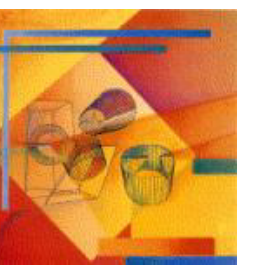


An Approach to Remote Direct Pointing using Gray-code

Makio Ishihara and Yukio Ishihara

Fukuoka Institute of Technology
m-ishihara@fit.ac.jp

Yamaguchi University
i.yukio@yamaguchi-u.ac.jp

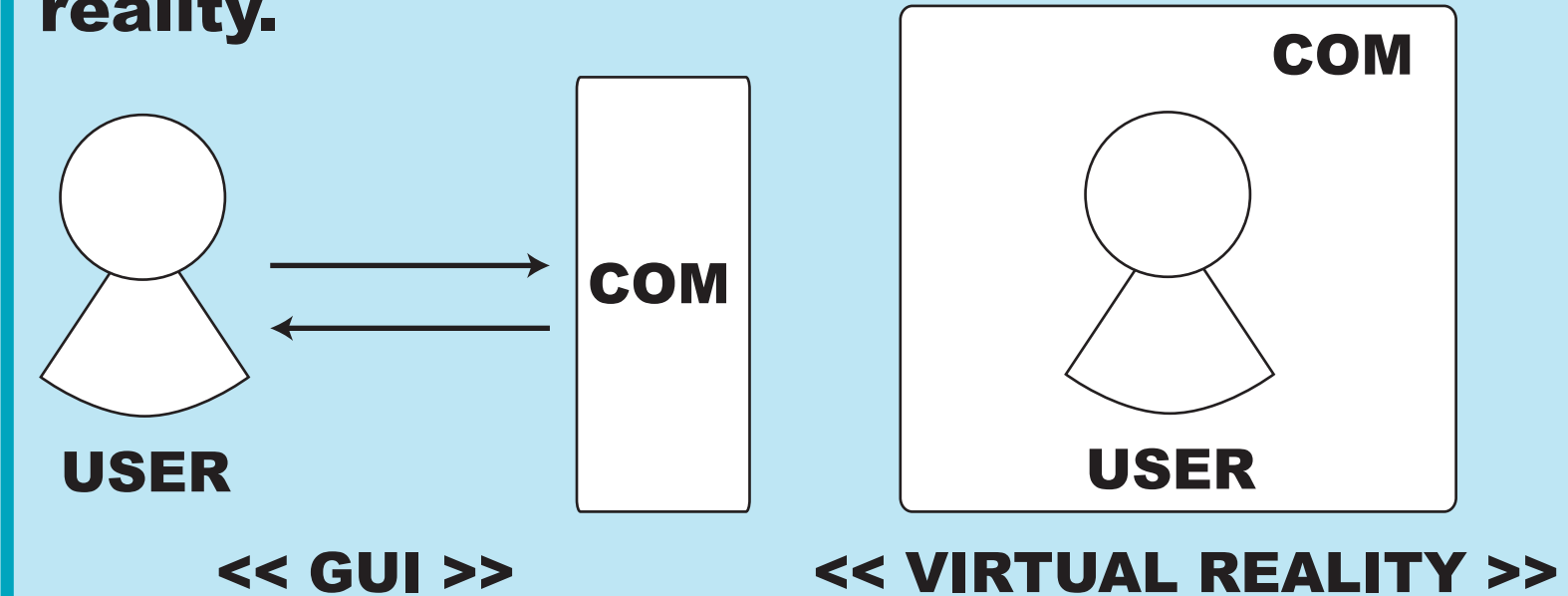


AVI2006

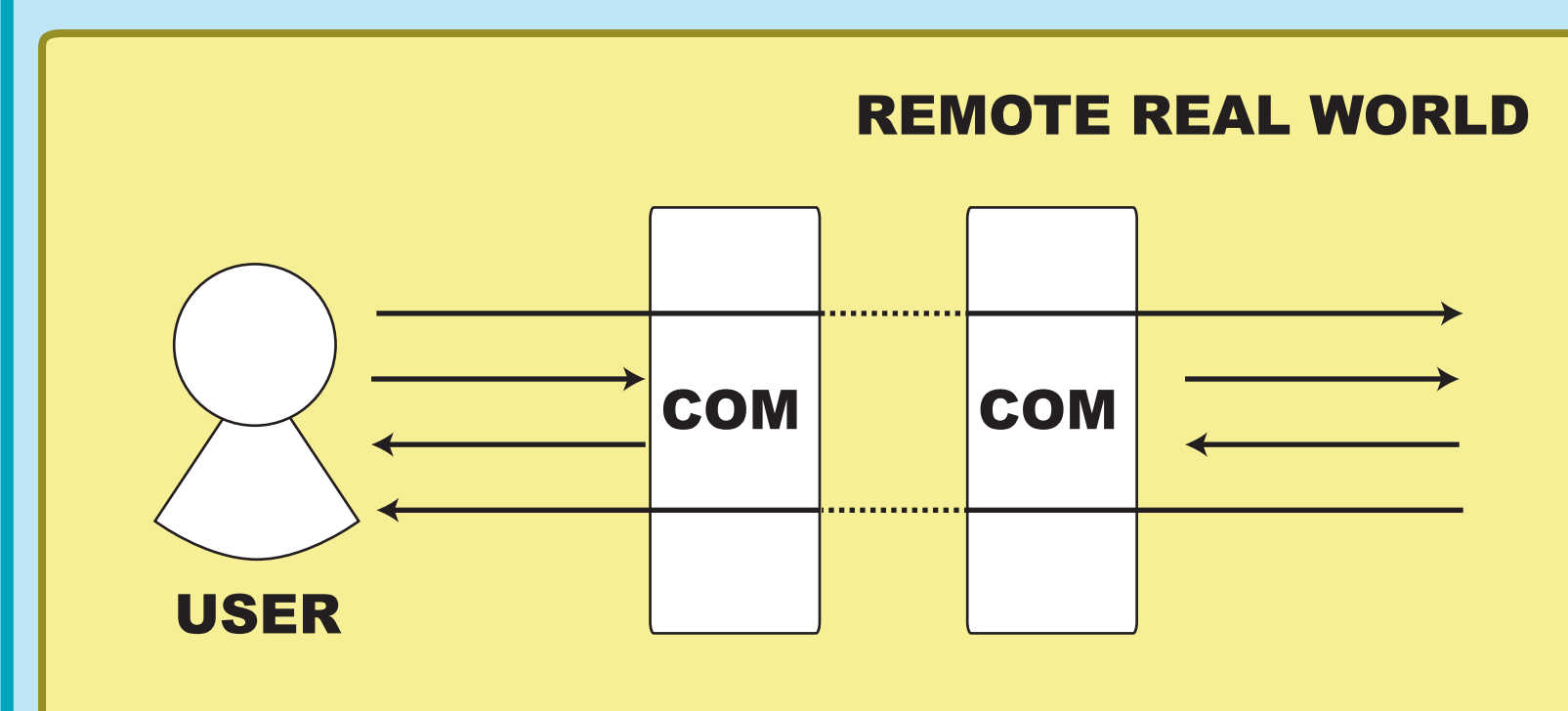
International Working Conference
on Advanced Visual Interfaces
May 23-26 2006, Venezia, Italy

