Using Immersive 3D Terrain Models For Fusion Of UAV Surveillance Imagery

Sean Owens*, Katia Sycara† and Paul Scerri‡
Carnegie Mellon University, Pittsburgh, PA, 15213, USA

Teams of small and micro UAVs currently require at least one operator per vehicle to watch video and plan paths. Each of the operators has a dull and difficult job to constantly monitor the video and coordinate with other operators to ensure the region of interest in covered. This paper presents initial steps towards an approach that would allow a single operator to utilize data from several UAVs and interact with the data in a more natural and less stressful way. The concept is to paint video directly onto a 3D model of the environment and allow the operator to interact with the model as they would a computer game. The location of any of the UAVs need not be known to the operator. The operator might eventually mark areas of the environment to be searched more or less carefully or often and allow the UAVs to cooperatively and autonomously determine paths that achieve this.

I. Introduction

Teams of small unmanned aerial vehicles (UAVs) show great promise as a reconnaissance tool for tactical military situations, disaster response, and search and rescue.[4,8] In this paper, we are particularly interested in tasks such as monitoring a security perimeter or area for any incursion. Very small UAVs are limited to a very light sensor payload, which must be effective at a significant distance, 100m or more, from objects being sensed. One of the few sensors light and low power enough to be used at these distances are EO (electro-optical, i.e. video) sensors. Due to the height at which the video is taken and the targets that are being sought, it is infeasible to use computer vision techniques to automatically process the data, instead human operators must interpret the imagery.

Unfortunately, monitoring video from even a single UAV is both difficult and dull.[13] While much work has looked at how to make this task easier, any technique that requires the operator monitor live video streams will tightly limit the number of UAVs a single operator can utilize.[3,14] Moreover, multiple operators controlling and monitoring individual UAVs is not a scalable solution for two reasons. First, the inter-operator coordination required for determining which UAV goes where does not scale well with the number of UAVs, hence the coordination and path-planning for the UAVs would need to be done autonomously. Second, more importantly, communication bandwidth is a highly constrained resource so streaming video from many UAVs simultaneously may not be feasible. Hence, any approach relying on operators monitoring live video streams cannot scale up to multiple UAVs and therefore cannot take full advantage of emerging UAV technology.

In this paper, we propose a fundamentally different approach, where video is taken and merged into an interactive 3D model that an operator can interact with, analogous to the way they would interact with many video games. The video images are painted into a mosaic over a 3D terrain model. While the mosaicking has been performed in other work,[9,11] typically the interface still follows the track of the UAV,[2,12] while in this work we completely break the connection between the UAV's location and what the operator is seeing in the interface. One of the consequences of this is that having multiple UAVs or multiple cameras on individual UAVs simply results in the model being updated more frequently and makes the operators task easier, instead of harder. A centralized mechanism can be used to plan UAV paths that will get the

*Research Programmer, Robotics Institute.
†Research Professor, Robotics Institute.
‡Systems Scientist, Robotics Institute.

data required to update the interface, potentially taking user preferences for priority and update rates into account. Critically, communication bandwidth can be effectively utilized by requesting that only UAVs with highest priority video use downlinks, while other UAVs move into position to get other useful imagery. For example, when a UAV is traversing a recently traversed area, it could relinquish communication channels for a UAV that is above a moving object.

The approach has a range of additional benefits. First, the 3D model of the environment can be disseminated more widely, since it is a more concise and useable representation of the environment than a video stream. For example, the model would be more useful to operators in the field, distracted by other tasks, than video streams would be. Second, the operator does not need to stop and start video (as has been done in previous work) to check details they might have missed. Instead, they interact with the latest data, zooming in and out, inspecting details at their leisure. Third, the 3D model naturally puts the imagery into its context, reducing some of the problems of gaining situational awareness,[1,5–7] i.e., rather than an isolated frame, they see the image next to images taken near by and in the context of the terrain. For example, rendering on the 3D terrain model might show that friendly and opposing forces are separated by a bluff that prevents them from directly seeing each other.

Figure 1. Video sample taken from UAV.

