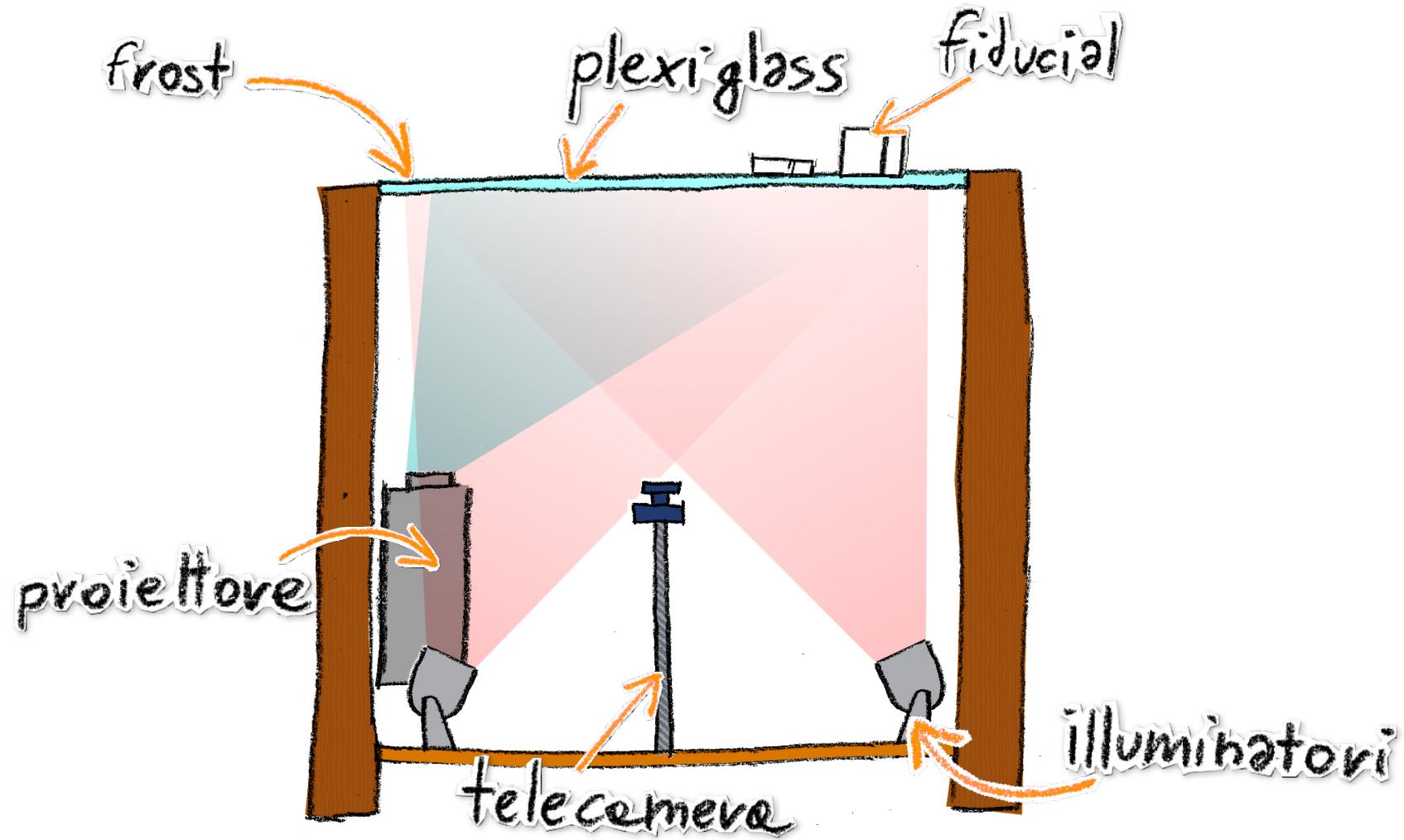
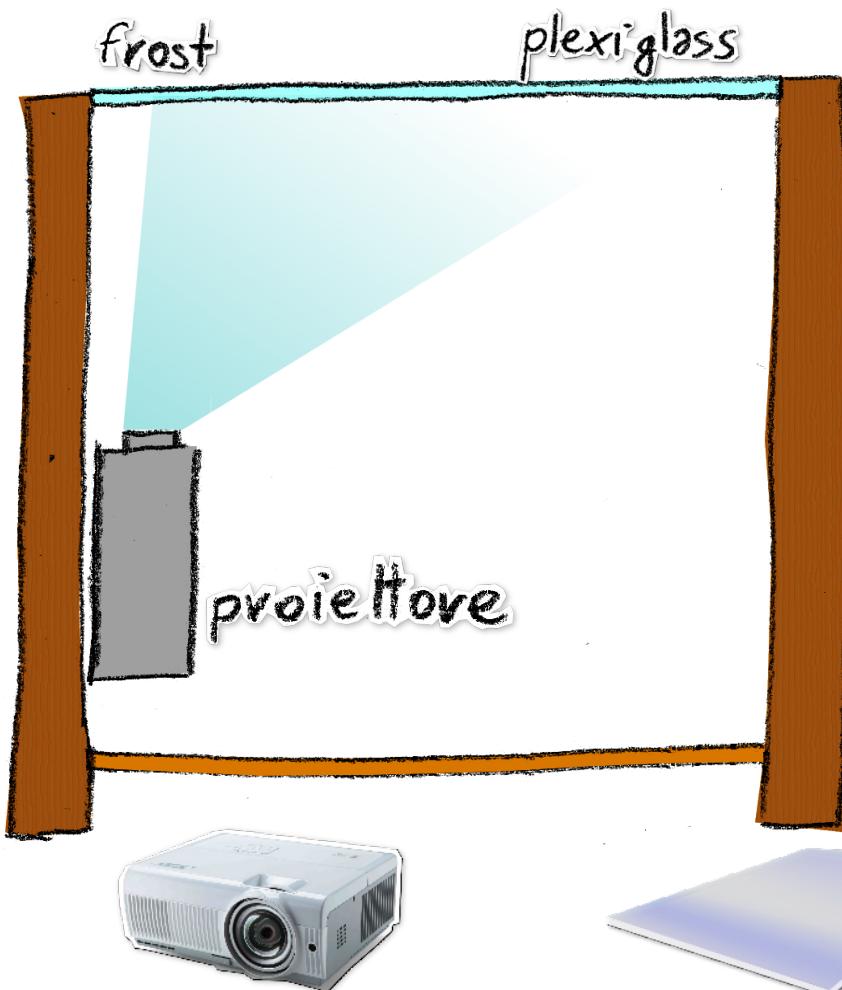