Aim of the study

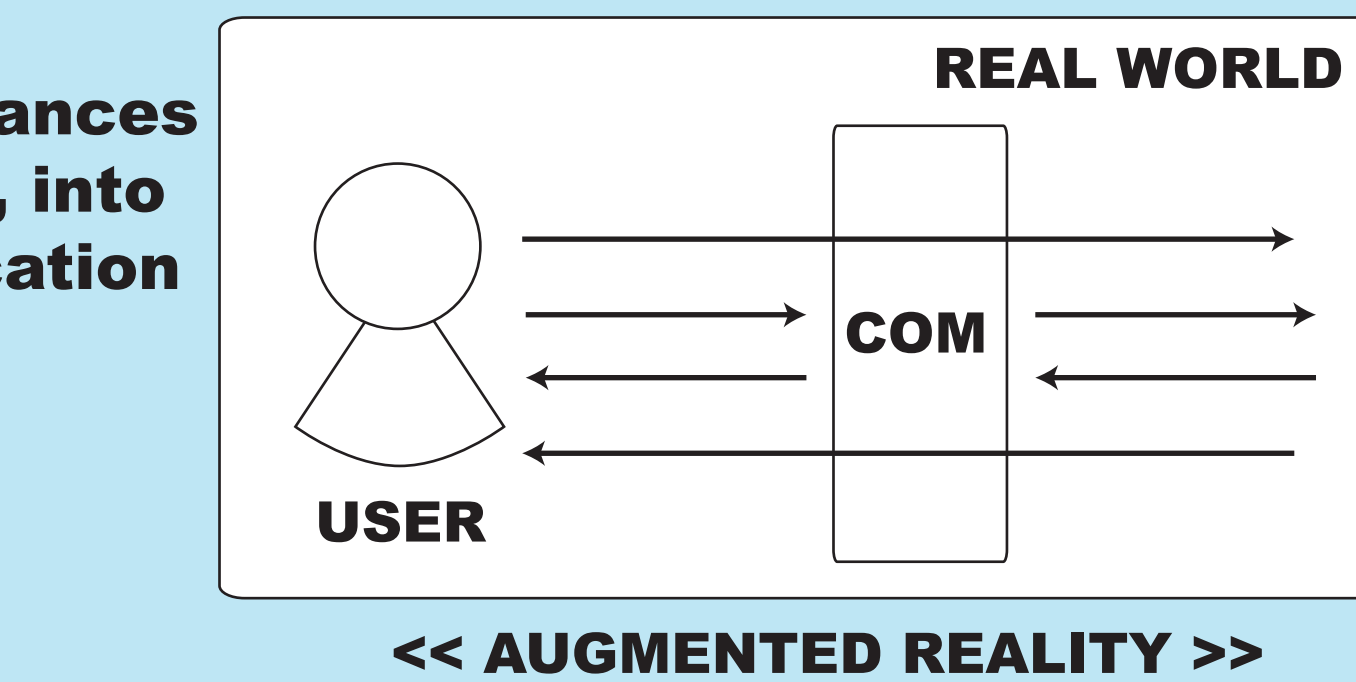
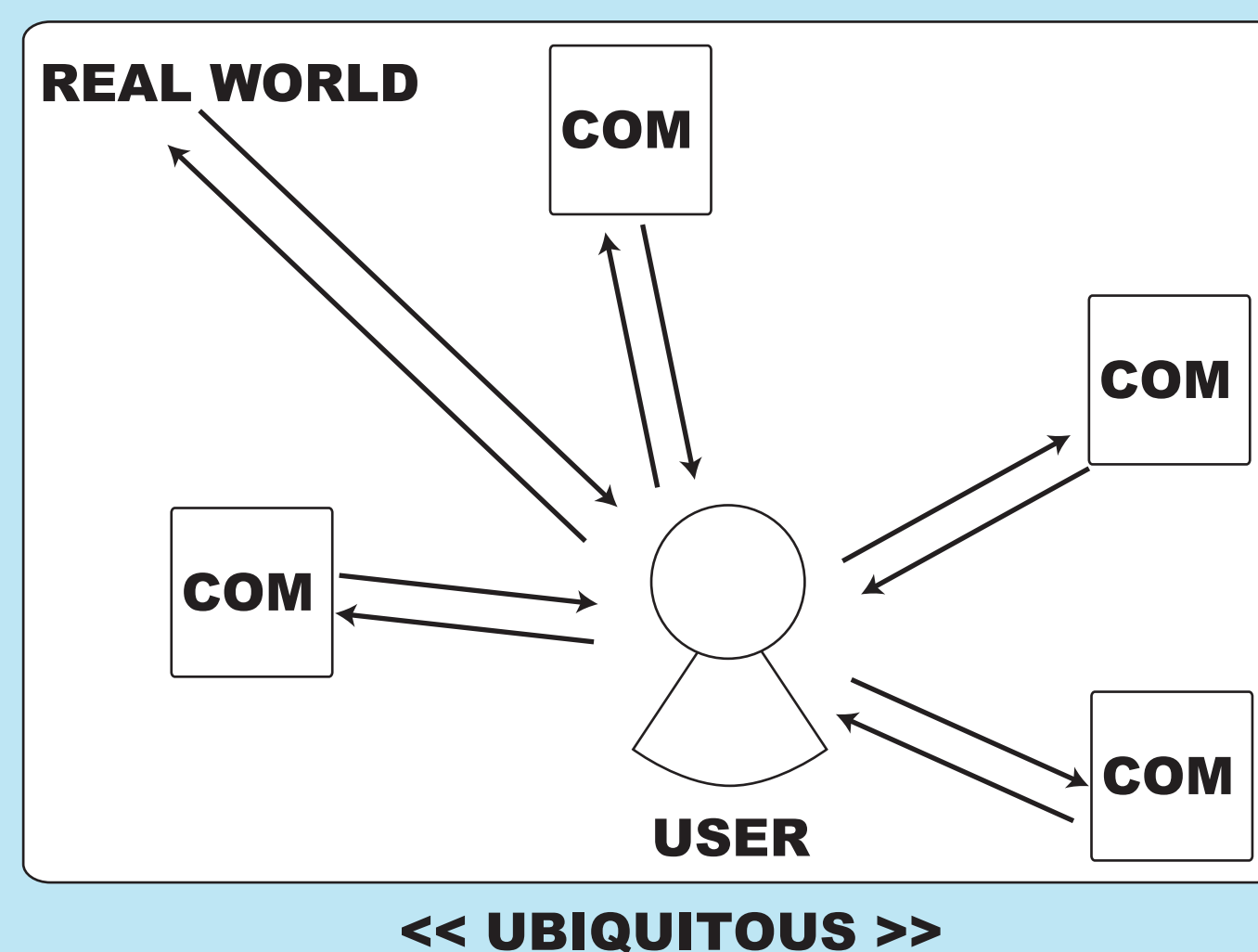
HCI Styles were introduced by Jun Rekimoto in 2001. He described the interaction styles among Human, Computers, and Real world as GUI, Virtual, Ubiquitous, and Augmented reality.