A very large range of technical challenges must be addressed to achieve this objective. These range from planning the UAV paths to managing the communication channels to the interface design. The primary aim of this work is to determine whether operators find the interface easier to use than live video, to provide confidence going forward. Hence the aim is to do enough to get a working prototype as quickly as possible. Specifically, we focus on some of the issues in getting and drawing the imagery on a 3D terrain model. Telemetry data matched precisely with images is required to exactly place an image frame on the terrain. Unfortunately, the data is never good enough and there is some unknown time delay between telemetry data and imagery reaching a common processing location.[10] In the future, we plan to use feature matching and sophisticated smoothing techniques to create mosaics, but in the current work, much simpler techniques are used.

The paper presents preliminary efforts towards achieving this goal. In particular, it presents the infrastructure that takes imagery from a UAV and renders it on a high fidelity 3D model that can be manipulated in a natural way. The user can manipulate the 3D model, viewing it from any angle and zooming in and out as required. The UAVs autonomously fly around the environment, trying to maximize coverage, but currently without taking into account the terrain. Planned usability tests will determine whether operators can more quickly and easily determine the location of some emergency ponchos placed in the environment.

II. Concept

Our approach attempts to reduce the load on the operators by taking them away from full motion video streams. We create a 3D terrain model that is painted (or 'textured') in real time with selected frames of imagery from multiple UAV full motion video streams. The terrain model is given an initial texture based on satellite imagery or older aerial imagery to provide the context of landmarks and geographical features (roads, buildings, rivers, etc.) As the live imagery streams in, individual frames are selected and applied to the model, replacing the original, older and possibly lower resolution initial texture. The overall concept is shown in Figure 2.

The operator can inspect the imagery in context and from varying angles and viewpoints, including viewpoints that no UAV occupied while the imagery was being captured. He can navigate around the model at his own pace, adapting his path to his own priorities, rather than being limited to the physical UAV paths. The operator may also interactively view the terrain model textured with imagery from different times, or specify areas of interest in order to update UAV planning priorities, resulting in additional passes by UAVs

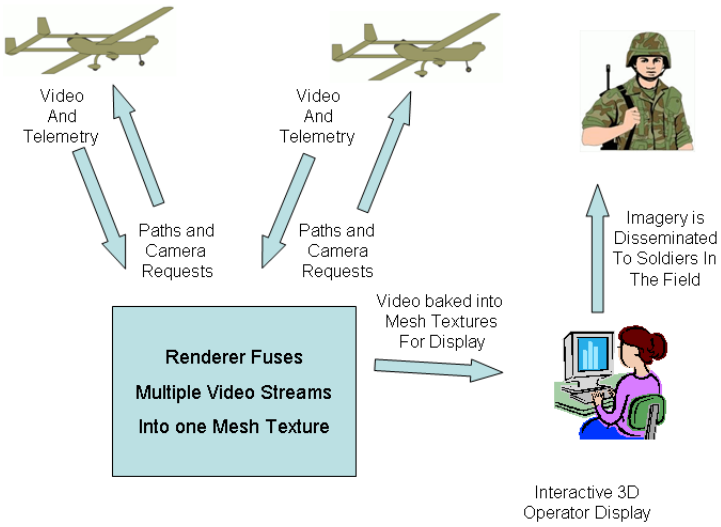

Figure 2. Overall system concept, showing data flows between UAVs, ground control and soldiers in the field.

at higher resolution (zoomed or lower altitude) or from different angles. This type of interaction is familiar to many people that have played strategic computer games where the player has a "God's Eye" view of the world. Figure 3 shows a screenshot from one such game, Panzer Command.

The 3D terrain model we construct is based on terrain elevation data from a high resolution (one meter resolution) LIDAR scan of the terrain. While such high quality terrain elevation data may not be available in all cases, especially for areas that are initially being explored, it is likely to be available for fixed, high-value assets (i.e. urban terrain, airfields, and bases).

Figure 3. "Panzer Command" game interface, where user has a "God's Eye" view of the simulated environment.

III. Implementation

It is generally a good idea when interpolating an image, (i.e. rotation, etc) to iterate over the pixels in the destination image and perform an inverse transform back to the source to sample colors. This is the approach taken in this implemention but the process is more complex and consists of multiple stages.

The goal is to sample colors from fresh video imagery and transfer them into the texture that is assigned to the 3D mesh representing out terrain, while taking into account view frustrum clipping, perspective, and occlusion, as well as clipping to the viewport, i.e. the video frame image itself.

