EXPERIMENTAL PROTOTYPE FOR REMOTE TOWER SYSTEMS DESIGN

Jaime López-Araquistain, Iván Campaña, Luca Bergesio, Juan A. Besada

Abstract

Small and medium-sized airports are poorly occupied or occupied for a small period of time, hence the remote control towers are a good solution to control air traffic in these airports. A surveillance and monitoring air traffic prototype has been built to test surveillance, control and visualization concepts. The prototype is comprised of a hybrid real-synthetic scenario using augmented reality techniques to evaluate the operation of the surveillance/vision systems, observing the behavior of this system with different targets, backgrounds, inclement weather, etc. and helping the design/experimentation of novel control procedures in the airport area. This allows the analysis of risky situations and the controller training without putting at risk neither people nor material goods.

Keywords – remote control tower, trajectory prediction, drones, surveillance system, augmented reality, decision support tools.

Introduction

Concern about efficiency, comfortability and security in air transports, and especially in airports, has boosted in last decades caused by the constant increase of passenger traffic and the number of operations. Accordingly, priority in control services and air traffic management (ATC-ATM, Air Traffic Control – Air Traffic Management) is to facilitate efficient and safe transits of aircraft, with improvements in navigation and in operational procedures on ground. These processes are carried out in control towers, which have become in essential elements in the functionality of the current airports.

But control towers are expensive, so small and some medium airports face an economic dilemma because the cost for the tower operation is very significant for the few flights they manage in a given day, typically less than 10 flights per day. As a consequence, these types of airports are less profitable and less cost efficient whereas the air control equipment is worse and less complex than the one at bigger and more important airports. To overcome that problem, remote towers, allowing to operate these airports by air traffic controllers from a bigger airport or controlling several small airports form a central facility [1][2][3], are appearing. This solution consists of transmitting images in real time from cameras placed in the airport (and other surveillance and control information) to a remote control tower located hundreds of kilometers away. This is the case of Ornskoldsvik Airport in Sweden, which was the world's first remote control tower with no humans in the airport [4].

Surveillance and Monitoring Air Traffic Prototype

This paper describes a model built in Universidad Politécnica de Madrid for the research in remote control tower technologies and operations for advanced monitoring in airports. The idea is to use this model to test many concepts of surveillance and monitoring of air traffic, and see how could be extended to airports of low or medium size. The model is composed of a hybrid system encompassing a physical model of the aircraft fleet implemented using drones, and a simulation of other operations, synchronized with real time operations, using high fidelity trajectory simulation and augmented reality techniques. This hybrid simulation-physical prototype has been created in order to help designing and validating the key elements of the system: sensor systems, surveillance data-fusion chain, traffic management systems (separation, conflict detection and resolution, etc.).

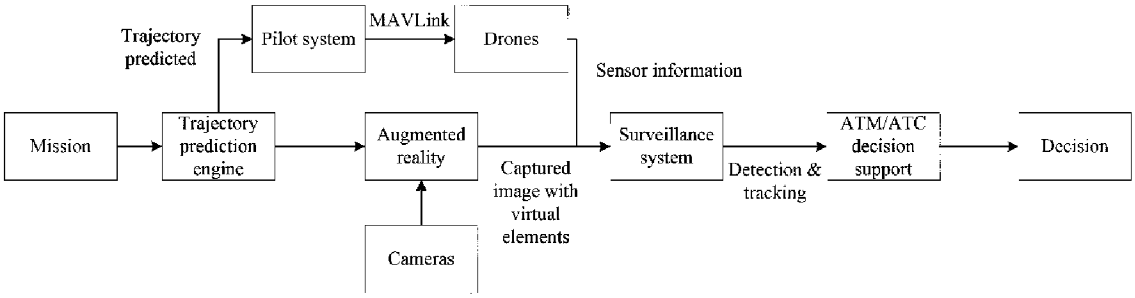


Figure 1. Surveillance and Monitoring Air Traffic Prototype

The structure of the surveillance and monitoring air traffic prototype can be observed in Figure 1. All the process begins with the knowledge of the missions to be performed (similar to detailed flight plans), which, once they are known and delimited, they are passed to trajectory prediction engine. This engine obtains the drone trajectory that has to be followed to fulfill the estimated mission. This prediction is used to recreate a drone virtual movement in an augmented reality scenario and it is passed to the pilot system for actual physical drones. In this last case, the trajectory is used to create instructions that drones have to do to carry out the mission, so a trajectory translation to MAVLink instructions [5] is made and it is included in a MAVLink file which is sent to drones.

