} = \begin{bmatrix} -\frac{1}{d_{e,i}} & 0 & \frac{x_i}{d_{e,i}} & y_i \\ 0 & -\frac{1}{d_{e,i}} & \frac{y_i}{d_{e,i}} & -x_i \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ v_z \\ \omega_z \end{bmatrix} = \mathbf{L}_{i,r(2\times4)}\mathbf{v}_{r(4\times1)}. \tag{4}$$

The drone's reduced velocity, $\mathbf{v}_r$ can be computed for the eight feature points, as

$$\mathbf{v}_r = -\lambda \mathbf{L}_s^+ (\mathbf{s}^* - \mathbf{s}) = -\lambda \begin{bmatrix} \mathbf{L}_{1,r} \\ \vdots \\ \mathbf{L}_{8,r} \end{bmatrix}^+_{(16\times4)} \begin{bmatrix} (x_1^* - x_1) \\ (y_1^* - y_1) \\ \vdots \\ (x_8^* - x_8) \\ (y_8^* - y_8) \end{bmatrix}_{(16\times1)} \tag{5}$$

where $\lambda > 0$, $\mathbf{L}_s$ is the Interaction (image Jacobian) matrix, $(\cdot)^+$ corresponds to the Moore–Penrose pseudoinverse.

It should be noted that Gen6D computes the distances $d_{e,i}$ of the camera end point from the target object 3D bounding box estimated vertices.

The terms may also be estimated using the drone's GPS-coordinates, if this is available with a good precision. This assists in adjusting the zoom factor (through the camera's focal length) and the resulting manual focus. The servoing relies on the Visual Servoing Platform (ViSP)[25]. This platform computes the pseudo-inverse matrix in (5) in an efficient and reliable manner.

5. Simulation & Experimental Studies

The validity of Gen6D for computing the distance d_e from the object was accomplished with a stationary camera. The cap was placed at various distances $d_e \in \{5, \ldots, 100\}$m, and the dynamic (absolute) camera zoom was set at (36×1.6). Figure 5 presents the actual (red line) versus the computed distances (blue line) from Gen6D.

It should be noted that YOLO v8 was capable of identifying the object in all distances (sample photos are shown in Figure 4), if the Manual Focus was properly adjusted, since this tuning was crucial at large distances and zoom factors. YOLO's confidence varied between 0.98 (at small distance) down to 0.59 (when $d_e = 100$m), primarily attributed to the number of pixels involved in the decision making process and the object. As an example, when another UAV was flying at distances close to 150m, YOLO could identify this as a drone, making this method useful for geofencing purposes. Prior to forwarding these images to Gen6D, YOLO was used in all image processing software.

 doi:10.20944/preprints202604.0178.v1

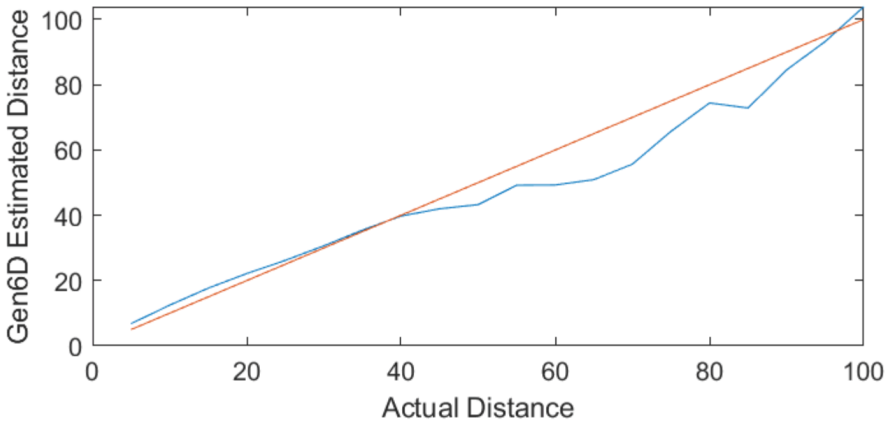

Figure 5. Estimated (Gen6D) vs. Actual Distance (Camera-object).

5.1. Drone and Gimbal Utilization

A quadrotor drone was manufactured at NYUAD, as shown in Figure 6. This drone has a maximum payload of 10kg, and maximum takeoff weight of 20kg. This drone can easily lift the utilized Gremsy Pixy gimbal and the attached to it Harrier 36X zoom block camera. The camera takes photos at HD-resolution and has a 36× dynamic zoom range. The utilized carbon-fiber 22in propellers provide significant thrust for over 45 minutes of flight time with the onboard 6S 44Ah LiPo batteries. The drone communicates with the base station using the MAVLink protocol, its FCU is the Pixhawk Orange with ArduCopter firmware, while all interprocess communications are carried out using the Robot Operating System (ROS). The position in X-Y plane is the EKF fusion of a Here+ GPS sensor and the internal IMU, while for the altitude the internal Barometer is fused with IMU measurements. The onboard companion computer was chosen to be an Intel NUC Enthusiast 11 with an i7 processor and NVIDIA RTX2060 graphics card.