The initial terrain texture and terrain mesh are loaded and built, and georeferencing is used to map triangles in the mesh into the initial terrain texture. Then a list of texel points contained by each triangle is built and assigned to the triangle for later use in baking new video data into the texture. Rather than iterate over every texel and try to map it to a triangle, the triangles are iterated over and each triangle is rasterized to the texture space. Each texel point in a triangle has it's texture coordinates stored as well as the corresponding 3D world space coordinates (on the triangle) that the texel gets texture mapped too.

For each video frame that is baked into the terrain texture, a set of visible texels is calculated, based on the viewpoint the frame was recorded from, i.e. the UAV's location and attitude. This is done by first projecting all terrain mesh triangles into the view frustrum of the UAV, culling any triangles that are completely outside the frustrum. The remaining triangles are projected onto the 2D view plane, and

intersection tests are performed and Constructive Area Geometry is used on intersections to reduce each triangle to only the part visibile from the viewpoint of the UAV, unobscured by other triangles.

After all triangles are checked for culling and occlusion, each triangle that is determined to be at least partly visible is used to sample colors from the video frame to be written into the texture. The triangle's set of texel points are iterated over. Each texel point is in turn projected to the viewplane and then tested against the visible triangle portion, and for being within the bounds of the viewport. For each texel that passes through this process of culling, occlusion and testing, and survives, color is sampled from the projected location in the video frame and written into the mesh texture. After all rasterized texels for all visible triangles have been examined, the baking process is complete.

Figure 4 shows the overall process that takes a video image and renders it on the 3D terrain. Figures 5 and 6 show additional detail on aspects of this overall process. Figure 5 shows in more detail how a triangle in the terrain mesh is mapped to a pixel in the terrain image. Figure 6 shows how visible triangles in the terrain mesh, from the perspective of the video image are determined.

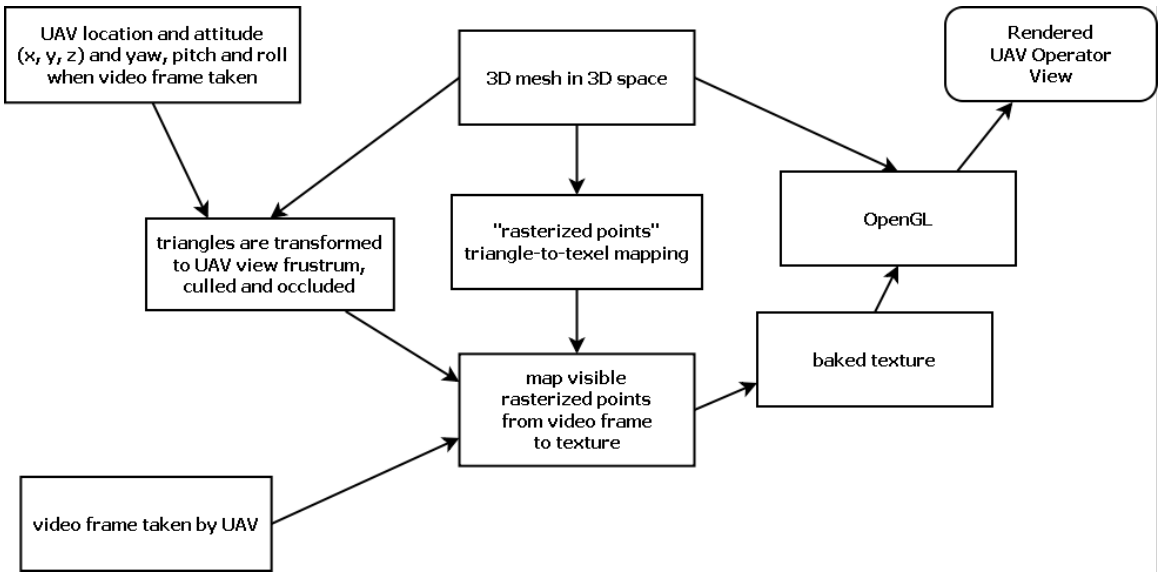

Figure 4. Overall rendering pipeline.

The initialization process works as follows:

1. Load the original, geo-referenced, aerial photo as a texture - this is the data structure that will be updated with video frames. Extract the pixels into an array of pixel color values.

2. Create a mesh representing the terrain from the Digital Elevation Map. The mesh is represented in an object space generated from the real world coordinates of the DEM.

3. The georeferencing of the DEM and the texture is used to calculate and store for each vertex of the mesh the u,v coordinates that specify (for that vertex) the 2D coordinates in the texture that are to be used by the pipeline to texture the mesh (an orthographic projection of the mesh triangles onto the mesh).

