

# BASEBALL BAT ASSIGNMENT

---

## TASK OVERVIEW »

To track the bat in baseball we first need to extract the bat from individual frames. This is the task for this assignment. You are given some PNG images showing different swings and different camera views, and the task is to extract the coordinates of the bat in the images.

## INPUT DATA »

PNG files containing images of batters from different swings and viewpoints. All frames are independent--there is no requirement to track over time or “triangulate” from different viewpoints, all that is required is 2D bat extraction.

You may use a region of interest per frame around the bat and just process that rectangle. This makes the task significantly easier. (It is not unrealistic that a real system could use this--for example, the system may track the ball through the strike zone, which means that the bat position must be near the strike zone at that time.) If you use a region of interest, you must draw this on your output images.

Ideally your program would be robust to other similar input files, rather than highly tuned such that it only works on the input files provided, however, we do not expect candidates to get a full solution in the time available. We are interested in seeing your approach, the quality of your code, and how far you get towards finding something that could be robust with further development.

## REQUIRED OUTPUT »

**IMPORTANT NOTE: YOU MUST INCLUDE THESE OUTPUT FILES ALONG WITH YOUR CODE WHEN YOU SUBMIT. IT IS NOT ENOUGH THAT YOUR PROGRAM CAN CREATE THESE FILES BY RUNNING IT, YOU NEED TO ATTACH THE OUTPUT FILES.**

1. A text file which gives the position of the bat in each image. It is up to you to decide the best way to encode the bat position and the format of this file.
2. PNG images showing where your program has located the bat in the image, i.e., some drawing over the top of the original image to show that your algorithm is working. **IF YOU USE A REGION OF INTEREST, YOU MUST DRAW THIS ALSO.**

## GENERAL SOFTWARE REQUIREMENTS »

1. The solution must be written in C++ but can be on any platform you choose.
2. You should write production quality code.
3. The components of your solution should be re-usable. i.e., they would plug-in to a larger piece of software without refactoring or re-organising.
4. Uses classes and ensure that public interfaces are as clear and useful as it is possible to make them.
5. You may use libraries such as opencv and take any approach that you feel is suitable for solving the problem.

## GET IN TOUCH »

If you have any questions regarding the task please contact Anthony Daniels:  
[anthony.daniels@hawkeyeinnovations.com](mailto:anthony.daniels@hawkeyeinnovations.com)