Figure 6. NYUAD drone and gimbal.

5.2. Small Distance Experiment

The drone was rotating at an upward facing hemispherical path, defined around the cup at radius of $d_e = 2$m and centered at the origin of the x_0, y_0, z_0 axis system of Figure 2. The waypoints were parametrically configured to lie every 30° in elevation angle, while the azimuth angle increased progressively every 15°. While hovering at each waypoint, the absolute zoom factor varied $f \in \{3.6, 5\}$ and consecutive RGB-images were taken resulting in a total of $72 = \frac{360}{30} \times \frac{90}{15}$ frames. The selected object corresponds to a coffee-mug and sample frames appear in Figure 7 for various zoom factors and 75° azimuth angle.

 doi:10.20944/preprints202604.0178.v1

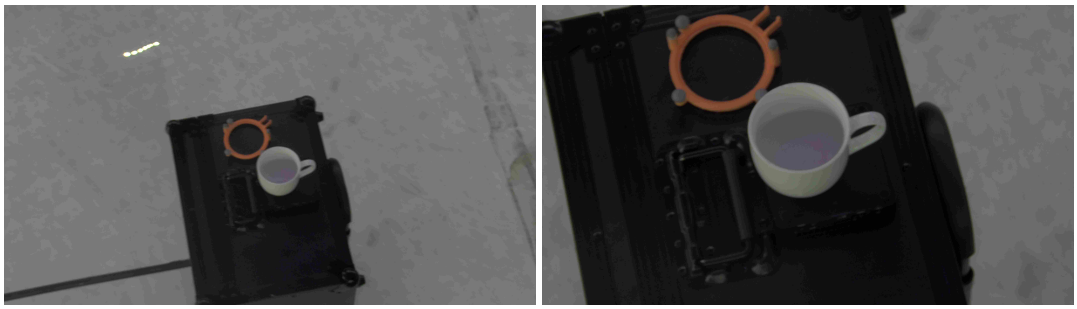

Figure 7. Visual object frames with various zoom factors.

The NYUAD-drone was used in these experiments and its position was inferred through a Vicon-based Motion Capture System (MoCaS). The drone hovering resulted in $[U_x, U_y, U_z]^\top = [4, 4, 5]^\top$cm positioning uncertainty, while the attitude uncertainty was $\left[U_\phi^+, U_\theta^+, U_\psi^+\right]^\top = [1.5°, 1.5°, 1°]^\top$, as shown in Figure 8.

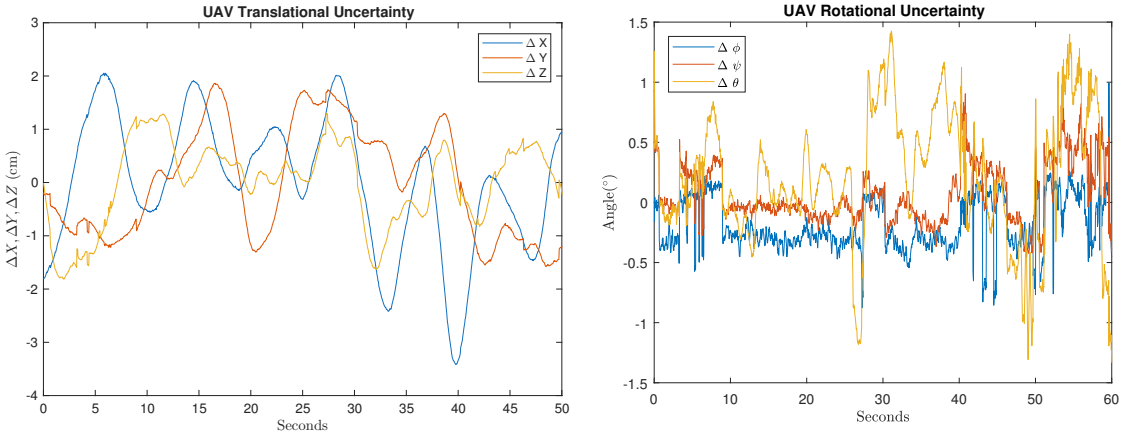

Figure 8. UAV Pose Uncertainty around the steady state for each axis.

The drone's pose was also recorded using the aforementioned high accuracy MoCaS. The error between Gen6D's d_e translation estimation and the ground-truth provided by the MoCaS is shown at the left part of Figure 9 for a 2× zoom-factor. Given the camera's coordinates $[x^e, y^e, z^e]^\top$ and its orientation $[\phi^e, \theta^e, \psi^e]^\top$ provided by Gen6D, the projection of the camera's forward looking z-axis by ViSP and the object's plane passing through its center is shown in the right part of Figure 9. It is apparent that Gen6D provides a relatively accurate zero-shot estimate of the camera-pose with respect to the object.

