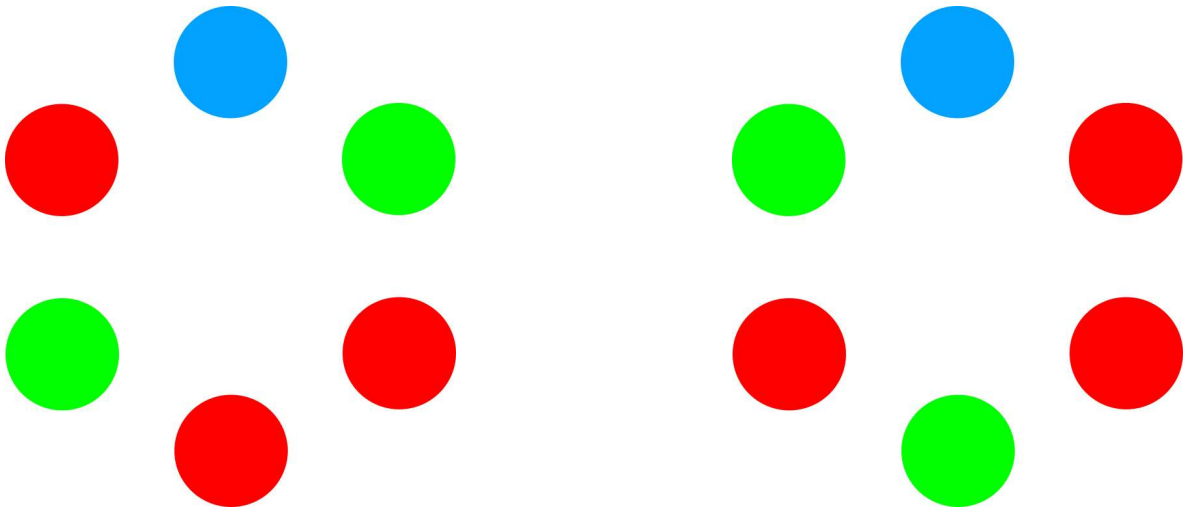

**(a)** GRRGR Target **(b)** RRGRG Target

**Figure 1:** Example of hexagonal targets and their string identifiers

# 1 Extrinsic parameters calibration

The problem of automatic calibration of the relative pose among multiple cameras is of utter importance to an holographic communication system. While the intrinsic parameters of the cameras can generally be calibrated once and stay relatively stable over time, the extrinsic parameters of the camera would need to be relaibrated each time the system is installed, as as one a regular basis to account for micro-motion of the support.

Thus, having an automatic calibration procedure that can be applied without human intervention is of high importance. In this section, we present the system we developed for our prototype.

## 1.1 The target

At the core of our calibration system, we use a series of hexagonal targets with six large color dots positioned on each of the hexagon corners. One of the dot is blue and represent the top of the hexagon, which allow to determine the orientation of a target once detected in an image. The five other dots are set to be either green or red. This encode a 5-bit binary string for each specific target; see Figure 1. As such, 32 unique targets can be created.

## 1.2 Rough target detection

Detecting the targets described in Section 1.1 can be achieved with a simple computer vision algorithm. The first step is to segment out pixels which are likely to be part of such a target. Overall, these pixels should be of a pure red, green or blue color, but in practice the ambient light and the image acquisition system will affect the display color. In the end, the formula we used to get a rough segmentation mask $\mathring{\boldsymbol{M}}$ of the set of candidate pixels is:

$$\mathring{\boldsymbol{M}}_{[\boldsymbol{p}]} = \min_c \boldsymbol{I}_{[\boldsymbol{p},c]} < t_{minColor} \vee \max_c \boldsymbol{I}_{[\boldsymbol{p},c]} - \min_c \boldsymbol{I}_{[\boldsymbol{p},c]} > t_{diffColor}, \tag{1}$$

where $\boldsymbol{p}$ is the pixel coordinate, $c$ id the channel and $t_{minColor}$ as well as $t_{diffColor}$ are user defined

threshold. The pixel is selected if any of the condition is true. While it would be better, according to the definition of a candidate, to select pixels for which both conditions are met, we found out in practice that to avoid the risk of false negative, it is better to select more candidates early on, and then filter them out.

