Integrated Visual Representations for Programming with Real-world Input and Output

**Background**
Programs with real-world input and output (I/O) have become popular.

**Problem**
Gulf of execution and evaluation in programming has been deepened.

Programming with example data is needed.

**Our approach:** Integrate graphical representations of the real world into text-based IDEs

**Existing approaches:** Programming by Example is good for the end-user but not always the best for the programmer. Text-based programming + external tools help the programmer but could be better when integrated.

**Related work:** Visual Programming focuses on new kinds of visualization of program code and data while we investigate use of existing graphical representations: photos and videos.

So, how can we integrate graphical representations?
We model the program as follows and assign graphical representations to each component.

\[ \text{out} = f(\text{in}, c) \]

where

- \(c\): constants - static input to the program
- \(\text{in}, \text{out}\): variables - dynamic input and output of the program
- \(f\): functions - specification of the program

**Photos for understanding static data used in the program**
Graphical representations of constants, [Kato et al., CHI’13]

**Videos for understanding dynamic behavior of the program**
Graphical representations of variables, [Kato et al., UIST’12]

**Annotations on photos & videos for specifying program behavior**
Graphical representations of functions

1. create an image processing component by annotating example video input
2. connect components to build a graph and see it in action by playing the video
3. edit component implementation if needed

**Future outlook**
3D graphical representations, Multimodal programming, Everyone as a programmer, Liveness in programming

**References**