4. For each triangle, we pre-build and cache a set of *rasterized points*. Each rasterized point is specified in Object Space (3D) and in texture space (u,v 2D). The u,v coords specify the center of a texel in the texture and the corresponding point in object space on the triangle.

For each video frame to be painted on the terrain:

1. Get the corresponding video camera position and attitude (i.e. orientation or 'pose') and set/calculate the Viewing transform.

2. Set Viewing Frustrum perspective and projection to match the video camera (field of view, near and far clip planes, aspect ratio of pixels).

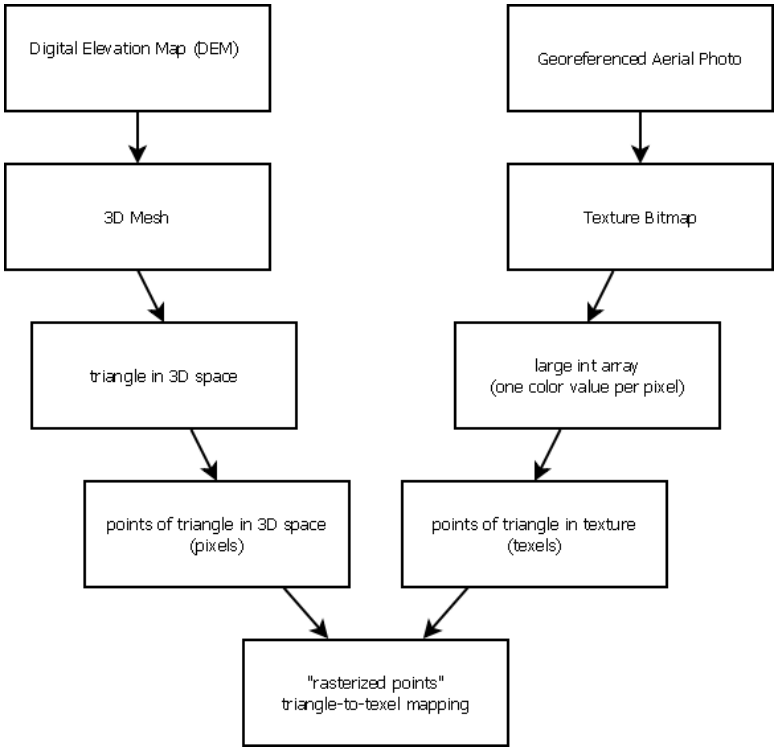

Figure 5. Triangle to texel mapping.

3. For every triangle in the model, that is not at least partially inside the view frustrum we mark it as unviewable and ignore in further processing.

4. For those triangles that are in our view frustrum, project them from world space all the way into 'eye space', i.e. onto the video frame. We store these 2D coordinates along with each triangle.

5. Sort the triangles by distance from the camera viewpoint in worldspace, closest to farthest.

6. Remove any parts of triangles that are obscured by other triangles in front of them.

7. Use the 2D video frame coordinates to sample the color from the video frame and the copy it into the corresponding texel in the aerial texture.

IV. Prototype

The software described above was integrated with a complete UAV control system and flown in a carefully controlled situation. A small Procerus UAV was used to provide the video stream, with the UAV autonomously covering an area of about 200m by 200m.

LIDAR data was used to create a high fidelity model of the area, however the data was nearly three years old so it did not correspond exactly with the current situation. The mesh created from the LIDAR data is shown in Figure 7. The area is not flat, with some small hills nearly 20m high.

High fidelity aerial imagery was also available, to provide the a priori texture that was updated with live data. The aerial data, rendered onto the terrain mesh is shown in Figure 8. Notice that this imagery was also about two years old, so incoming imagery did not match it exactly.

Figures 9 and 10 show a single video frame rendered on the mesh. Near the middle of 9 parts of the grid can be seen through the video image. This is because the screenshot is taken from a different angle than the UAV took the video image and those parts of the terrain were obscured. In Figure 10 the operator has moved their perspective closer to the ground, making it more apparent how the video image has been rendered on the terrain.

