

Combining Programmatic and Direct Manipulation

Ravi Chugh, Brian Hempel, Mitchell Spradlin, Jacob Albers



What if you could change a program just by changing its output?

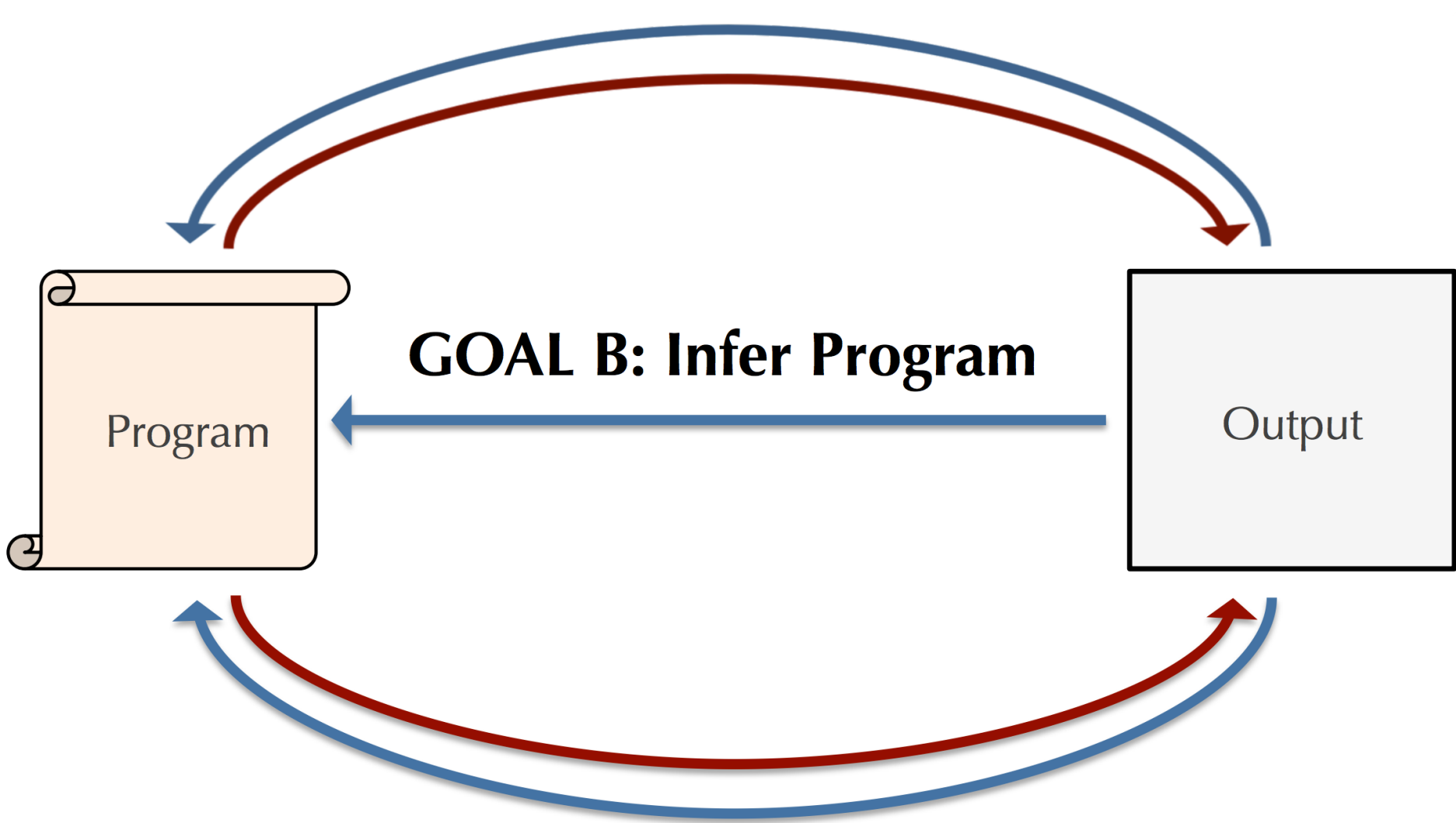
1 Programmatic vs. Direct Manipulation

	Programmatic Manipulation	Direct Manipulation	VISION: "Prodirect" Manipulation
Immediate Feedback	✗	✓	✓
Rapid Prototyping	✗	✓	✓
Abstraction Capabilities	✓	✗	✓

2 Challenges and Goals

GOAL A: Live Synchronization

Infer "Small" Program Update During Direct Manipulation



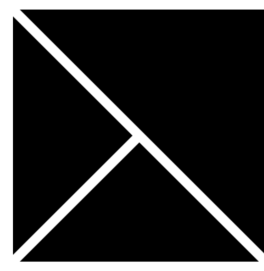
GOAL C: Ad Hoc Synchronization

Infer "Large" Program Update After Direct Manipulation

3 Applications

Many kinds of programming might benefit from **prodirect manipulation**. Good first targets are graphical applications such as data visualization, animation, HTML/CSS, and slide show presentation. Our initial tool explores prodirect creation of vector graphics.