On the other hand, the prediction of virtual drone trajectory is used to move the drone in an augmented reality scenario which is placed over the actual captured image taken by a set of cameras from the physical area where the physical drones may perform their maneuvers. This real-time image captured from the real scene is augmented with the virtual elements (virtual scenario elements and virtual drones) and it is preceded to the surveillance system, as well as sensor information obtained from drone. In this system, the detection and tracking of mobile objects is performed, which is passed to the ATM/ATC decision support tools to assess different separation, perform conflict detection and resolution, etc.

In the next sections the different systems will be explained in detail.

Trajectory Prediction Engine

As it has been previously introduced, a high fidelity trajectory prediction engine is in charge of transforming the mission into a predicted set of positions. The missions are defined as high level objectives containing the general information: main waypoints and points of interest (POI). Waypoints are used to control the trajectory positions and POI to control the drone yaw. Therefore, the trajectory engine models the predictions generating from the mission different segments with predefined movements as straight lines, circles and helixes.

This engine models the quadrotor as a rigid body, where the dynamic equations of its movement can be written as [5]:

$$\frac{d\dot{x}_e}{dt} = \frac{(\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi)U_1 - k_s(\dot{x}_e - w_x)}{m}$$

$$\frac{d\dot{y}_e}{dt} = \frac{(\cos\phi\sin\theta\ \sin\psi - \sin\phi\cos\psi\ )U_1 - k_s(\dot{y}_e - w_y)}{m}$$

$$\frac{d\dot{z}_e}{dt} = \frac{(cos\phi\ cos\ \theta)U_1 - k_u(\dot{z}_e - w_z)}{m} - g$$

where $(\dot{x}_e, \dot{y}_e, \dot{z}_e)$ is the quadrotor's speed with respect to the same earth-fixed frame, (ϕ, θ, ψ) are the rotation angles (Euler angles) around the Cartesian axes, (w_x, w_y, w_z) the Cartesian local wind speeds respect to the same earth-fixed frame, U_1 the average square thrust, g the acceleration due to gravity, m is its mass, and (k_s, k_u) are the horizontal and vertical friction coefficients.

With these equations, and constraining the movement with four additional equations we can obtaining a movement prediction. In our case we constrain the problem defining the flights by the horizontal speed in polar coordinates, vertical speed, and yaw. A full study of the effect of the constraints is made in [6].

Therefore, defining the horizontal and vertical speed and the yaw, the trajectory can be sampled obtaining all the relevant parameters of the drone movement.

Pilot and Drones System

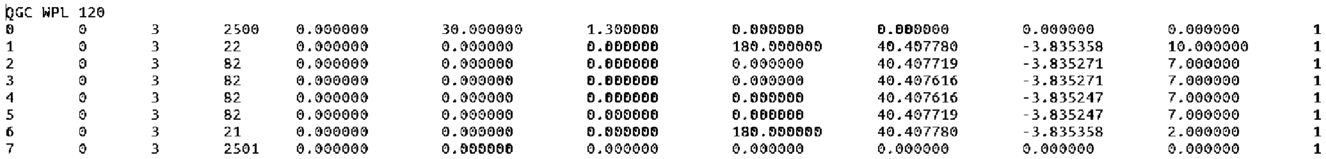
Trajectory prediction engine provides as output the predicted trajectory which is used in the pilot system. From the predicted trajectory, the necessary instructions to be passed to the drone are calculated by the pilot system.

These instructions are translated into a flight plan, which is ultimately a set of waypoints and legs, i.e. an initial position, a movement and a final position. For a flight plan it must be know the 3D position of waypoints and its yaw, as well as the 3D position of next waypoint and its yaw, because the drone movement between two waypoints is stipulated by the instruction. The drones used are the parrot Bebop drone 2 of Parrot [7], which are commercial low cost drones controllable through MAVLink [5] or Micro Air Vehicle Link, which is a protocol for communicating with small unmanned vehicle. It is based on a very lightweight, header-only message marshalling library [8].