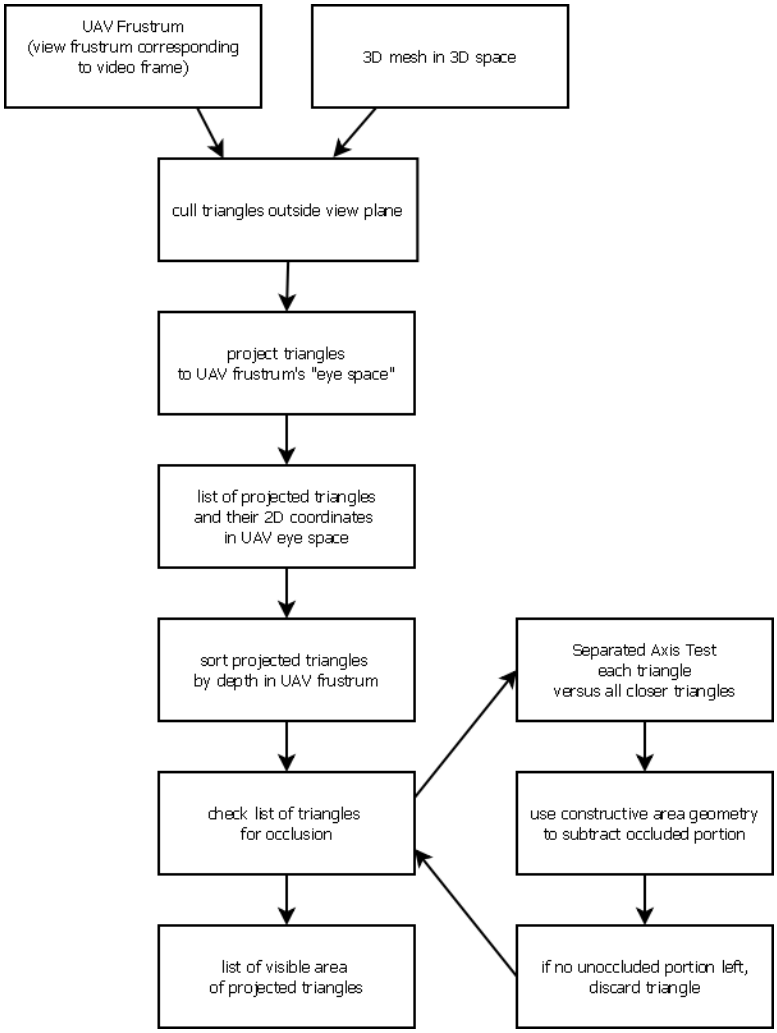

Figure 6. Culling visible triangles.

Finally, the video frame and a priori terrain are rendered together on the mesh, resulting in the interface seen in Figure 11. Notice that the video image is lit differently to the a priori image, but the a priori image provides important context for the video image. Over time, as more video images are added the a priori image will be completely obscured.

V. Future Directions

One of the immediate technical challenges is that of interfacing with UAV telemetry for location and pose data, to accurately determine the video sensor's field of view, in order to texture the image accurately on the terrain model. UAV telemetry data is typically imprecise or noisy, and may also lack timing synchronization with the video stream.[10] We will apply computer vision techniques to register the imagery against our initial aerial photography texture.

Another challenge is keeping pace with the relatively large amount of data produced by the multiple incoming video streams, in order to update the model in real time. In order to keep up with incoming video we preprocess video frames and evaluate the information gain they represent, to automatically select only high-value frames as updates to the model. This also has a bearing on the issue of conserving communication bandwidth. Currently we perform this process on the ground, but in the future, it may take place on the UAV itself, thus reducing bandwidth used for video streaming. We also plan to implement a second approach that estimates information gain based on the planned UAV path rather than it's current location and pose and compare bandwidth reduction and target location accuracy versus the first approach.

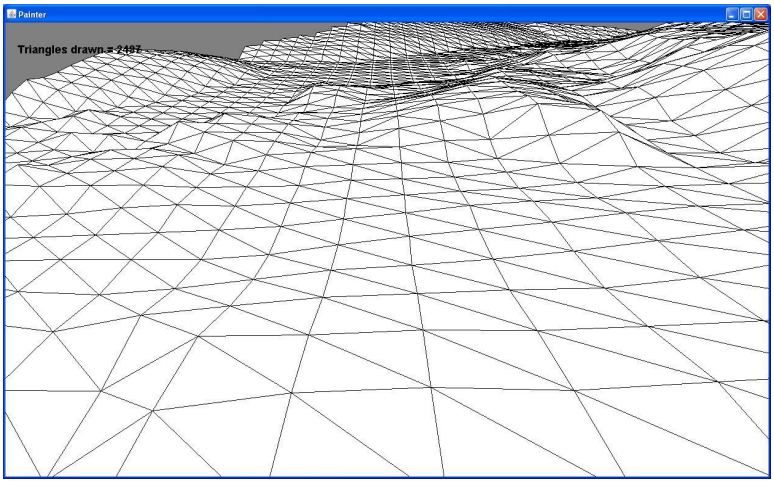

Figure 7. Mesh created from LIDAR data.