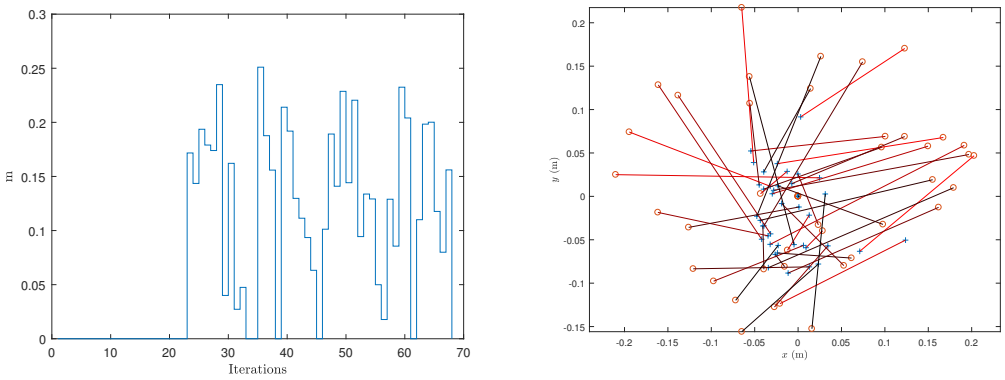

Figure 9. Camera translation error (left) and camera forward-axis projection to the object's plane (right).

 doi:10.20944/preprints202604.0178.v1

Owing to the drone's vibrations at hover and due to the optimization attempted by Gen6D, there are instances (less than 8%) resulting in significant error of the camera's apex, as shown in Figure 10. Blue colored are the correct pyramids induced by YOLO (see Figure 2), while the golden colored ones correspond to large successive distances $\|d_e(k+1) - d_e(k)\|$. Given that the drone can move with a maximum velocity $\max(\|\mathbf{v}\|) \leq 2\text{m/s}$ in 3D space and given that the frame rate is 3 FpS, then the previous quantity can be bounded by $\|2/3\|$m.

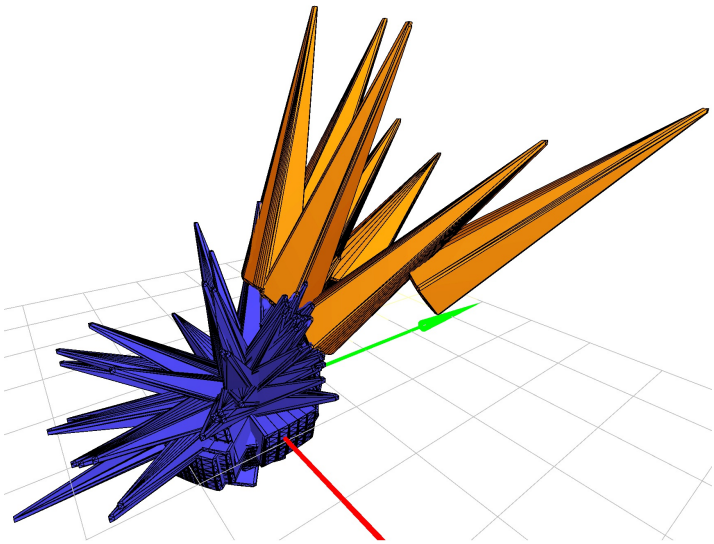

Figure 10. Camera Apex and Assorted Pyramids.

The intersection of pyramids, given the camera's pose from the Gen6D (MoCaS) appears as light (dark) blue while the cap is also shown in Figure 11. The left (right) part corresponds to the $2\times$ $(5\times)$ zoom factor. As expected, the volume of the convex hull withe the same uncertainty, obtained through MoCaS is significantly smaller than the one for Gen6D. This is because the there is a significant pose error (see Figure 9) in Gen6D. Overall, this is a rather computationally demanding procedure and was performed in the CGAL library [26]. Typically Gen6D provides an additional 120% error (volume increase) compared to the MoCaS. It should be noted that the convex hull computation of the detected object is useful in grasping algorithms [27,28].