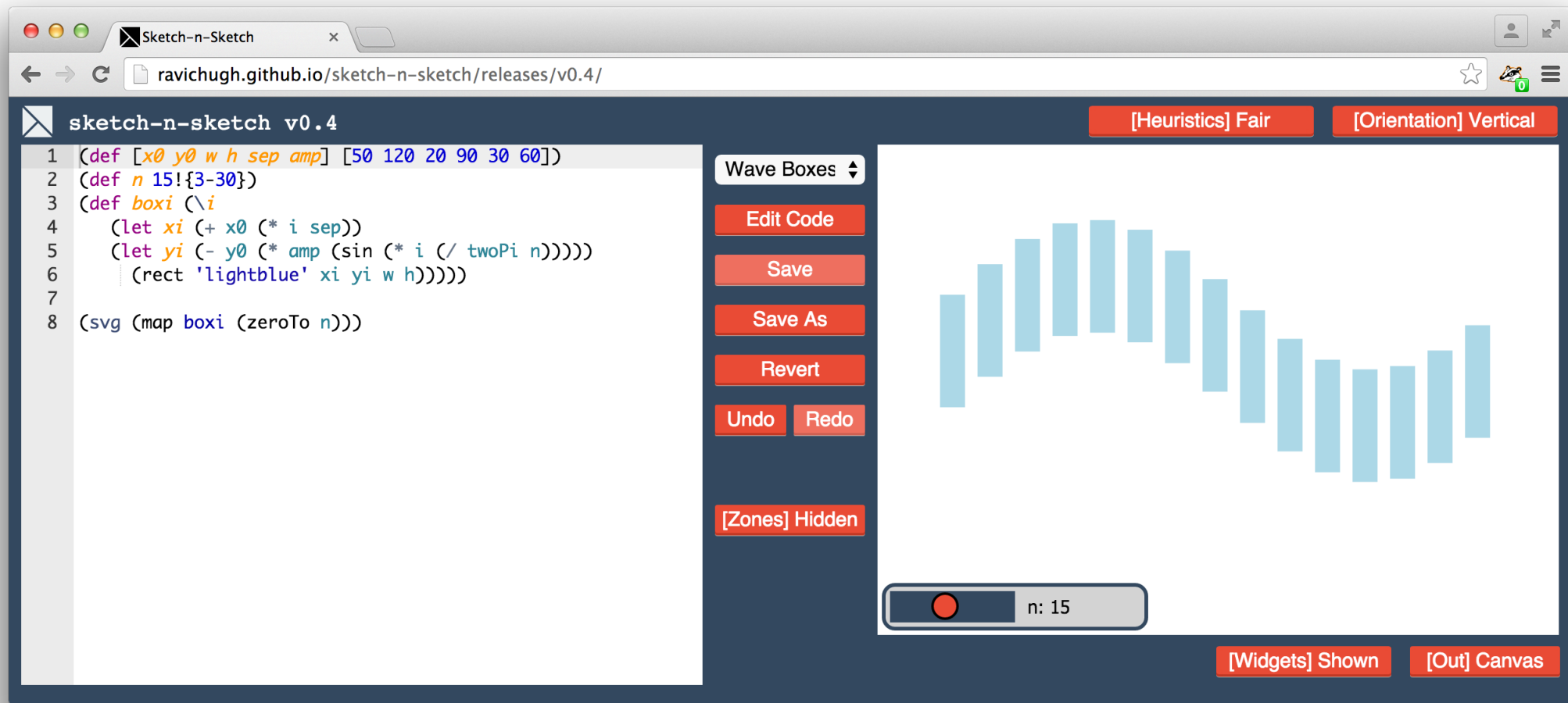
4 Prodirect Manipulation for SVG



Sketch-n-Sketch

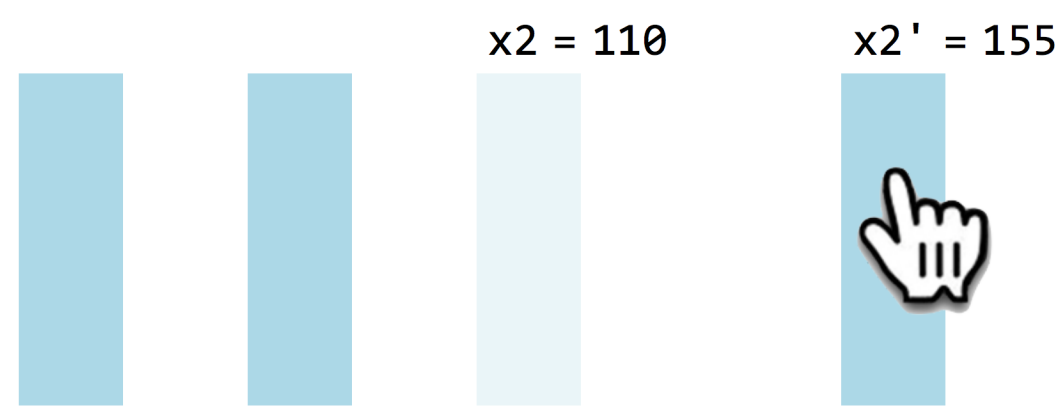
ravichugh.github.io/sketch-n-sketch

Users write programs to generate SVG vector graphics and then directly manipulate the graphics output on the right to change the program on the left.