Thus, a flight plan would be defined by a set of consecutive waypoints and instructions, which are:

- Take off
- Land
- Go to a waypoint
- Circle trajectory
- Delay
- Start/finish video capture

An example of a MAVLink file is shown in Figure 2. In a MAVLink file the first column represents the sequence of the mission item, whereas the second and the third columns are internal parameters of MAVLink. The fourth column is the instruction, and in the example the values that this column takes are, in the first row is to start a video capture, the second row is to take off, the next four rows are to go to a waypoint, the next row is to land and the last one is to finish a video capture. And the next columns are the different parameters that are needed for each instruction. In the movement instructions (take off, land, go to a waypoint and circle trajectory) it is necessary a 3D position of a waypoint and its yaw, whereas in action instructions (delay, video and photo capture) these parameters are not needed, the parameters used are the time of a delay in the waypoint, frame per seconds of the video capture or the resolution in megapixels, among others.



```
QGC WPL 120
0     0     3     2500   0.000000     30.000000     1.300000     0.000000       0.000000     0.000000      0.000000      1
1     0     3     22     0.000000     0.000000      0.000000     180.000000     40.407780    -3.835358     10.000000     1
2     0     3     82     0.000000     0.000000      0.000000     0.000000       40.407719    -3.835271     7.000000      1
3     0     3     82     0.000000     0.000000      0.000000     0.000000       40.407616    -3.835271     7.000000      1
4     0     3     82     0.000000     0.000000      0.000000     0.000000       40.407616    -3.835247     7.000000      1
5     0     3     82     0.000000     0.000000      0.000000     0.000000       40.407719    -3.835247     7.000000      1
6     0     3     21     0.000000     0.000000      0.000000     180.000000     40.407780    -3.835358     2.000000      1
7     0     3     2501   0.000000     0.000000      0.000000     0.000000       0.000000     0.000000      0.000000      1
```

Figure 2. MAVLink Example

Everything explained so far in this section is about how MAVLink instructions are calculated, but once this translation is done, it is necessary to communicate the instructions to the drone. It is done through a script in C with the parrot libraries to establish communication with the drone, using a laptop connected to the Wi-Fi network created by the drone. Once the communication is established, the first thing that must be done is the MAVLink file and then send it from laptop to drone, so the drone begins the trajectory estimated. But in the script also there is a set of callbacks, which ask for different information to the drone and this sends it back to the laptop. The main information requested is GPS coordinates of the drone, because this information is employed in the surveillance system. The rest of the requested information may be displayed on the screen for a controller to know the status of the drone, as this includes the status of the battery and the sensors, the Wi-Fi signal, the attitude…

Augmented Reality

The emulation of virtual aircraft/drones using augmented reality models can be made from the predicted trajectory. In this case it can emulate synthetically video flows that define different scenarios with simulated targets, where the background can be synthetic, real or a mix of both. All of the part of augmented reality it has been made with Unity [9] which is a cross-platform game engine developed by Unity Technologies and used to develop video games for PC, consoles, mobile devices and websites.

The synthetic background that it has made consists of a virtual airport placed in a field, simulating the real scenario where the tests have been done. The real scenario is located in Montegancedo Campus, of Universidad Politécnica de Madrid, Madrid and it looks as can be seen in Figure 3. The virtual background is a field with an airport with two tracks and a hangar, and it can be seen in Figure 4. All the elements that composed the scenario are open source models, as well as the virtual drone that is used.