The next step is to perform Connected Components Analysis (CCA) on $\mathring{\boldsymbol{M}}$ (He et al., 2017). Both the queen and rook contiguity can be used for the CCA. The clusters can then be filtered out based on their size and some morphological metrics of the shape of the object (Ao et al., 2020). We added a min threshold $t_{minArea}$ and max threshold $t_{maxArea}$ on the area of each cluster P:

$$t_{minArea} \leq \sum_{\boldsymbol{p} \in \mathrm{P}} 1 \leq t_{maxArea}, \tag{2}$$

Such simple threshold will already remove the very large and small clusters selected by error in Equation (1). We also added a threshold $t_{axisRatio}$ on the ratio between the minor and major axis of the pixel cluster:

$$\frac{\sigma_{min}}{\sigma_{max}} > t_{axisRatio} \tag{3}$$

where $\sigma_{min}$ and $\sigma_{max}$ are the square root of the eigenvalues of the covariance matrix of the pixels coordinates in P. This removes clusters that are obviously not round.

The next step is to detect among the remaining clusters the ones that correspond to a target. To this end, each clusters are considered in turn. Then, the five nearest clusters are found. The centroids of each six clusters are considered. An ellipse is fitted upon the six points:

$$\boldsymbol{A} \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix} = \mathbf{1} \tag{4}$$

where $\boldsymbol{A} \in \mathbb{R}^{|6 \times 3|}$, with $\boldsymbol{A}_{[i,.]} = \begin{bmatrix} x^2 & xy & y^2 \end{bmatrix}, [x, y] = \bar{\mathrm{P}}_i - 1/6 \sum_{j=1}^{6} \bar{\mathrm{P}}_j$ is the derivative of the ellipse equation:

$$\alpha x^2 + \beta xy + \gamma y^2 = 1, \tag{5}$$

fitted to the zero mean centroids of the clusters under consideration $\bar{\mathrm{P}}_i - 1/6 \sum_{j=1}^{6} \bar{\mathrm{P}}_j$.

Equation (4) is over-determined, with six equations for only three unknown, and can be solved using a least square solution:

$$\begin{bmatrix} \hat{\alpha} \\ \hat{\beta} \\ \hat{\gamma} \end{bmatrix} = (\boldsymbol{A}^\top \boldsymbol{A})^{-1} \boldsymbol{A}^\top \mathbf{1}. \tag{6}$$

The residual error can then be computed as:

$$\boldsymbol{r} = \boldsymbol{A} \begin{bmatrix} \hat{\alpha} \\ \hat{\beta} \\ \hat{\gamma} \end{bmatrix} - \mathbf{1} \tag{7}$$