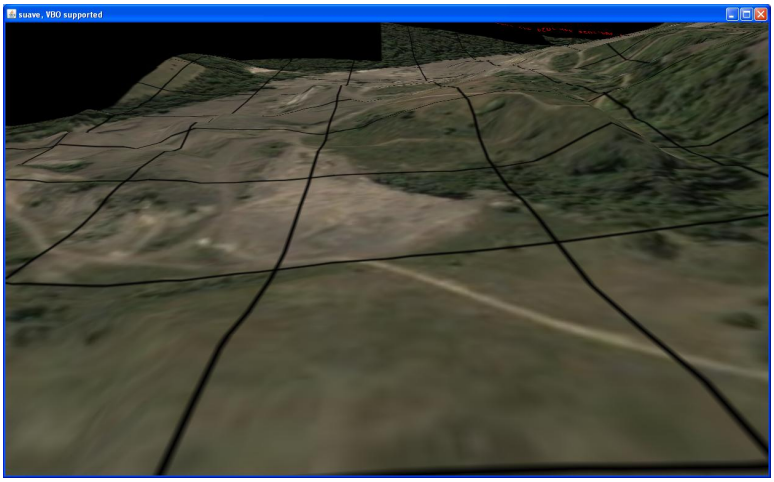

Figure 8. A priori aerial imagery rendered on the terrain mesh.

A third challenge is providing an effective means for the operator to interact with the model in real time. The operator can dynamically select older vs. newer imagery to be displayed on the model to examine changes over time. He may also alter the parameters that control whether to keep older higher resolution (from lower altitude passes or zoomed sensors) imagery preferentially over more recent lower resolution imagery. Situational awareness of the sensor coverage and freshness will be maintained by altering the luminosity of the textured imagery to reflect age, with the freshest imagery being the brightest, and then becoming darker over time. Lastly, the interface will allow the operator to give input to the UAV route planning by marking areas as higher priority for scanning.

Finally, controlling the UAVs manually to gain full coverage in an efficient manner is difficult with a small number of UAVs and quickly becomes impractical with a larger number. We will use information gain metrics to automate tasking of the UAVs in heterogeneous terrain to relieve the user of this burden. We will do this by building a reward map based on a discretization of the search space. The reward map is updated as we update the terrain model, taking into account points of view and occlusion by the terrain. As mentioned above, operators may mark areas as high priority to also change the reward map. When the UAVs plan their paths, areas of the model that have not been imaged at all, have not been imaged recently, or have only been imaged at low resolution, will return higher reward values to the UAV path planning algorithm.