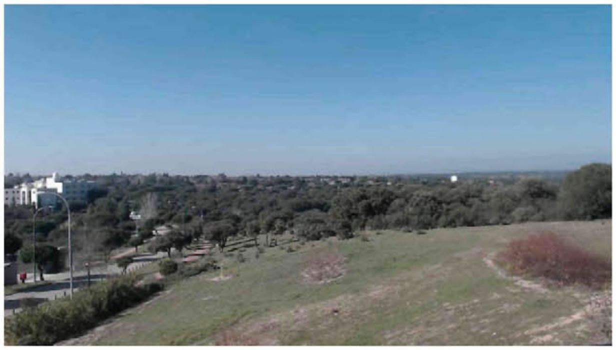


Figure 3. Real Scenario

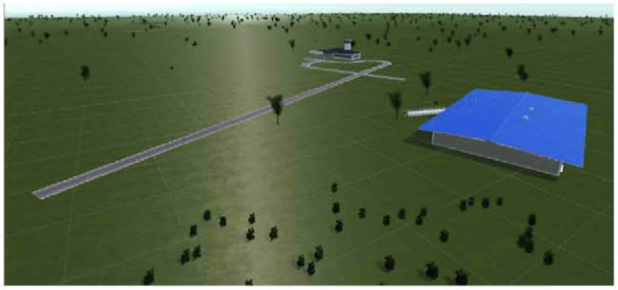


Figure 4. Synthetic Scenario

Taking the two previous scenarios and mixing some elements of the virtual and the real background the hybrid real simulated scenario background is obtaining as it can be seen Figure 5.

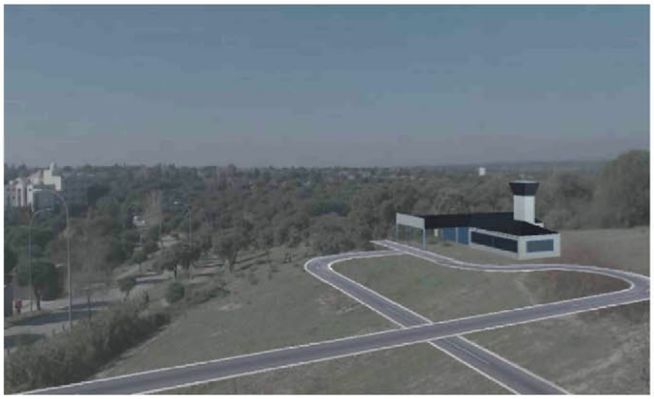


Figure 5. Hybrid Real Simulated Scenario

The simulated aircraft trajectories follow realistic paths integrated using our high-fidelity trajectory prediction engine, and we built tools to integrate the real-time surveillance/vision system described in next section with this emulation, so that hybrid real/simulated scenarios may be defined. From the 3D position and attitude of the virtual drone, obtained from the predicted trajectory, the 3D drone model and a Unity camera model, the view of the drone from the real camera position is emulated and it is overlaid to the background model, as shown in Figure 6.

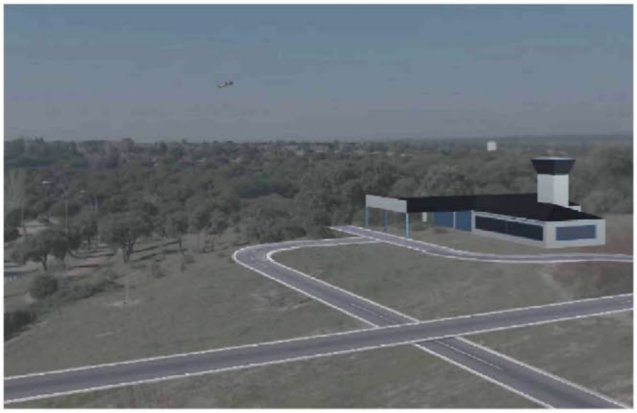


Figure 6. Hybrid Real Simulated Scenario with Virtual Drone

Also, from this same position a view taken from the drone position may be obtained. An example of what the virtual drone observes is shown in Figure 7.

Figure 7. Drone Observation

Using this process, it can be checked the performance of the whole system in different scenarios; changing the targets, the background, the weather, etc., enabling the analysis of risky situations, and also aiding to controller training without putting at risk neither people nor material goods.

Surveillance System

To be able to emulate remote tower visualization means and to perform the detection and tracking of unmanned aerial vehicles we developed a surveillance/vision system. The surveillance/vision system consists of several cameras and other sensors working synchronously and in real time. As it has been said before, cameras take images in real time and once the virtual elements are incorporated to the captured image, this image reaches the surveillance/vision system and it is processed by a color detection and tracking algorithm and stereoscopic vision so detection and position of drones in 3D are obtained. Apart from the artificial vision tracking, the GPS coordinates received from the drone through MAVLink are also integrated in the data fusion process.

The tracking system has a distributed tracking architecture, in which each of the cameras that composed the system has a dedicated positioning and tracking system that calculates the kinematic state of the targets in the projected plane of the camera. This type of architecture allows the system to be simple and scalable, so it is easy to incorporate or eliminate some cameras in the detection and tracking process. To obtain the tracks the system has a local tracker for each of the cameras, whose structure can be seen in the Figure 8.