We extend AUGMENTED REALITY, that enhances communication between users and real world, into the interaction style that enhances communication between users and *remote* real world.



Goal is to show a prototype of this style



and describe how the interaction of the style works. A prototype enables remote users point at remote objects and draw directly on them. Remote users can see the objects with marks they have drawn.

We call it *Remote Direct Pointing*.

Remote Direct Pointing

Description

A camera captures images and sends them to Cam# view.

Gray-code generator generates the corresponding gray-code binary pattern to the number from Gray-code manager and then sends it to Projector.

Application gets the discovered location from Gray-code manager and then sends the computer screen to Projector.

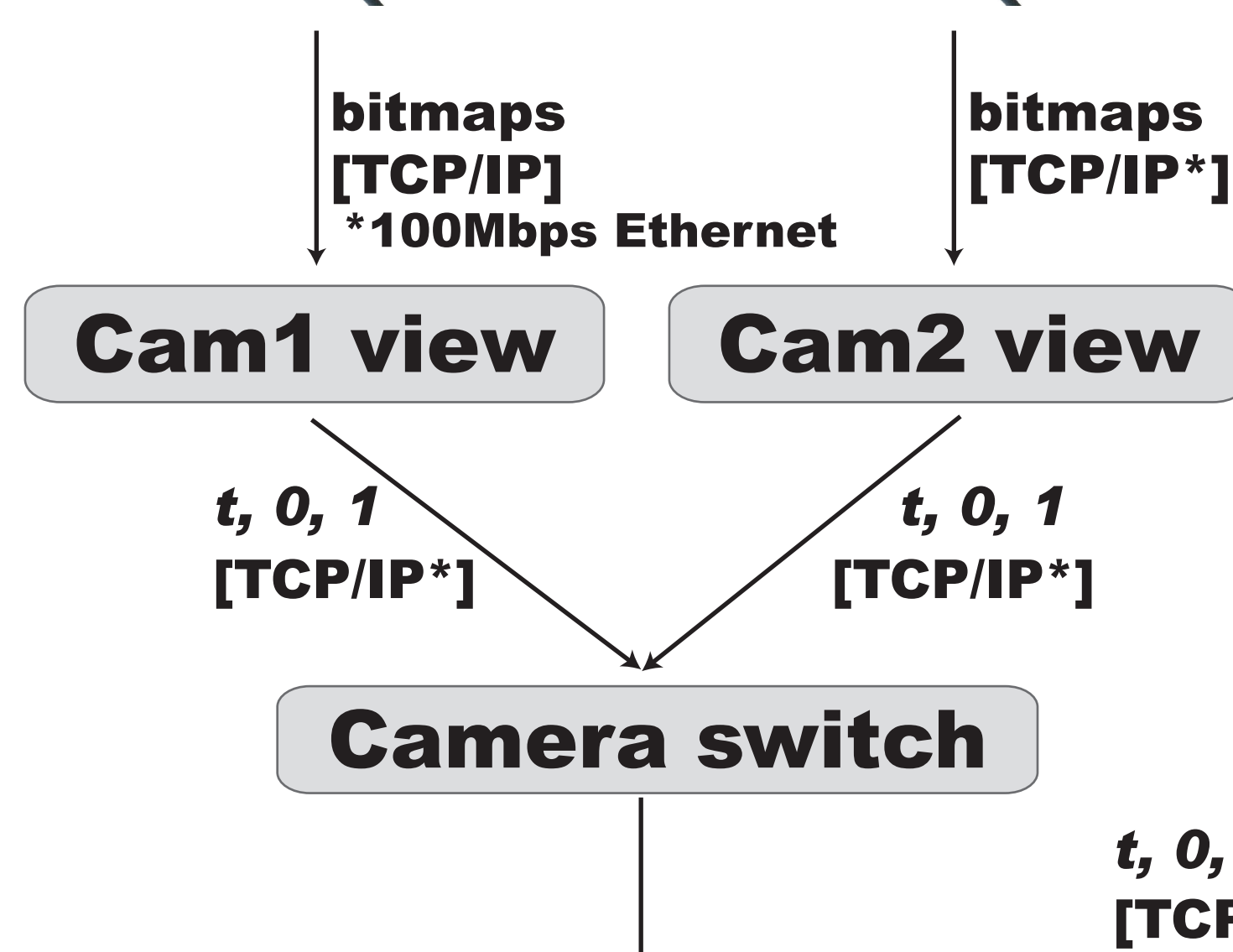
Cam1
Logicoool Qcam QV-700NHS
320*240 in Reso. 30fps