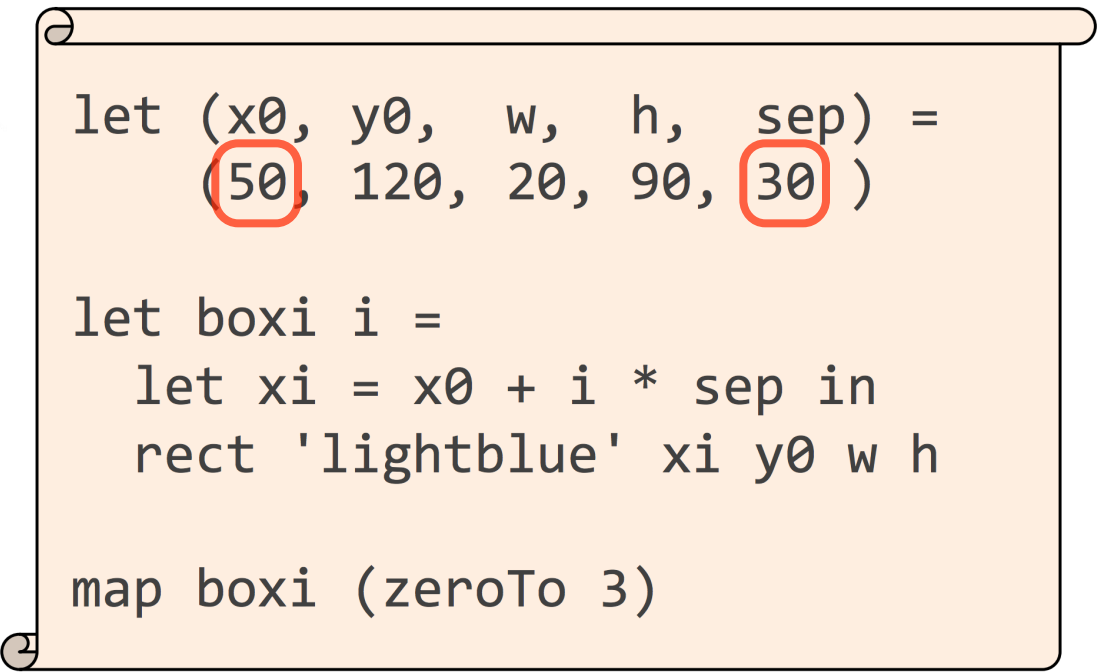


Sketch-n-Sketch supports **live synchronization** (Goal A) and has growing support for directly **drawing shapes** (Goal B) and interactively **relating shapes** (Goal C).

Live Synchronization



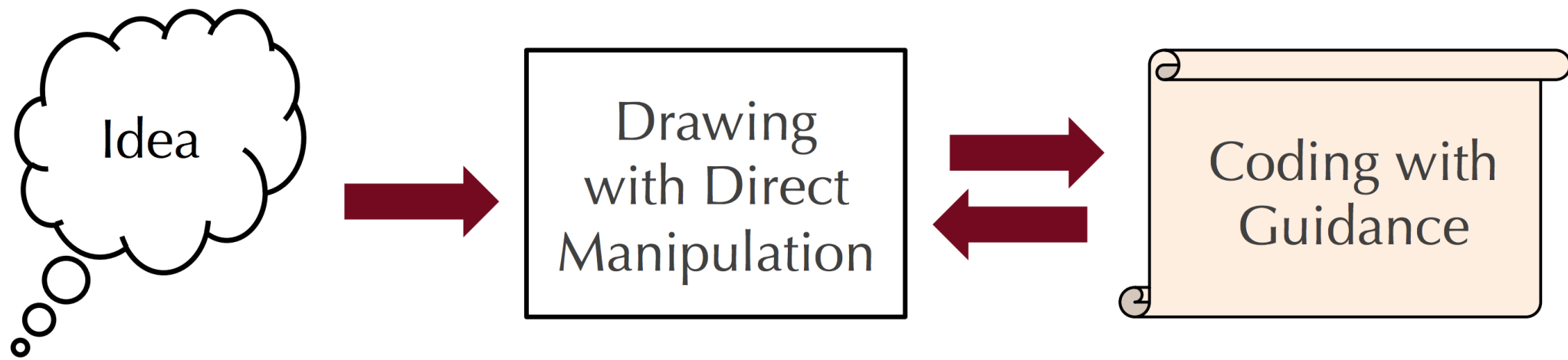
User drags object to new position...



Program updates in real time.

When there is ambiguity about what parts of the program to change (red boxes above), Sketch-n-Sketch uses heuristics to choose which locations to modify.

5 The Future



- Novel User Interface Capabilities
- Intelligent, Interactive Program Synthesis
- Domains Beyond Graphics