

# ICRA2016 Tracking Writeup

Nate Chodosh

August 29, 2015

## 1 Image Processing

The main data type of the tracking algorithm is the Model which stores the relevant information for tracking and exposing a MBR. The fundamental representation of a Model is its contour, which consists of a set of points that make up the edge of the structure. The Model implements tracking with a series of vectors that store the object's center, orientation, and the contour found in that frame for each processed frame. The image processing aims to update this data type every frame with new information.

The main step in processing is filtering and locating contours. The first filtering step is to remove the blue channel from the image to separate the structures from any light projected on the stage. Next the image is converted grayscale and then to a binary image using an adaptive threshold which sets any pixels near rapid intensity to 255 (white) and every other area set to 0 (black). This effectively highlights the edges of the contour as well as any other noise in the image. The white areas are then expanded using an erode method to close any small gaps in the edges of the structures. Then the find contours method is used to extract all of the shapes present in the image. Finally, some of the noise is filtered out by only taking contours with an enclosed area in a specified range.

The output images used to project blue light also require some processing to account for the distortion inherent in the optical setup. Before tracking the Models some calibration is in order to correct for the distortion. An auto calibration routine projects 4 points in succession on to the stage, asking for user input to confirm that the algorithm has identified where in the camera coordinate space the projected points correspond to. From those 4 pairs of points the perspective transformation between the DMD coordinate space and the camera space. Applying the inverse of this transformation before displaying images on the DMD effectively negates the distortion.

## 2 Real Time Contour Tracking and Exposure

The first step in tracking is to find all of the potential models. A model object is created for every viable contour which the algorithm then begins to track while the user selects which models are of interest. Then the tracking is restarted using only those models.

The actual tracking itself is performed by capturing a frame and using it to update all of the models from the previous frame. For each model a small region of interest around the previous position of the model is identified and contours are extracted from it using the previously described method. Then these contours are searched for the one whose area most closely matches the area of the initial model. If the two areas are close enough the model is considered to be initially 'found' in that frame. If the object is found the contour is then used to determine the other two pieces of tracking information, the center and the orientation. <REFERENCE FIGURE HERE?>

Calculating the center is done by simply computing the center of mass of all of the contour points. Determining the orientation is more difficult since the algorithm is designed to operate on arbitrary contours which could have any amount of rotational symmetry. To achieve this the algorithm generates a polar histogram of the contour centered at the object's center. The histogram has 360 slots, one for each degree. Each slot contains the average distance from the center to all points in the contour that lie in that direction. In this form a rotation of the object corresponds to a phase shift of this signal. Calculating the phase shift that gives the minimum change in angle from the previous frame then gives the new orientation. If the found orientation signal is too different from the reference one the model is not considered found anymore. When a model isn't found the algorithm assumes no change from the previous frame.

While the tracking occurs the user is presented with a visualization of all the tracked models current state. From that window the user can select which model to expose at any time. There are two categories of exposure to select from. Within each category the user can select different types of exposure for each model which are then toggled on off throughout the experiment. There are many types of exposure but the only ones used in these experiments were an exposure of a rectangle over one half of a contour and a solid block over the entire contour.

### 3 Gear Tracking

Due to the high amplitude noise in the <NAME FOR GEAR DATA> a completely different algorithm is needed for tracking. The core concept in the gear tracking is unraveling the input image into a polar representation and then using a cost function on the pixels to compute the lowest cost path from the top of the image to the bottom in order to extract the edge of the gear, where the cost of a path is the sum of the costs of each pixel traversed. The polar representation of a frame can be seen in <GEAR FIGURE>. Depending on the exposure state of the gear different cost functions are used to acquire the edge.

The edge found (shown in blue in the figure) is converted back into Cartesian coordinates. Then a circle is fitted to the data to find the center of the gear. The edges found by this process are still too noisy to be used in their raw form to identify the orientation of the gear. Instead the edge in its polar form, which is simply a function from angle to distance to the edge is searched for the tips of the gear. A pixel is considered a tip when it is greater than all of it's neighbors in a vicinity approximately the size of a gear tooth (shown in green on the figure). Only considering the angle of each tip simplifies the data to set of numbers which are then fit to a set of 12 equally spaced numbers with an offset from 0 degrees (shown in read on the figure). The best offset then gives the orientation of the gear.