Cam2

256 x 256 pixels

8 bits binary B/W gray-code patterns

Projector
BENQ Digital Projector PB2120
1024*768 in Reso. 60Hz



Cam# view gets images from the camera and sends one of three signals t , 0, or 1 to Camera switch.
 t : Trigger to start projecting gray-code binary patterns, 0/1: Response to gray-code binary patterns.
Cam# view also works for user interface. It has a screen to display images from the camera. It sends t when a user clicks on the screen, and then it captures black(0) / white(1) at the pixel.

Camera switch transfers signals from the specified Cam# view to Gray-code manager until the current gray-code processing is completed.

Gray-code manager controls Gray-code generator as following the signals from Camera switch. It also sends the discovered location to Application at last.
For t , it sends the initial binary pattern's number
For 0/1, it stores them and sends the next number

Performance

It takes 150ms to respond to a gray-code binary pattern correctly. It takes 2.4 sec to discover a location because 16 patterns are required (8 vertical and 8 horizontal). It equals 0.417fps.

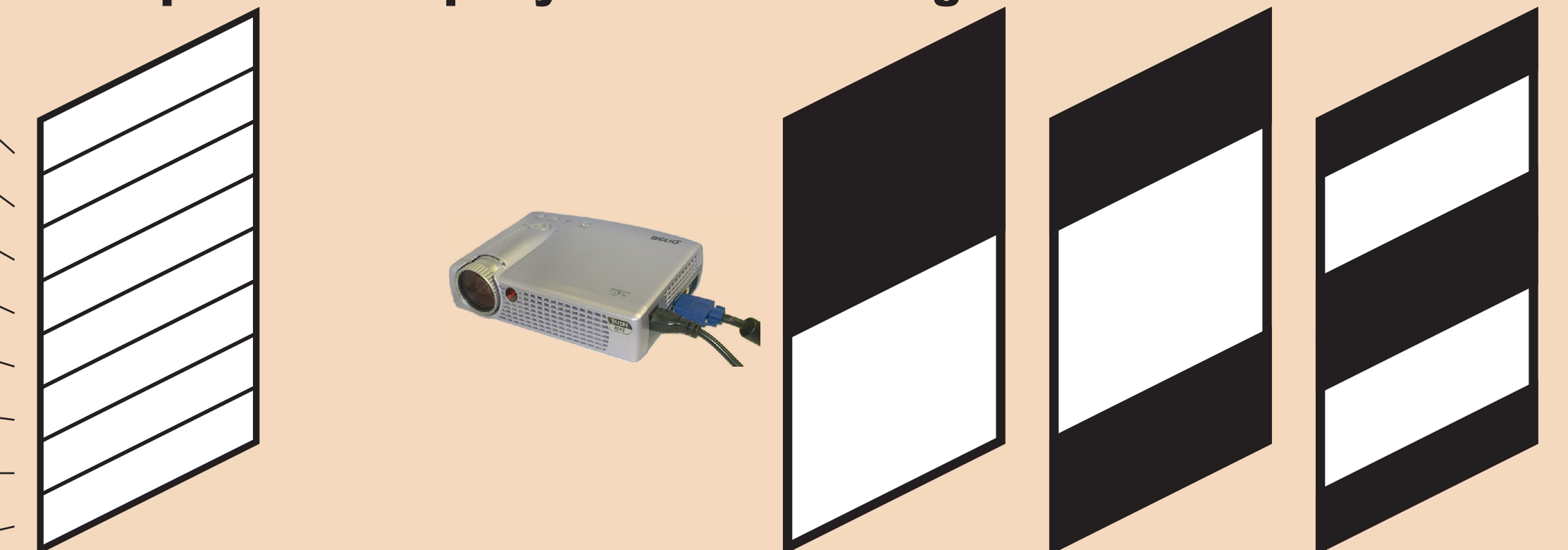
What is Gray-code?