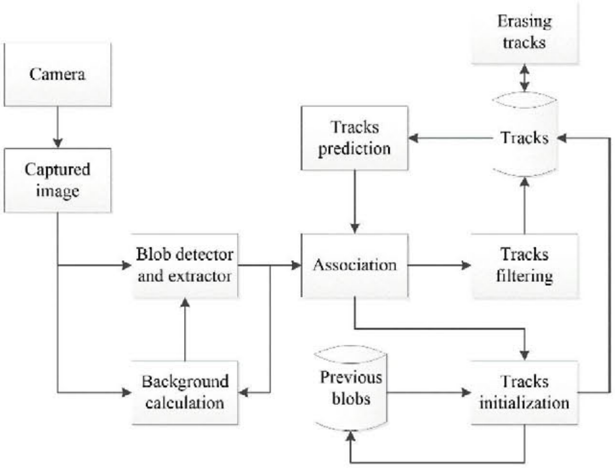


Figure 8. Local Tracker Structure

Additional details of the camera tracking system can be found in [10], [11], [12] and [13]. We extended these algorithms to make use of color cameras in HSV (Hue, Saturation and Value) color space. Again, the system is based on background estimation, blob extraction, centroid estimation and association to preexisting tracks, and Kalman filtering of centroids. The basic purpose of this process is to aid in target detection and centroid association making is more robust.

Once the detection and tracking system is explained, now is the turn of the stereoscopic vision. For the realization of the stereoscopic vision the tracks associated with the detections that exist for each camera in that frame are needed.

The first thing to do is to pass the filtered positions of the centroids of each of the tracks associated by each of the cameras, since points that occupy these centroids are those that are used to realize the triangulation and the obtaining of the point 3D in the real world. Next process is the association of tracks of each camera to have a global track, so it is necessary to calculate the point in real world coordinates from two centroids of associated tracks [14].

This calculation is made through a triangulation process as can be seen in the Figure 9.

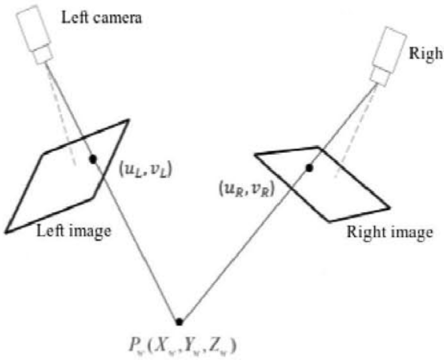


Figure 9. Geometry of the Positioning System of Objects through Stereoscopic Vision

P_w is the position of a reference point obtained through triangulation. To convert this geometry into algebraic expressions, we define M_L and M_R, which denote the projection matrices of the calibrated world to the image plane of each camera, and λ_L and λ_R as the respective multipliers. In addition, the 2D projections of a reference point P_w in the left and right image planes respectively are defined. From those definitions we have the following model of the projection of a given point to camera frame (P_f), where M may be M_L or M_R for each camera.

$$\lambda \begin{pmatrix} P_f \\ 1 \end{pmatrix} = M \begin{pmatrix} P_w \\ 1 \end{pmatrix}$$

From this relation, the following system of linear equations is obtained:

$$AP_w = b$$

where:

$$A = \begin{pmatrix} M_L(1,1) - u_L M_L(3,1) & M_L(1,2) - u_L M_L(3,2) & M_L(1,3) - u_L M_L(3,3) \\ M_L(2,1) - v_L M_L(3,1) & M_L(2,2) - v_L M_L(3,2) & M_L(2,3) - v_L M_L(3,3) \\ M_R(1,1) - u_R M_R(3,1) & M_R(1,2) - u_R M_R(3,2) & M_R(1,3) - u_R M_R(3,3) \\ M_R(2,1) - v_R M_R(3,1) & M_R(2,2) - v_R M_R(3,2) & M_R(2,3) - v_R M_R(3,3) \end{pmatrix}$$

$$b = \begin{pmatrix} u_L M_L(3,4) - M_L(1,4) \\ v_L M_L(3,4) - M_L(2,4) \\ u_R M_R(3,4) - M_R(1,4) \\ v_R M_R(3,4) - M_R(2,4) \end{pmatrix}$$

To estimate the position of the reference point, the equations can be solved using the least-squares approximation, and the result is:

$$P_w = A^T \cdot A \cdot (A^T)^{-1} \cdot b$$

This gives the x, y and z coordinates of the 3D point in the real world. This procedure may be extended trivially to more cameras.