# Architettura del tavolo



# Architettura del tavolo

## Immagini nel visibile



Proiezione dell'immagine visibile sulla superficie

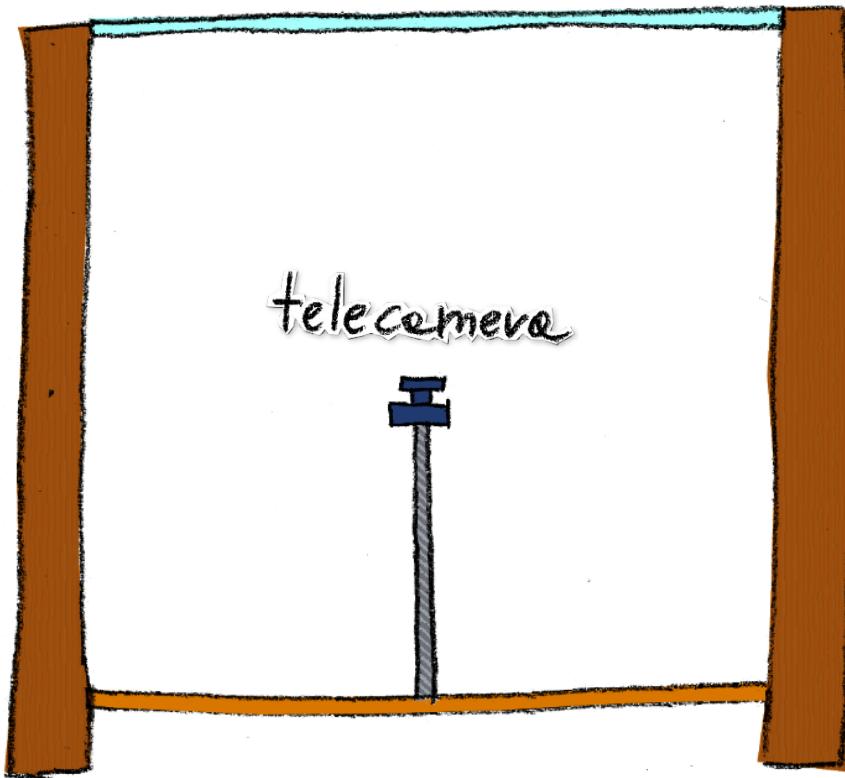
Componenti:

- ✗ Proiettore Acer s1210
- ✗ Lastra in Plexiglass PMMA
- ✗ Foglio di frost 129 heavy frost