Gray-code is a way to encode values into bits binary sequences with a property that only one bit changes between any two consecutive sequences.

For three bits binary sequences,

0. 0 0 0
1. 0 0 1
2. 0 1 1
3. 0 1 0
4. 1 1 0
5. 1 1 1
6. 1 0 1
7. 1 0 0

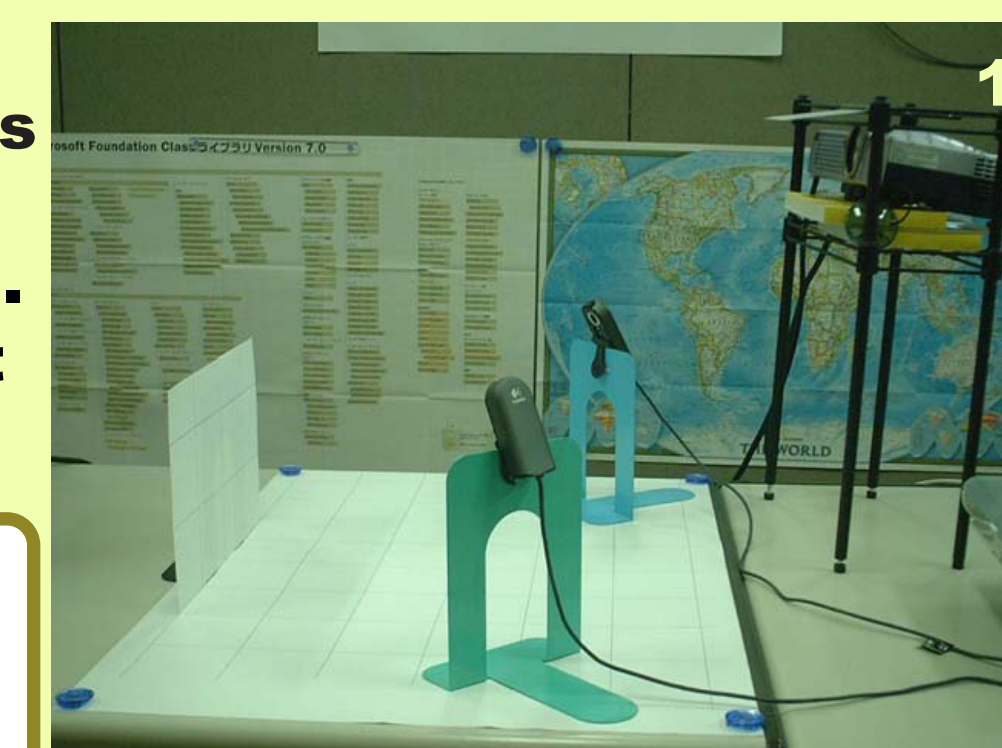
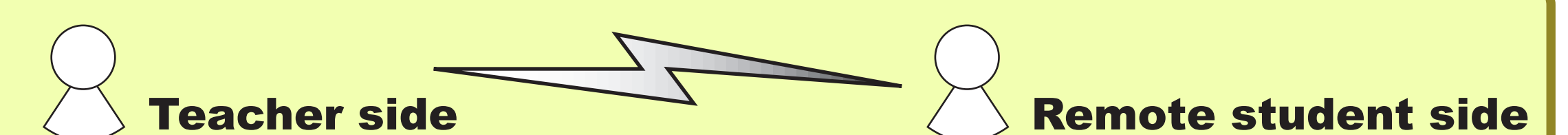
Gray-code binary patterns are the visual representation of Gray-code. They are a series of black and white stripes that represent the specified bits. As for projecting them in order, the space is uniquely divided into regions.



Due to the property of Gray-code, 1-bit change between consecutive sequences, only 1 strip-boundary occurs at the same location. A region can be identified within +/-1 pixel error.