Once the point is obtained in real world coordinates, it proceeds to bring the point to the plane of the image to obtain the position it occupies in the image after the triangulation and thus check if the two centroids used refer to the same track or, if, on the other hand, they are not centroids to be associated by referring to different detections. To do this the reverse process to the previous one is done.

In this case, the association is made by comparing distances. To do this, the distance between the original point and the point obtained after triangulation in the image plane is calculated. If this distance is small and is the minimum of all distances related to these centroids, then it can be said that those tracks refer to the same target by which they are associated.

These tracks are then entered into a Kalman filter to determine the correct positions of the system of global tracks. Once these positions are obtained, the result of the stereoscopic vision is fused with GPS-derived coordinates that are obtained from the drone for real drones. For virtual ones, the position is obtained from the predicted trajectory, with a GPS error model.

ATM/ATC Decision Support

Controllers and pilots are increasingly being planned the computer-based decision support tools, shared information and other forms of automation to support Air Traffic Management (ATM) and Air Traffic Control (ATC) in remote control towers. To have a successful adoption of these automation concepts requires the joint work of technology and human performance using this technology, so technology and human performance must act as co-partners and thus the whole system will benefit. Current version of our system does not include yet this kind of tools, although it is prepared to be enhanced with them. In fact, one of the main purposes of the tool is assist us to develop new concepts both on the design of such tools and on visualization of their results.

Initial Results

The results presented next are not definitive, as this is still an on-going process. Currently, the basic modeling and surveillance chain is completed, but we still do not have higher level decision support systems implemented.

In Figure 10 an output of the system with the image with the detections made by the surveillance/vision system overlaid over the detected drones is depicted. In this picture you may see two detected drones, a real one in the sky, and a virtual one taxiing above the virtual taxiway near the one of the crossing runways. This image references to the image captured by the left camera.

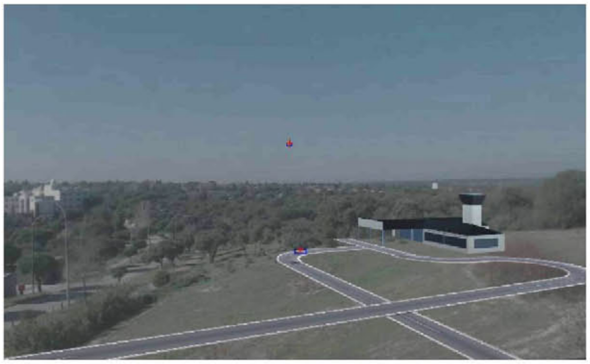


Figure 10. Detection Image Result

This figure is only an image taken from a video, but in the total playback of the video the drones are detected satisfactorily and at no time false alarms appear that would damage the decision maker.

Another potential output of the surveillance/vision system is the image with the result of fusing stereoscopic vision and GPS coordinates. Vision bases estimation is marked in next image (Figure 11) as a red cross while blue crosses refer to projected GPS coordinates, sent by the real drone and obtained for the virtual drone. As it can be observed, the GPS coordinates have more error, but they are enough close to see the trajectory that the drones are making. Additionally, GPS coordinates enables creating much more robust tracking, not suffering from typical problems vision based surveillance (occlusions, shadow effects, light changes effects, etc.). So the fusion system enables getting robust tracks due to the use of GPS-derived measures, and high accuracy in most situations due to the use of vision based surveillance.