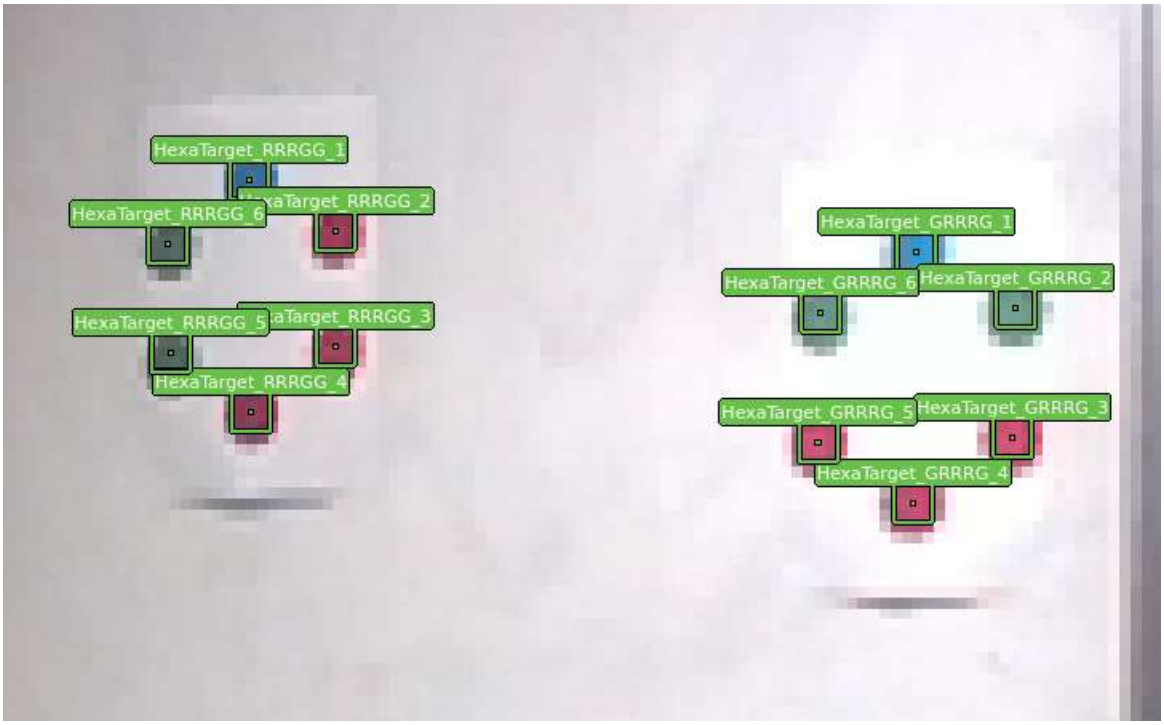


**Figure 2:** Output of our hexagonal target detector. The targets are detected, and uniquely identified by their color arrangement.

If the largest residual error value after optimization is less than a threshold $t_{ellipse}$, then the clusters are considered to be a target, removed from the list of clusters to treat and sent to the next step:

$$\max_i \left| \boldsymbol{r}_{[i]} \right| < t_{ellipse}. \tag{8}$$

else the clusters are left in the pool of selectable clusters

### 1.3 Detected target analysis

First, the color of the points shall be detected. The mean color of the pixels in each clusters is computed as an RGB triplet. Each component is multiplied by a gain, used to compensate for the color temperature of the camera. Then, the color of the largest component in the triplet is assigned to the cluster. As a single blue cluster shall exist in the target, if more than one blue cluster is detected, the target is ignored instead.

The points in the target are then sorted in clockwise order, so that the binary string corresponding to the target can be extracted. Finally, the target is labelled so that the subpixel position of each cluster can be refined by the next step of the processing pipeline.

### 1.4 Subpixel target alignment

The position of the dot is of utter importance in the quality of the final calibration. Indeed, small perturbation in the sub-pixel position of a given reference point can have a detrimental impact on the estimation of the relative pose of the camera and the target (Jospin et al., 2019). The effect is the most detrimental when an error is made in the relative orientation of the camera and the target, as the

lever arm effect can lead to very large errors when multiple cameras are oriented using the same target. While this can be somewhat mitigated by using multiple targets at once, it is still of utter importance to get a good estimate of the sub-pixel position of the center of each point in the target. We have considered using the method proposed by Jospin et al., 2019, based on an image formation model, but this has proved to be challenging, as the siye of the cluster is way larger than a single pixel. Instead, a naive estimator could be the centroid of the cluster:

$$\frac{1}{|\mathrm{P}|}\sum_{\boldsymbol{p}\in\mathrm{P}}\boldsymbol{p}, \tag{9}$$

which proved to be suboptimal in practice, especially for smaller clusters. To improve the estimate of Equation (9), we used a weighted centroid estimate of a dilated cluster of pixels, with a structural element E:

$$\frac{\sum_{\boldsymbol{p}\in\mathrm{E}\oplus\mathrm{P}} w(\boldsymbol{p})\cdot\boldsymbol{p}}{\sum_{\boldsymbol{p}\in\mathrm{E}\oplus\mathrm{P}} w(\boldsymbol{p})}, \tag{10}$$

with the weights $w(\boldsymbol{p})$ computed by comparing the color of each pixels to the mean color of the cluster:

$$w(\boldsymbol{p}) = 1 - \frac{\left\| \boldsymbol{I}_{\boldsymbol{p},\cdot} - \frac{1}{|\mathrm{P}|}\sum_{\boldsymbol{q}\in\mathrm{P}} \boldsymbol{I}_{\boldsymbol{q},\cdot} \right\|_2}{\max_{\boldsymbol{r}\in\mathrm{E}\oplus\mathrm{P}} \left\| \boldsymbol{I}_{\boldsymbol{r},\cdot} - \frac{1}{|\mathrm{P}|}\sum_{\boldsymbol{q}\in\mathrm{P}} \boldsymbol{I}_{\boldsymbol{q},\cdot} \right\|_2}. \tag{11}$$

This approach proved to be a more robust estimate in practice, as we saw less targets being incorrectly oriented during our calibration experiments.

### 1.5 Perspective-n-Points and Bundle adjustment

Once the position of each target points has been determined in the images, the relative pose of each target and each images can be obtained using a Perspective-n-Points (PNP) solver (Li et al., 2012). The function return the pose of a camera in a certain frame of reference given a set of 3D points in that frame of reference, and their corresponding 2D projection on the image frame.