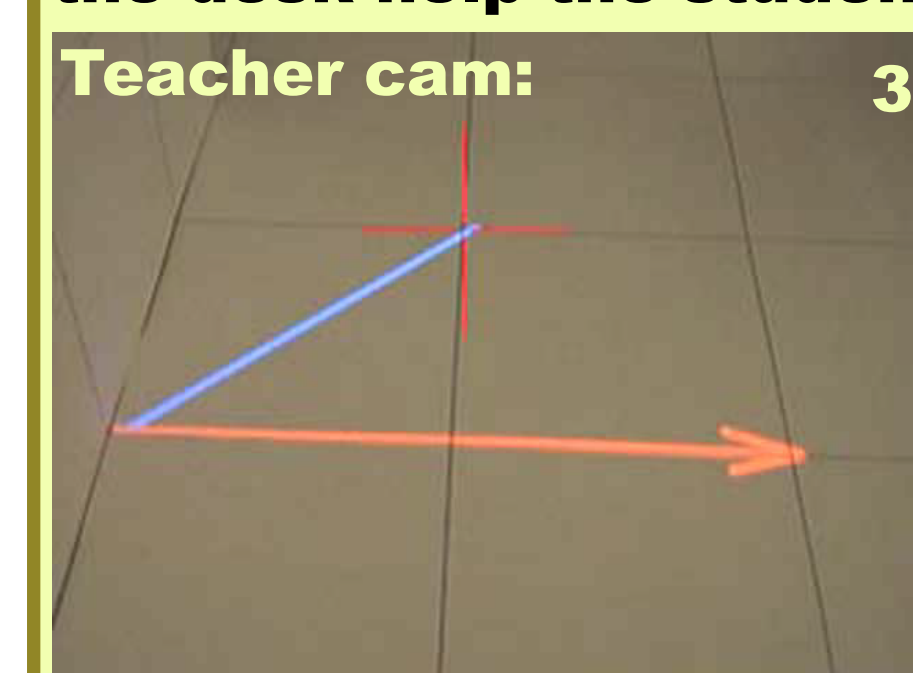
Demonstration

Scenario is that a teacher explains an outer product to a remote student using Remote Direct Pointing. The teacher is on site and is also an operator of the system. The student takes part in through the web.

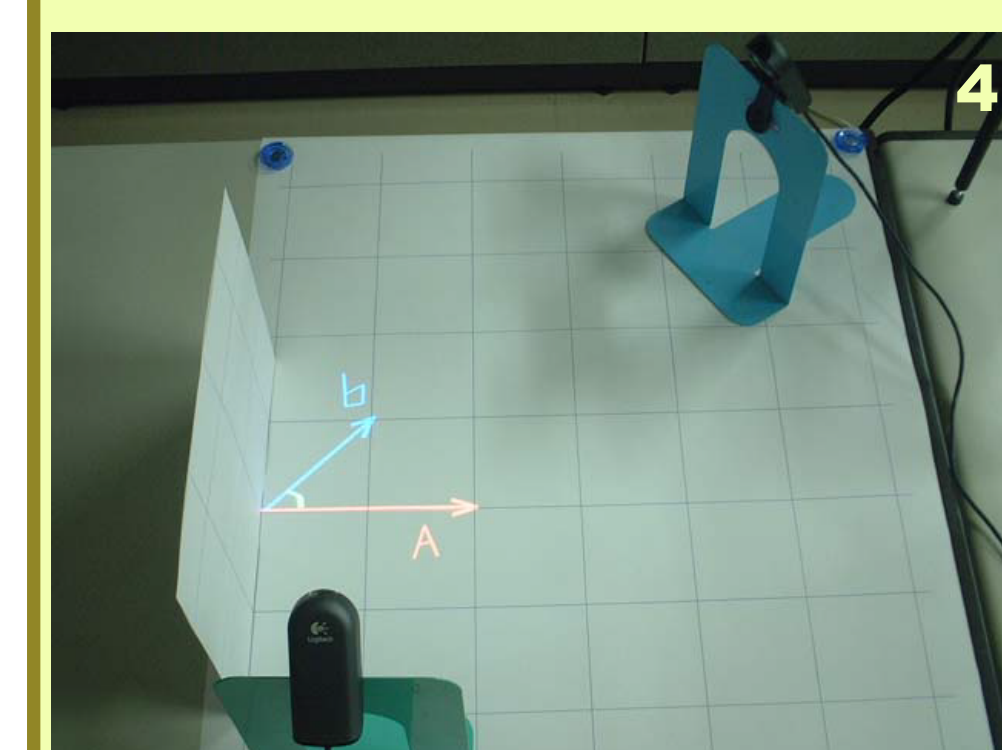


[Setup of Remote Direct Pointing]
A projector can be seen on the rack in the top right corner as well as two CCD cameras in the center. The projector projects a computer screen and also reflects it onto the desk by the mirror that is attached over the projector. The teacher uses the front camera and the student uses the back one.