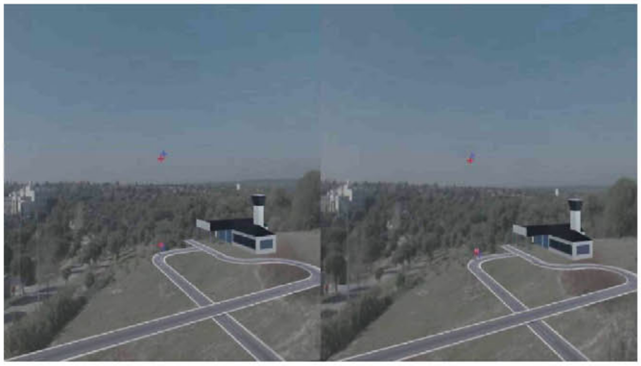


Figure 11. Stereoscopic Vision and GPS Coordinates Result

Finally, all the results are shown in a configurable video wall simulating a remote tower. The video wall is shown in Figure 12 and it is composed by 9 screens, with different potential configurations. The basic (starting one) represents:

- Screen 1: Real drone observation.
- Screen 2: Virtual drone observation.
- Screen 3: Stereoscopic vision and GPS coordinates image.
- Screen 4: Real drone flight plan.
- Screen 5: Virtual drone flight plan.
- Screen 6: Predicted trajectories.
- Screen 7: Real drone sensor information.
- Screen 8: Virtual drone sensor information.
- Screen 9: Predicted trajectories over real map.

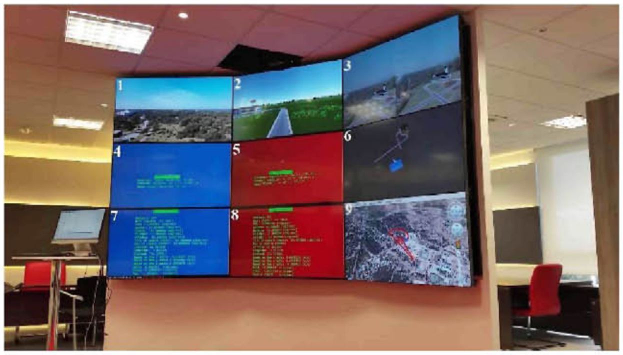


Figure 12. Video Wall Implementation

- The system allows us to change from one configuration to another using a natural pointing and voice interface. It also allows us to select a screen and put it in full screen. Another configuration is the presented in Figure 13. This is a single drone configuration showing the real drone information that the system has. Each screen displays:
- Screen 1: Real drone flight plan.
- Screen 2: Real drone sensor information.
- Screen 3: Skype to communicate with the pilot.
- Screen 4: Real drone observation.
- Screen 5: Predicted trajectories.
- Screen 6: Weather.

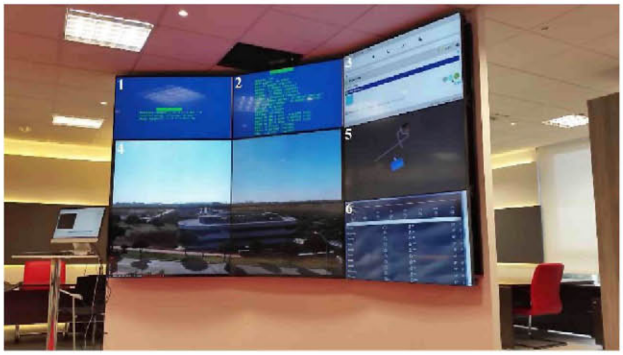


Figure 13. Drone Configuration

An additional configuration is a surveillance oriented one (Figure 14). In this configuration it is displayed:

- Screen 1: Stereoscopic vision and GPS coordinates image.
- Screen 2: Predicted trajectories.
- Screen 3: Weather.
- Screen 4: Skype to communicate with the ATCO (Air Traffic Control Officer).

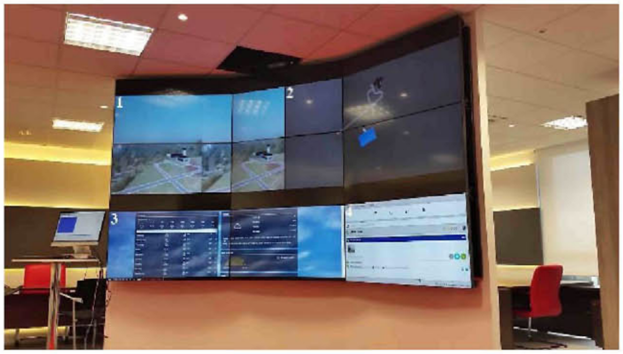


Figure 14. Surveillance Configuration

Conclusions

The paper describes a hybrid real/virtual model of a remote tower deployment. The prototype functionalities in detail, and also their interrelations, the implementation constraints and the complete HW/SW deployment to model a typical remote tower scenario are described.