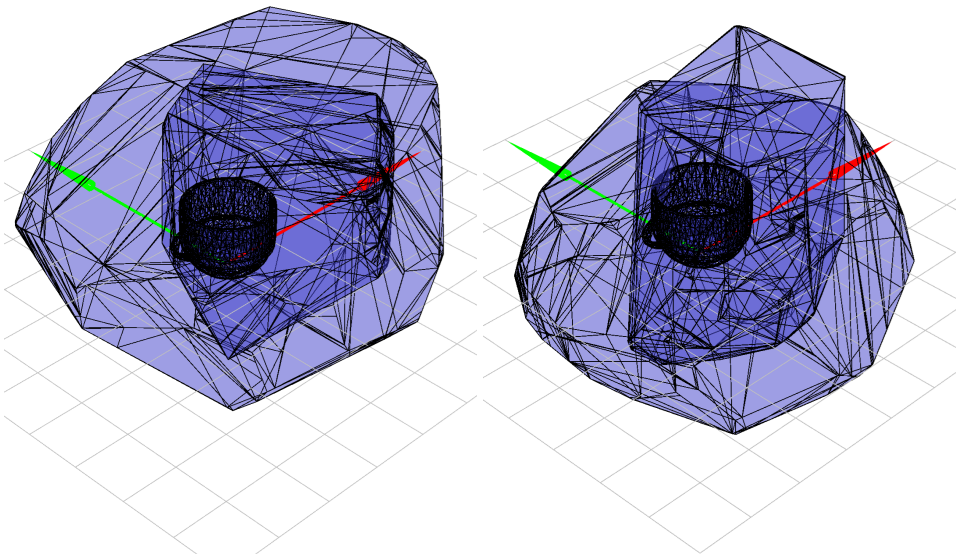

Figure 11. Object volume estimation using "Pyramid" Intersection left (right) $2\times$ $(5\times)$ zoom) [light (dark) blue estimation using MoCaS (Gen6D)].

 doi:10.20944/preprints202604.0178.v1

The convergence of the volume of the computed convex hull for the noted 2× and 5× zoom factors appear in Figure 12, where, as expected, the MoCaS results are better than the ones computed from Gen6D. This is due to the inaccurate computed pose of the camera. Nevertheless, the volume estimation of Gen6D is not considered to drift significantly for grasping tasks in comparison to the ground-truth, taking into account the zero-shot nature of experimentation. Furthermore, the Gen6D-results with larger zoom are slightly better than the ones with the smaller zoom, attributed mainly to a better approximation of the camera's pose.

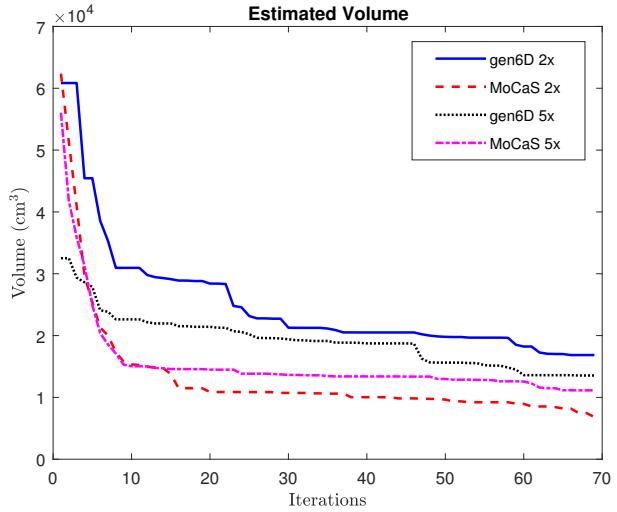

Figure 12. Object convex-hull volume reduction for various zoom factors.

It should be noted that for improved translation estimation, the camera's orientation plays a significant role in the speed of convergence. Assuming congruent cones (with circular cross section), the intersection of neighboring cones is rapidly reduced, when the angle between them is close to 90°, as shown in Figure 13 for the same angle parameters α and β as in [29]. α corresponds to the cone's generator angle (related to the opening of the FoV), while β is related to the angle between these cones, separated by $d\cos(\beta)$. In general d is related to d_e, and for large zoom-factors (small α-angles), where the UAV is at large distances from the object (large d-distance), the convex hull is accurately computed despite any significant pose-errors. For a drone following the suggested visual-servoing scheme, $\beta \simeq 0°$, indicating that the convex hull computation depends very heavily on the computed (by Gen6D) camera pose.

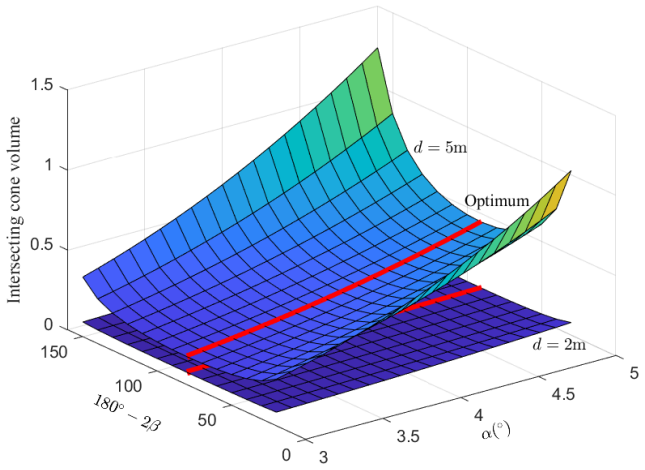

Figure 13. Intersecting congruent-cone volume reduction vs. α and β angles [29].

 doi:10.20944/preprints202604.0178.v1

5.3. Large Distance Simulation Study

Initially the team derived preliminary indoor experimental results. However due to the state-of-war in the Gulf region, the team was unable to re-derive these results outdoors, since UAVs were not allowed to fly in UAE. Instead, the team reports a sim-to-real case, where the drone is simulated in Gazebo 11. The varying zoom camera is simulated using a custom made plug-in, whereby the sensor focal length variation given by OEM is used to compute the relevant FoV and pass it over to the Gazebo camera sensor plugin dynamically. Similarly a secondary plug-in emulating the DoFs of a typical three DoF gimbal was developed to provide Roll-Pitch-Yaw stabilization mid-flight. The drone's location was derived with an IMU and a GPS-RTK mechanism capable of taking measurements at five times per second. If UAVs are allowed to fly in UAE, and under the assumption of this paper's acceptance, the team will perform and provide the cross-verified experimental results.