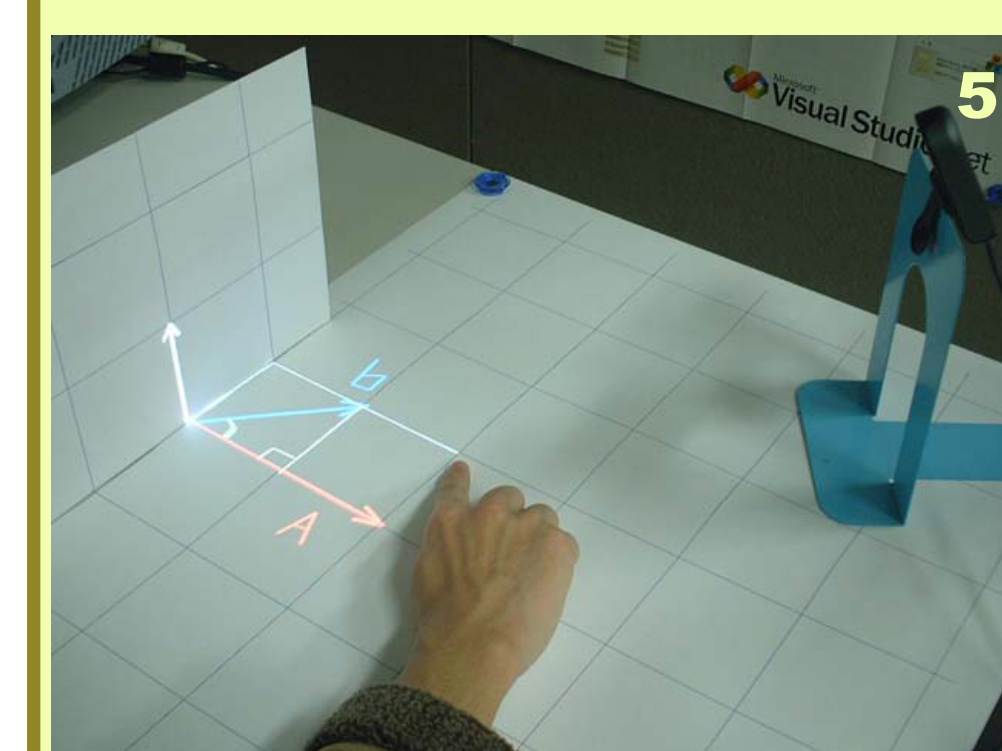
2. The student sees the desk only through the student camera. The coordinates on the desk help the student draw a diagram.



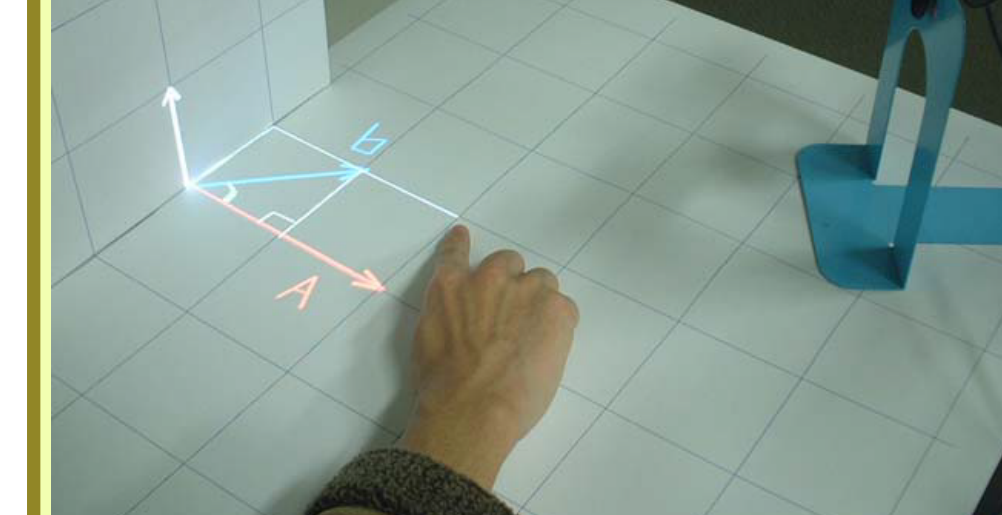
3. The teacher is drawing two vectors on the desk. The teacher positions the red cross sign representing a pointer and clicks on it to start and end a line.



4. The teacher has finished drawing the two vectors and symbols. The symbols are made up with lines. Next the teacher asks the student to draw the direction of the outer product $A \times B$ and the magnitude.



5. The student has drawn the direction of the outer product but not the magnitude. The teacher is advising the student on the magnitude by pointing at the location with his hand.



6. Finally the student is drawing a rectangle representing the magnitude.

Future work

Change of viewpoint - Remote users are required to ask the operator to move their viewpoint at the moment.

Drawing freehand - In our prototype, we can draw lines not squiggles. We will improve the performance in terms of FPS.