The potential benefits of the prototype elaborated to perform are:

- It may be used to research on user interfaces for remote tower operations.
- It also may be used to design decision support tools and infer their results in a real scenario. It may therefore serve as a design/training tool to gain expertise in the creation of remote towers for small and medium airports, helping the developed solutions allow guaranteeing an efficient service (hopefully reducing reaction times, increasing safety, reducing the stress of the controller and its workload, etc.)
- It may be used as a showcase to demonstrate those new technologies.
- Finally, it may be used for educational purposes for ATM, drones and technology applications systems.

References

[1] Eier, D.; Huber, H., "Advanced Ground Surveillance for Remote Tower," Integrated Communications, Navigation and Surveillance (ICNS), 2008, vol., no., pp.1,9, 5-7 May 2008

[2] Papenfuss, A.; Friedrich, M., "Head Up Only – a Design Concept to Enable Multiple Remote Tower Operations," Digital Avionics Systems Conference (DASC), 2016 IEEE/AIAA 35th, vol., no., 25-29 Sept. 2016

[3] Eier, D; Gringinger, E; Klopf, M., "Semantic Information Management in a SWIM Enabled Remote Tower Environment," Integrated Communications, Navigation and Surveillance (ICNS), 2016, vol., no., 19-21 April 2016

[4] "Ornskoldsvik." (2015). [Online] Available: http://www.traveller.com.au/ornskoldsvik-airport-sweden-worlds-first-remotecontrol-air-traffic-control-tower-with-no-humans-inside-gjnlzl

[5] "Mavlink" [Online]. Available: http://qgroundcontrol.org/mavlink/start

[6] Frontera Sánchez, G. (2016). "Applications of Formal Languages to Management of Manned and Unmanned Aircraft (Doctoral dissertation, Telecomunicacion)".ISO 690

[7] "Bebop Drone" [Online]. Available: http://developer.parrot.com/docs/bebop/

[8] Crespo, G., Glez-de-Rivera, G., Garrido, J., & Ponticelli, R., "Setup of a Communication and Control Systems of a Quadrotor Type Unmanned Aerial Vehicle," Design of Circuits and Integrated Circuits (DCIS), 2014 Conference on

[9] "Unity" [Online] Available: https://unity3d.com/es

[10] Besada, J. A., Portillo, J., García, J., Molina, J. M., Varona, A., & González, G. (2001, August). Image-Based Automatic Surveillance for Airport Surface. In 4th International Conference on Information Fusion, Fusion (pp. 11-18). ISO 690.

[11] Besada, J. A., García, J., Portillo, J., Molina, J. M., Varona, A., & Gonzalez, G. (2005). Airport Surface Surveillance Based on Video Images. IEEE Transactions on Aerospace and Electronic Systems, 41(3), 1075-1082.ISO 690 .

[12] Kay, S. (1993). "Fundamental of Statistical Signal Processing: Estimation Theory." Englewood Cliffs, NJ: Prentice-Hall.

[13] Bar-Shalom, Y., & Li, X. (1993). "Estimation and Tracking: Principles, Techniques and Software." Norwood, MA: Artech House.

[14] Li, J., Bernardos, A. M., Tarrío, P., & Casar, J. R. (2015, February). A Combined Vision-Inertial Fusion Approach for 6-DOF Object Pose Estimation. In Seventh International Conference on Machine Vision (ICMV 2014) (pp. 944518-944518). International Society for Optics and Photonics.ISO 690

Acknowledgements

This work was supported in part by Universidad Politécnica de Madrid Project "Tecnologías Avanzadas para la Monitorización y Gestión Remota del Tráfico Aéreo de Vehículos Pilotados y no Pilotados" (RP1509550C02), by Fundación para el Desarrollo de las Telecomunicaciones (FUNDETEL) Project "Tecnologías y Sistemas de Control y Gestión de Tráfico Aéreo" (GPD.SJC.001), and by the Spanish Ministry of Economy and Competitiveness under Grants TEC2014-57022-C2-1-R and TEC2014-55146-R.

Email Addresses

jaime.lopez@grpss.ssr.upm.es

icampana@grpss.ssr.upm.es

luca.bergesio@grpss.ssr.upm.es

besada@grpss.ssr.upm.es