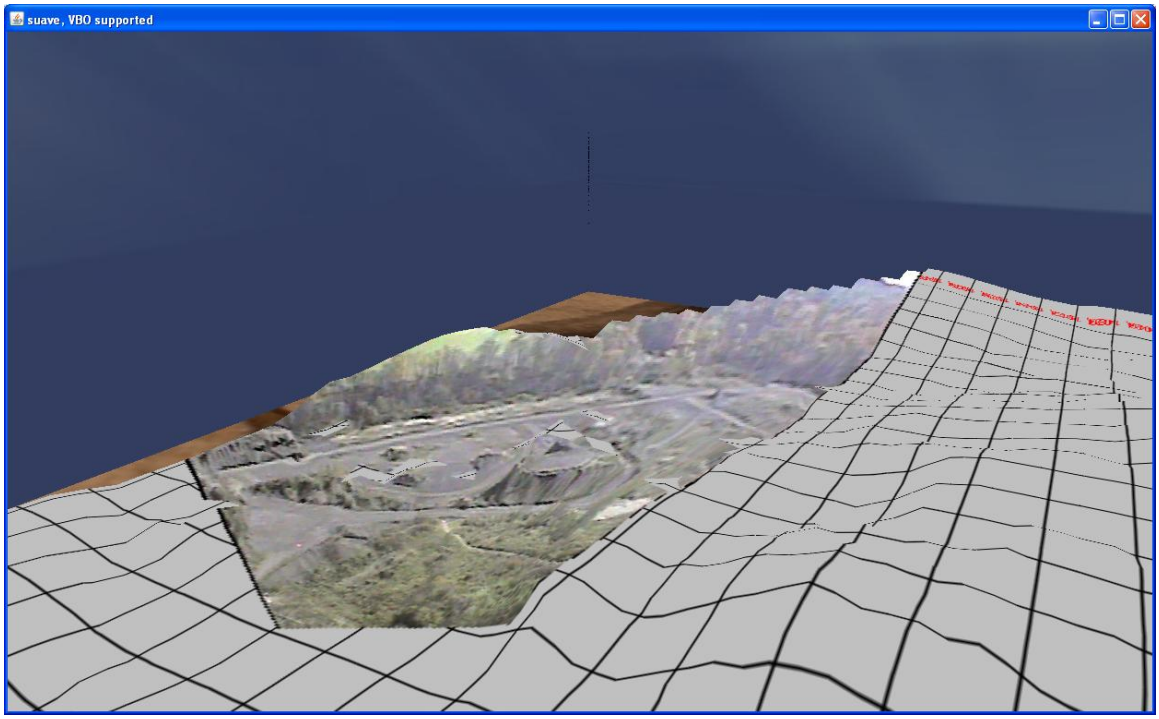

Figure 9. Video frame rendered on the terrain mesh.

Acknowledgements

This research has been funded in part of the AFOSR grant FA9550-07-1-0039 and the AFOSR MURI grant FA9550-08-1-0356.

References

[1]J. Blinn. Where am I? What am I looking at. *IEEE Computer Graphics and Applications*, 8(4):76–81, 1988.

[2]G. Calhoun, M. Draper, and J. Nelson. Advanced Display Concepts for Uav Sensor Operations: Landmark Cues And Picture-In-Picture. In *Human Factors and Ergonomics Society Annual Meeting Proceedings*, volume 50, pages 121–125. Human Factors and Ergonomics Society, 2006.

[3]G. L. Calhoun, M. H. Draper, M. F. Abernathy, M. Patzek, and F. Delgado. Synthetic vision system for improving unmanned aerial vehicle operator situation awareness. *Enhanced and Synthetic Vision 2005*, 5802(1), 2005.

[4]T. Coffey and J. A. Montgomery. The emergence of mini uavs for military applications. *Defense Horizons*, 22, 2002.

[5]M. Cummings and P. Mitchell. Management of multiple dynamic human supervisory control tasks for UAVs. In *Human Computer Interaction International Human Systems Integration Conference*, 2005.

[6]J. Drury, L. Riek, and N. Rackliffe. A decomposition of UAV-related situation awareness. In *Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction*, pages 88–94. ACM New York, NY, USA, 2006.

[7]J. L. Drury, J. Richer, N. Rackliffe, and M. A. Goodrich. Comparing situation awareness for two unmanned aerial vehicle human interface approaches. In *Proceedings of the IEEE International Workshop on Safety, Security and Rescue Robotics*, 2006.

[8]M. A. Goodrich, T. W. McLain, J. D. Anderson, J. Sun, and J. W. Crandall. Managing autonomy in robot teams: observations from four experiments. In *HRI '07: Proceeding of the ACM/IEEE international conference on Human-robot interaction*, 2007.

[9]R. Kumar, H. Sawhney, S. Samarasekera, S. Hsu, H. Tao, Y. Guo, K. Hanna, A. Pope, R. Wildes, D. Hirvonen, M. Hansen, and P. Burt. Aerial video surveillance and exploitation. *Proceedings of the IEEE*, 89(10), Oct 2001.

[10]M. Mallick. Geolocation using video sensor measurements. In *Information Fusion, 2007 10th International Conference on*, 2007.

[11]T. Page. Incorporating scene mosaics as visual indexes into uav video imagery databases. Master's thesis, Air Force Institute of Technology, Wright-Patterson AFB, 1999.

[12]A. Pitman, C. Humphrey, and J. Adams. A Picture-in-Picture Interface for a Multiple Robot System.

[13]M. Quigley, M. Goodrich, and R. Beard. Semi-autonomous human-uav interfaces for fixed-wing mini-uavs. In *Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, volume 3, 2004.