A rough estimate of the holographic communication rig cameras positions and orientations, as well as of the absolute position of each target is obtained by building a solution step by step. First, a reference camera is selected, which local reference frame becomes the local reference frame for the holographic communication rig. Then, all targets detected in that camera are aligned into the holographic communication rig by using PNP based on the local coordinates of each target, and inverting the solution to get the pose of the target in the camera frame. Then, images with projection of the currently solved target can be aligned on the rig frame of reference using PNP. The procedure is repeated until all cameras and targets are aligned.

The final solution is then obtained by using a global bundle adjustment solver derived from the one that was developed for the work on topographic mapping by Cledat et al., 2020. The final solution gives the calibration of the holographic acquisition rig; see Figure 3.

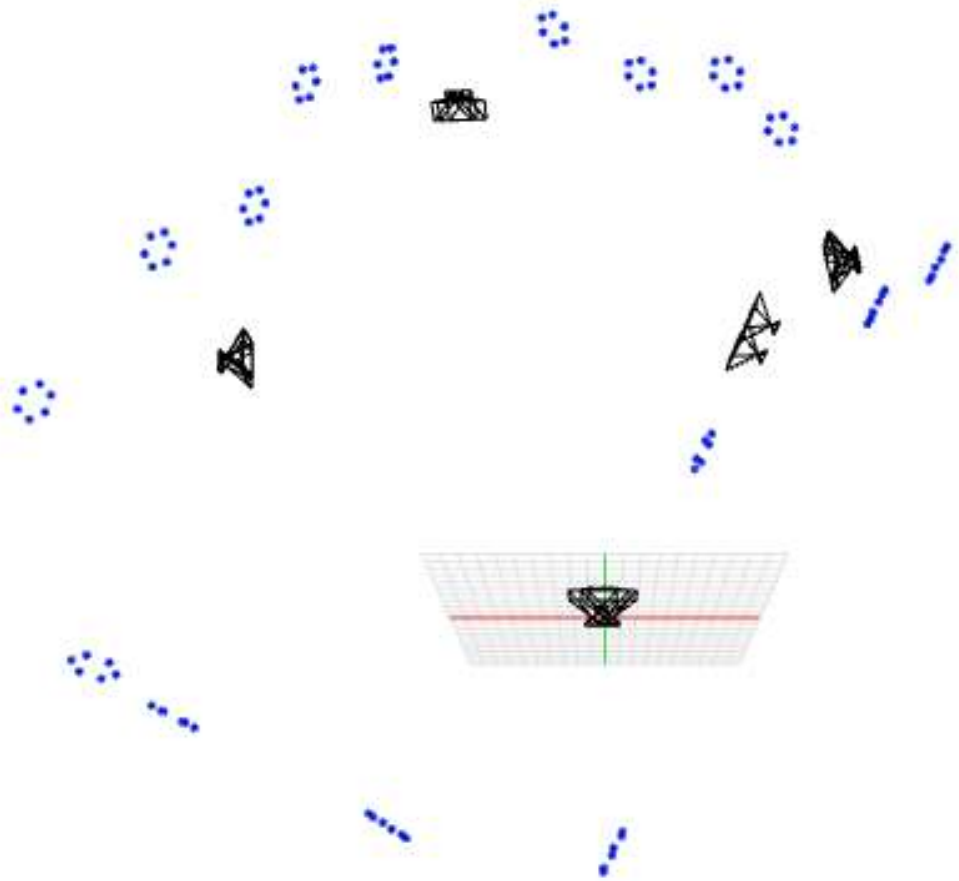

**Figure 3:** Reconstructed cameras and targets positions of the holographic communication rig.

## Bibliography


Ao, W., Fu, Y., Hou, X., & Xu, F. (2020). Needles in a haystack: Tracking city-scale moving vehicles from continuously moving satellite. *IEEE Transactions on Image Processing*, *29*, 1944–1957.

Cledat, E., Jospin, L., Cucci, D., & Skaloud, J. (2020). Mapping quality prediction for rtk/ppk-equipped micro-drones operating in complex natural environment. *ISPRS Journal of Photogrammetry and Remote Sensing*, *167*, 24–38.

He, L., Ren, X., Gao, Q., Zhao, X., Yao, B., & Chao, Y. (2017). The connected-component labeling problem: A review of state-of-the-art algorithms. *Pattern Recognition*, *70*, 25–43.

Jospin, L., Stoven-Dubois, A., & Cucci, D. A. (2019). Photometric long-range positioning of led targets for cooperative navigation in uavs. *Drones*, *3*(3).

Li, S., Xu, C., & Xie, M. (2012). A robust o(n) solution to the perspective-n-point problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *34*(7), 1444–1450.