



Summary: This project explores several basic elements in video surveillance. The task is to identify and track objects in an uncluttered outdoor scene. A fixed camera video sequence is analyzed and a report summarizing recognized object position and area (in each frame) is generated.

Input Data: The input sequences show one or more Styrofoam airplanes that glide through the air, sometimes landing softly. The planes enter and exit the field of view (FoV) at different positions. There are a variable number of frames in each sequence. The images are collected at 29.96 frames per second.

In all parts, the sequences must be processed in order. *A program must not look ahead in the sequence.* The output report must contain a frame by frame summary of identified planes, including their position and area. All reported positions are in pixels and are relative to the upper left hand corner of the frame as (X, Y) where X is the row offset and Y is the column offset. Output images are extremely helpful for test and debugging; but they should be generated under the debugging flag DEBUG, which is set to zero during evaluation runs.

A library of helpful routines is provided. It also defines the project API used for performance evaluation. These conventions should not be changed.

Directory Structure: The directories are organized as follows.

./	main working directory; holds all program files.
./seqs/	sequence directory; contains sequence-named directories of jpeg files.
./trials/	trial directory; contains sequence-named directories of report files, output images.

Extracting Sequences: Create a sequence directory “./seqs” and move seqbuild.py along with the four test sequences (in QuickTime format) (“20090818-02.MOV”, 20090818-03.MOV”, 20090818-07.MOV”, 20090818-20.MOV”). Then run seqbuild.py under python under Linux or other environment that supports mplayer. This will generate all sequence directories.

Project Files: The following files are included in the downloadable project zip.

P1-2.c	shell program containing the interface API (your code goes here).
utils.c utils.h	jpeg and directory access helper routines
report.c report.h	report generation API
mmm.c mmm.h	multi-modal mean background modeling and foreground extraction library
rollers.c rollers.h	density and blob analysis library
makefile	script for compile & execution interface
seqbuild.py	a script that creates all sequence directories from QT videos

First Deadline: Projects must be performed by two-person teams. By **Monday, 24 August 2009**, please send an email to the instructors (linda.wills@ece.gatech.edu and scott.wills@ece.gatech.edu) stating who are the members of your team. Please designate one member from each team to be the *contact person* who will upload all submissions from the team.

Checkpoint Deliverables: Schedule your team for a Checkpoint meeting through T-square's sign-up tool (<https://t-square.gatech.edu/portal>) during the week of **31 August 2009**. Only one person per team (preferably the contact person) should sign up the team for a slot, but all members of the team should participate in the checkpoint. The team must prepare a short presentation on the status of your project, using the template found at http://users.ece.gatech.edu/~scotty/8893/Fall09/Checkpoint_Template.odp. This presentation should include an illustration of your approach and a summary of preliminary or intermediate results. Bring a hardcopy of your presentation to the meeting (if it is a Skype meeting, please email the presentation to linda.wills@ece.gatech.edu beforehand). To communicate via Skype, connect to "scott.wills".

Final Submission: In order for your solution to be properly evaluated, there are a few requirements.

1. The file must be named **P1-2.c** and employ the provided interface for specification of sequence name, starting and ending frame number, and frame rate. The report (report.txt) should be produced in the trials directory under the sequence name using the report generation functions.
2. Wrap debugging print statements and output image writers in **DEBUG** conditionals (defined in **utils.h**). Level 0 is evaluation execution.
3. The file must contain a documentation section at the top that includes the following:
 - a. Your names and project number.
 - b. Description of the algorithms you are using.
 - c. Summary of insights you had, technical issues you encountered, and solutions you found, and any other interesting facts you discovered in doing this project.
 - d. Summary of related work from the literature, such as algorithms you are building on, extending, or competing with. This must include a brief discussion of how your work differs, builds on, and or extends the related work.
 - e. References (bibliography and URLs) for the related work and for algorithms or code you are using that you did not develop.

4. Your solution must be properly uploaded *by the team's contact person* to the submission site before the scheduled due date, **9:00pm on Wednesday, 9 September 2009**.

Project Submission Instructions:

When you are ready to submit a file as your answer to a part of the project, use your web browser to go to the web site: <http://www.ece.gatech.edu/~scotty/8893/projects/index.html>.

Click on FILE UPLOAD which will take you to the ECE 8893 File Upload page. This contains a form for you to enter your GT ID (e.g., gteopb) and the name of the file to upload (e.g., C:\My Documents\Framework\GeorgeBurdell.jpg).

Double check that the name of the file uploaded is the one you intended to submit.

If you submit a file as the answer to part of the project and later you would like to submit an improved answer, you may submit the more recent version of the file. Only the most recent one will be graded. However, versions submitted after the due date will not be graded.