The object was placed at a 100m distance $(d_e = 100)$m from the drone, shown in Figure 14; this allows the cup to be classified using YOLOv8, despite its low resolution 80×55 pixels (see Figure 4). The use of a global shutter camera, like the Harrier 36X, in comparison with a rolling shutter one is rather obvious in a UAV-system that has several vibrations. Furthermore, the need to switch in manual focus dependent on the estimated distance d_e by Gen6D is critical since the autofocus option can take up to 2s to focus the Region-of-Interest(ROI) and typically requires a stationary background. Similarly the manual focus lasts 750ms and narrowly depends on d_e; for this reason rather than using a MF continuous adjustment, we adjust it in batches according to the drone amplitude. In the ensuing simulation studies, the autofocus option was disabled while the used MF was dependent in a batch manner on d_e.

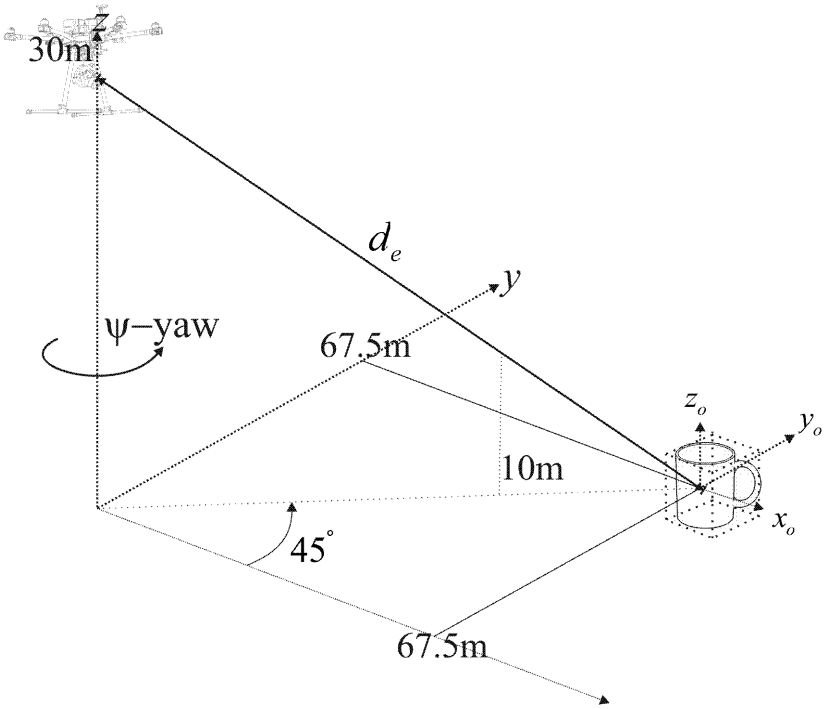

Figure 14. Experimental study.

Initially the drone starts at height $z_d = 30$m, while the object is located at $[x^\circ, y^\circ, z^\circ]^\top = [67.45, 67.45, 0]^\top$m w.r.t. the drone's body frame axis system at initialization. This initial placement is assumed to be a-priori known only for the first acquired camera frame to contain the object. This is done by the drone adjusting its yaw by 45° and the gimbal adjusting its pitch angle by 16.93° for the given $[x^\circ, y^\circ, z^\circ]^\top$. The presence of the object within the camera FoV results in YOLO detection and Gen6D estimation. Finally, the visual servoing pipeline is called to initiate motion. From that point onward the object's a-priori known position is not utilized anymore, as the visual servoing pipeline takes over to bring the UAV closer to the target object.

 doi:10.20944/preprints202604.0178.v1

Subsequently, following consecutive object detections, pose estimations and servoing loops, the drone descends towards the target using this angle and when it reaches 10m altitude, it switches its zoom from 36× to 10× to continue its trajectory. The drone at that time is at a distance $d_{e'} = \frac{10}{30} \times 100$m from the object. The visual servoing is found to maintain successful tracking and proximity velocity commands even at $d_e \leq 5$m.

Figure 15 shows several 3D-bounding boxes, derived from Gen6D, at distances $d_e = [100, 80, 60, 40, 20, 10]$m, ordered from top-left to bottom-right. YOLOv8 managed to classify the object at all d_e distances, with higher confidence as distance decreased. Furthermore, when the distance reduced from 40-to-30m, the zoom-factor changed resulting in a larger FoV (from $1.77° \times 1.11°$ to $6.35° \times 4.00°$) and an object-magnification (see at bottom row in comparison middle to far east image). We should state that the use of a Gimbal with good shock absorption and orientation error less than 1° is essential for this study [30,31]. We should state that when emulating wind-gusts the camera with large zoom $(36\times)$ failed to maintain the object within its FoV when $d_e \simeq 50$m, necessitating in this case a smaller focal length (zoom factor reduced to $10\times$). In this case, the object remained within the camera's FoV but YOLO's confidence was reduced since a few pixels were used to locate it.