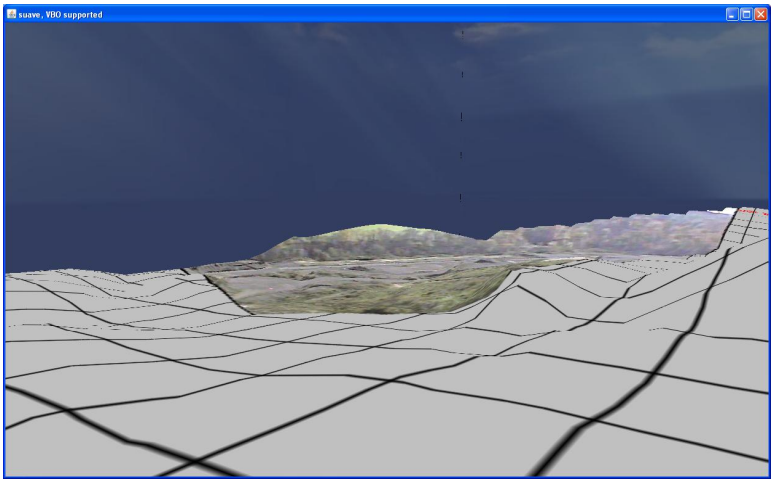

Figure 10. Video frame rendered on the terrain mesh from a lower perspective.

[14]K. Tso, G. Tharp, A. Tai, M. Draper, G. Calhoun, and H. Ruff. A human factors testbed for command and control of unmanned air vehicles. In *Digital Avionics Systems Conference, 2003. DASC '03. The 22nd*, volume 2, Oct. 2003.

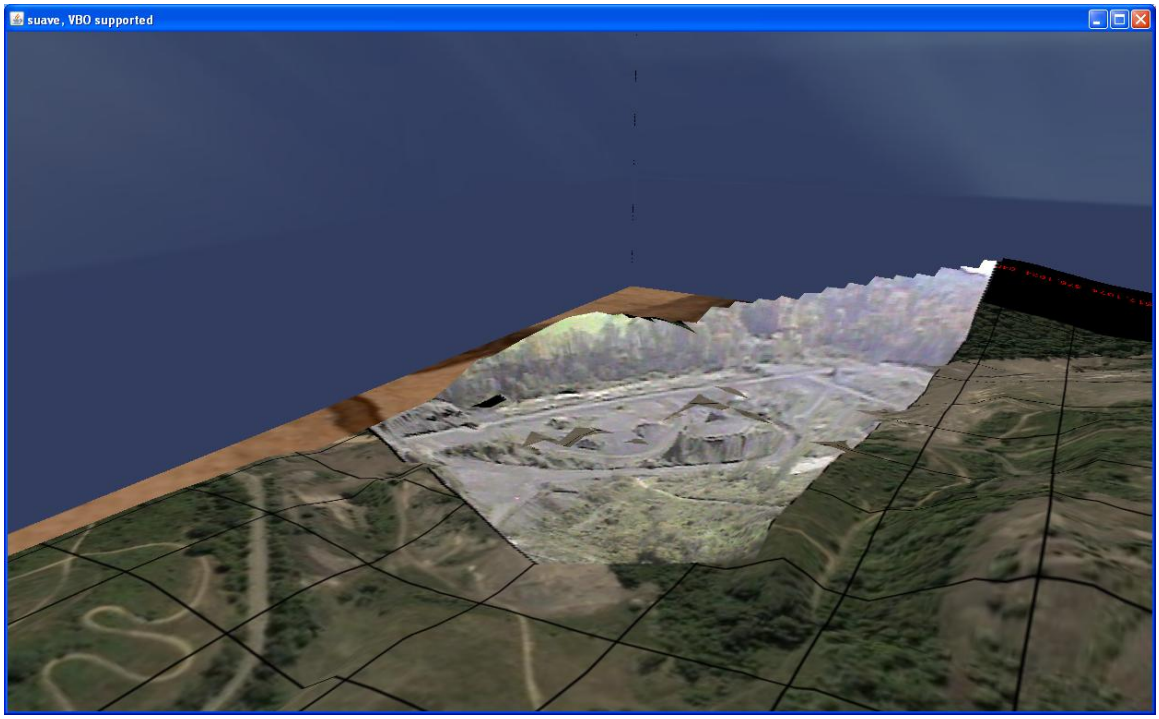

Figure 11. Video frame rendered on the a priori terrain image and mesh.