**Motivation**

Demonstration is an important part of teaching people how to program for the first time.

Typing code ties the lecturer to their laptop. This is problematic because it reduces lecturer mobility, reduces eye contact and engagement with the class, and makes it difficult and unnatural for the lecturer to point out things in the code.

Handwriting code on the board solves all of these problems. However, it also introduces another one: the code cannot actually run. Often times, it is useful to show the end result.

**Future**

- Gesture based invocation of Codeable.
- More advanced whiteboard skew correction.
- Integrate Codeable with projector to support interactive debugging, including setting breakpoints, viewing local variables.
- Use projector to overlay compiler errors onto whiteboard.
- Support for scripting languages like Python.
- Generation of personalized user dictionary via automated scanning of user code.

**Image Processing Pipeline**

**Image Acquisition**

Android phone is running "IP Webcam" webserver. RGB image is sent to laptop via Wi-Fi for Matlab preprocessing.

**Image Pre-processing**

- Image size normalization
- Binarization using Otsu’s Method
- Background estimation via morphological erosion
- Use region labeling to remove island regions
- Hough transform according to vertical lines of letters

**Experimental Results**

Improving the accuracy of Tesseract:
1. Deskew image
2. Custom dictionary with C keywords and common function and variable names.
3. Increase strength of dictionary.
4. Unambiguous handwriting font.

Post-processing tricks:
1. Compute Levenshtein edit distance for first few lines. Replace bad #includes.
2. Use regular expressions to match parenthesis and quotes.
3. Automatically append semicolon.

```c
#include <stdio.h>
int main (void){
  printf("Hello World!");
  return 0;
}
```