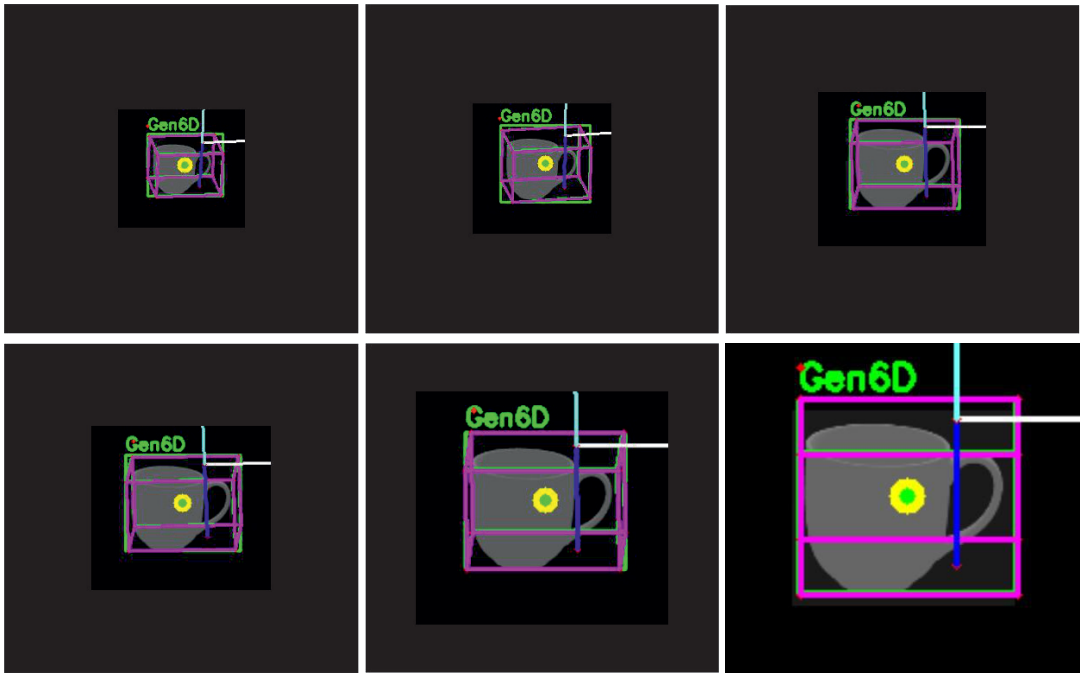

Figure 15. Gen6D-based 3D-bounding boxes Top (Bottom) row: 3D-boxes for $d_e = [100, 80, 60]^\top$m ($d_e = [40, 20, 10]^\top$m).

The drone's trajectory is very similar to the reference one of Figure 14 , while the euclidean distance evolution from the object is depicted in Figure 16. On purpose, the units on the horizontal is in iterations indicating that this is a simulation study. To account for most of the aerodynamic effects, the simulation study is quite detailed and resulted in a forward velocity of 1.8m/min. In the experimental testing this number is closer to 2m/s. As expected, the constrained visual servoing scheme brings the drone closer to the target, while accounting for any encountered jittering owing to vibrations.

The gimbal's pitch/roll/yaw angles where found to exhibit minor deviations of less than 0.5°, mostly owing to the simulated wind gusts and flight stack positioning uncertainties.

Overall, the identification and visual servoing shows potential in approaching long distanced objects. As long as these objects remain within the camera's FoV, the YOLO and Gen6D pipelines are called resulting in computing the 3D-bounding box and commanding proper visual servoing frame velocities to approach the objects' center.

 doi:10.20944/preprints202604.0178.v1

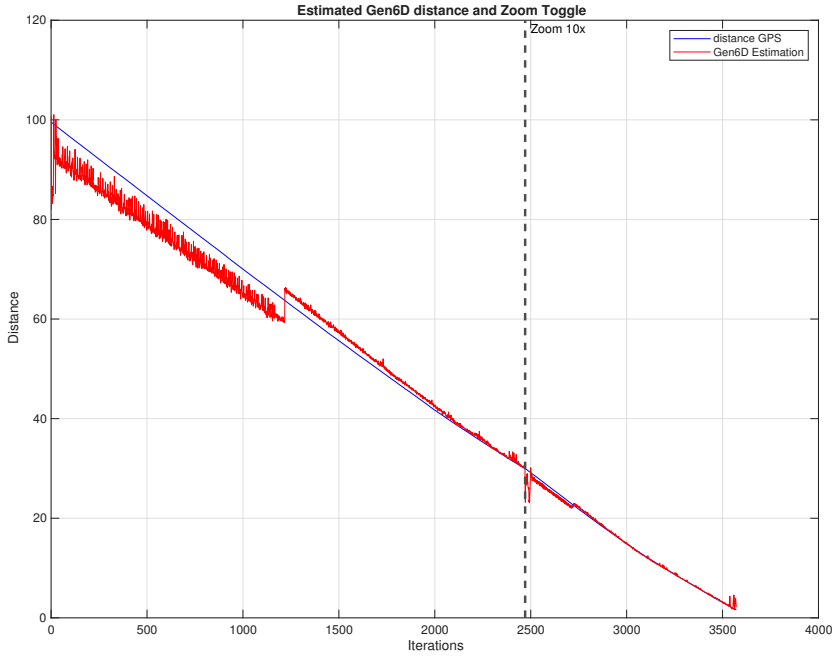

Figure 16. Experimental Drone-to-Object Distance.

6. Conclusions

The development of a zero-shot (single image) visual-based scheme for detecting and classifying objects at large distances using a UAV is presented. Rather than relying on 2D-YOLO inference, the AI-based zero-shot Gen6D algorithm infers the 3D-vertices of the object's bounding box. Furthermore, it computes the relative camera's pose with respect to the detected object. A visual servoing algorithm is employed to guide the drone closer to the object. The camera's zoom factor is adjusted to account for the distance reduction and the need to maintain the object within the camera's FoV, while there is a manual focus. The drone initially rotates towards the object and approaches it, while satisfying the inherent constraints. Experimental and simulation results validate the suggested approach.

Funding: The authors declare that no funds, grants, or other support were received during the preparation of this manuscript. This work is supported in part by the NYUAD Center for Artificial Intelligence and Robotics (CAIR), funded by Tamkeen under the NYUAD Research Institute Award CG010.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding authors. The raw data supporting the conclusion of this article will be made available by the authors, without undue reservation.

Acknowledgments: This work was partially performed at the Kinesis CTP facility, New York University Abu Dhabi, Abu Dhabi, 129188, United Arab Emirates.

Author Contributions: Conceptualization, A.Ts., N.E. and A.Tz.; Methodology, A.Ts., N.E. and A.Tz.; Software, A.Ts. and N.E.; Hardware, N.E.; Validation, A.Ts., N.E. and A.Tz.; Formal analysis, A.Ts., N.E. and A.Tz.; Investigation, A.Ts., N.E. and A.Tz.; Resources, A.Ts., N.E. and A.Tz.; Experimentation, A.Ts. and N.E.; Data curation, A.Ts., N.E. and A.Tz.; Writing—original draft preparation, A.Ts., N.E. and A.Tz.; Writing—review and editing, A.Ts., N.E. and A.Tz.; Visualization, A.Ts., N.E. and A.Tz.; Supervision, A.Tz.; Project administration, A.Tz.; Funding acquisition, A.Tz. All authors have read and agreed to the published version of the manuscript.

Conflicts of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Diwan, T.; Anirudh, G.; Tembhurne, J.V. Object detection using YOLO: Challenges, architectural successors, datasets and applications. *Multimedia Tools and Applications* **2023**, *82*, 9243–9275.

 doi:10.20944/preprints202604.0178.v1

2. Maiettini, E.; Pasquale, G.; Rosasco, L.; Natale, L. On-line object detection: a robotics challenge. *Autonomous Robots* **2020**, *44*, 739–757.
3. Xia, C.; Zhao, W.; Han, H.; Tao, Z.; Ge, B.; Gao, X.; Li, K.C.; Zhang, Y. MonoSAID: Monocular 3D Object Detection based on Scene-Level Adaptive Instance Depth Estimation. *Journal of Intelligent & Robotic Systems* **2024**, *110*, 2.
4. Mo, X.; Sajid, U.; Wang, G. Stereo frustums: A Siamese Pipeline for 3D Object Detection. *Journal of Intelligent & Robotic Systems* **2021**, *101*, 6.
5. Fidler, S.; Dickinson, S.; Urtasun, R. 3D object detection and viewpoint estimation with a deformable 3D cuboid model. *Advances in neural information processing systems* **2012**, *25*.
6. Vo, X.T.; Jo, K.H. Accurate Bounding Box Prediction for Single-Shot Object Detection. *IEEE Transactions on Industrial Informatics* **2022**, *18*, 5961–5971.
7. Verykokou, S.; Ioannidis, C. An overview on image-based and scanner-based 3D modeling technologies. *Sensors* **2023**, *23*, 596.
8. Chen, L.; et al. Real-time photogrammetry based on parallel architecture for 3D applications **2024**.
9. Murtiyoso, A.; Pellis, E.; Grussenmeyer, P.; Landes, T.; Masiero, A. Towards semantic photogrammetry: generating semantically rich point clouds from architectural close-range photogrammetry. *Sensors* **2022**, *22*, 966.
10. Liu, Y.; Wen, Y.; Peng, S.; Lin, C.; Long, X.; Komura, T.; Wang, W. Gen6D: Generalizable Model-Free 6-DoF Object Pose Estimation from RGB Images. In Proceedings of the European Conference on Computer Vision. Springer, 2022, pp. 298–315.
11. Liu, Y.; Wang, L.; Liu, M. YOLOStereo3D: A step back to 2D for efficient stereo 3D detection. In Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2021, pp. 13018–13024.
12. Dai, A.; Chang, A.X.; Savva, M.; Halber, M.; Funkhouser, T.; Nießner, M. ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In Proceedings of the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 5828–5839.
13. Xu, J.; Ma, Y.; He, S.; Zhu, J. 3D-GIoU: 3D generalized intersection over union for object detection in point cloud. *Sensors* **2019**, *19*, 4093.
14. You, Y.; Ye, Z.; Lou, Y.; Li, C.; Li, Y.L.; Ma, L.; Wang, W.; Lu, C. Canonical voting: Towards robust oriented bounding box detection in 3D scenes. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 1193–1202.
15. Bansal, K.; Rungta, K.; Zhu, S.; Bharadia, D. Pointillism: Accurate 3D bounding box estimation with multi-radars. In Proceedings of the Proceedings of the 18th Conference on Embedded Networked Sensor Systems, 2020, pp. 340–353.
16. Zhang, Y.; Hang, J.; Zhu, T.; Xiangbo, L.; Wu, R.; Peng, W.; Tian, D.; Sun, Y. FunctionalGrasp: Learning Functional Grasp for Robots Via Semantic Hand-Object Representation. *IEEE Robotics and Automation Letters* **2023**, pp. 1–8.
17. Zhao, X.; Jia, H.; Ni, Y. A novel three-dimensional object detection with the modified You Only Look Once method. *International Journal of Advanced Robotic Systems* **2018**, *15*, 1729881418765507.
18. Diazzi, L.; Attene, M. Convex polyhedral meshing for robust solid modeling. *ACM Transactions on Graphics (TOG)* **2021**, *40*, 1–16.
19. Thalhammer, S.; Bauer, D.; Hönig, P.; Weibel, J.B.; Garcia-Rodriguez, J.; Vincze, M. Challenges for Monocular 6D Object Pose Estimation in Robotics. *IEEE Transactions on Robotics* **2024**, *40*, 4065–4084.
20. Di Felice, F.; Remus, A.; Gasperini, S.; Busam, B.; Ott, L.; Thalhammer, S.; Tombari, F.; Avizzano, C.A. InstantPose: Zero-Shot Instance-Level 6D Pose Estimation From a Single View. *IEEE Robotics and Automation Letters* **2025**.
21. Di Felice, F.; Remus, A.; Gasperini, S.; Busam, B.; Ott, L.; Tombari, F.; Siegwart, R.; Avizzano, C.A. Zero123-6D: Zero-shot Novel View Synthesis for RGB Category-level 6D Pose Estimation. In Proceedings of the 2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2024, pp. 14204–14211.
22. Tsoukalas, A.; Xing, D.; Evangeliou, N.; Giakoumidis, N.; Tzes, A. Deep learning assisted visual tracking of evader-UAV. In Proceedings of the 2021 International Conference on Unmanned Aircraft Systems (ICUAS), 2021, pp. 252–257.
23. Sambugaro, Z.; Orlandi, L.; Conci, N.; et al. 3D reconstruction methods in industrial settings: a comparative study for COLMAP, NeRF and 3D Gaussian Splatting. In Proceedings of the Ceur Workshop Proceedings, 2024, Vol. 3762, pp. 212–217.

 doi:10.20944/preprints202604.0178.v1

24. Zhou, L.; Kaess, M. An efficient and accurate algorithm for the perspecitve-n-point problem. In Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2019, pp. 6245–6252.
25. Marchand, É.; Spindler, F.; Chaumette, F. ViSP for visual servoing: a generic software platform with a wide class of robot control skills. *IEEE Robotics & Automation Magazine* **2006**, *12*, 40–52.
26. Alliez, P.; Fabri, A. CGAL: the computational geometry algorithms library. In *ACM SIGGRAPH 2016 Courses*; 2016; pp. 1–8.
27. Roa, M.A.; Suárez, R. Grasp quality measures: review and performance. *Autonomous robots* **2015**, *38*, 65–88.
28. Miller, A.T.; Allen, P.K. Graspit! a versatile simulator for robotic grasping. *IEEE Robotics & Automation Magazine* **2004**, *11*, 110–122.
29. Beyer, W.; Fawcett, L.; Mauldin, R.D.; Swartz, B.K. The volume common to two congruent circular cones whose axes intersect symmetrically. *J. Symb. Comput.* **1987**, *4*, 381–390.
30. Ma, M.Y.; Huang, Y.H.; Shen, S.E.; Huang, Y.C. Manipulating camera gimbal positioning by deep deterministic policy gradient reinforcement learning for drone object detection. *Drones* **2024**, *8*, 174.
31. Wickers, A.; Alpen, M.; Horn, J.; Thomas, D.; Gündel, M. Spatial resolution evaluation for UAS-based inspection missions. In *Life-Cycle Performance of Structures and Infrastructure Systems in Diverse Environments*; CRC Press, 2025; pp. 505–513.