# Architettura del tavolo

## Immagini nell'infrarosso



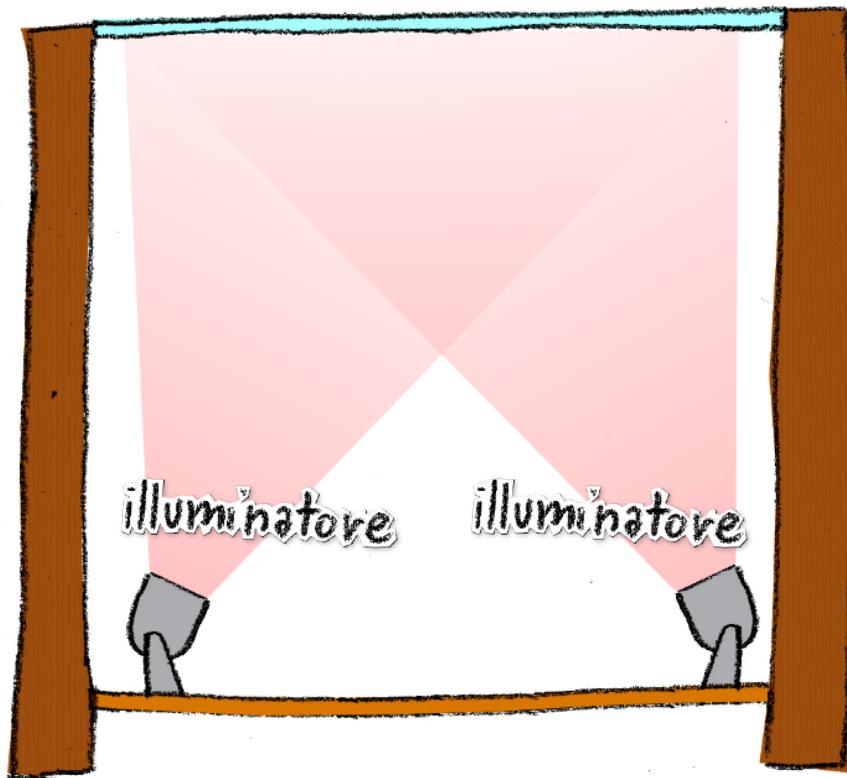
Ricezione dell'immagine

Componenti:

- × Videocamera PS3Eye

# Architettura del tavolo

## Immagini nell'infrarosso



Illuminazione nel range  
IR

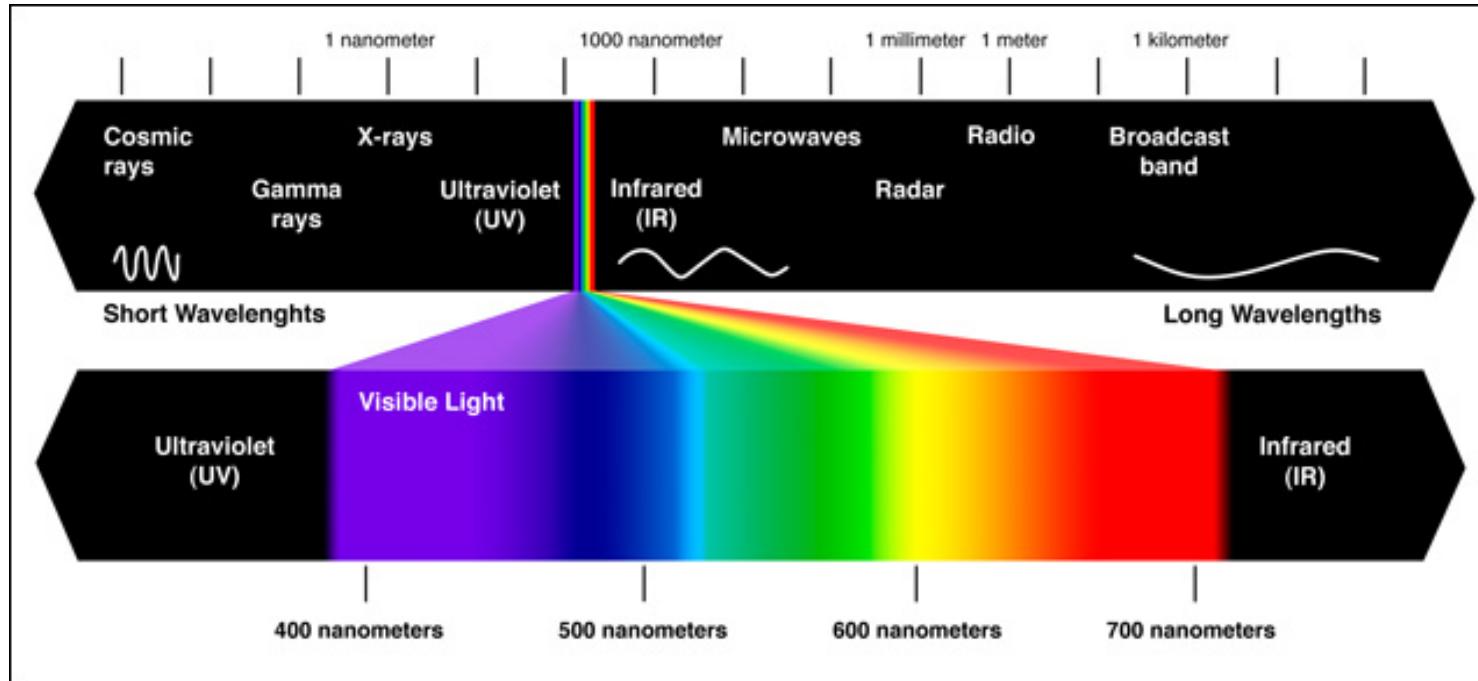
Componenti:

- × 4 illuminatori infrarossi



# Architettura del tavolo

## Perché l'illuminazione IR?



- ✗ Range dell'infrarosso (a partire da 850 nm circa)  
Ricezione dell'immagine relativa allo stato della superficie
- ✗ Range del visibile  
Display di un'interfaccia grafica coerente con lo stato della superficie

Approfondimenti: [lucidscience.com](http://lucidscience.com)

# Architettura del tavolo ps3Eye

- ✗ Lens mount
- ✗ Rimozione filtro IR
- ✗ Applicazione filtro visibile
- ✗ Lente grandangolare

- Vantaggi di usare la ps3Eye:
- ✗ facilità di rimozione del filtro IR
  - ✗ facilità di cambio del lens mount
  - ✗ prezzo contenuto
  - ✗ framerate elevato



# Architettura del tavolo

## Visione d'insieme delle componenti

Ps3Eye privata del filtro IR e dotata di filtro per il visibile per captare lo stato della superficie

Illuminatori per diffondere luce IR all'interno del tavolo

Proiettore Acer s1210 per restituire un'interfaccia grafica. Emette solo nel visibile.

Teli di tessuto nero oscurante ignifugo per evitare interferenze con la luce esterna al tavolo

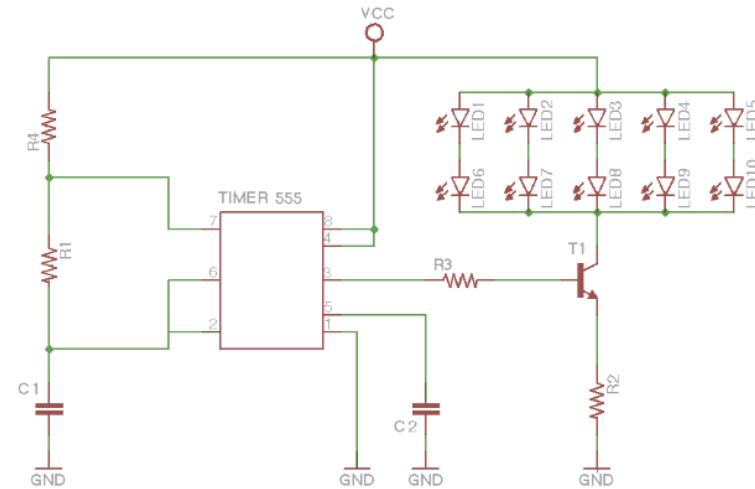
(la luce del sole o di una lampada contiene componenti nel range di frequenze degli IR!!!)

# Architettura del tavolo

## Tentativi vari di illuminazione IR

- ✗ Sincronizzazione di un circuito a LED IR con la telecamera

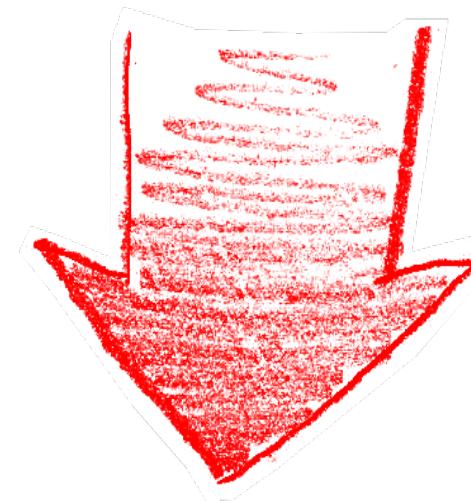
- ✗ Costruzione di un illuminatore IR con un timer 555



- ✗ Utilizzo di una telecamera incorporata con l'illuminatore (implementata nel prototipo2)

Range Visible

Range IR



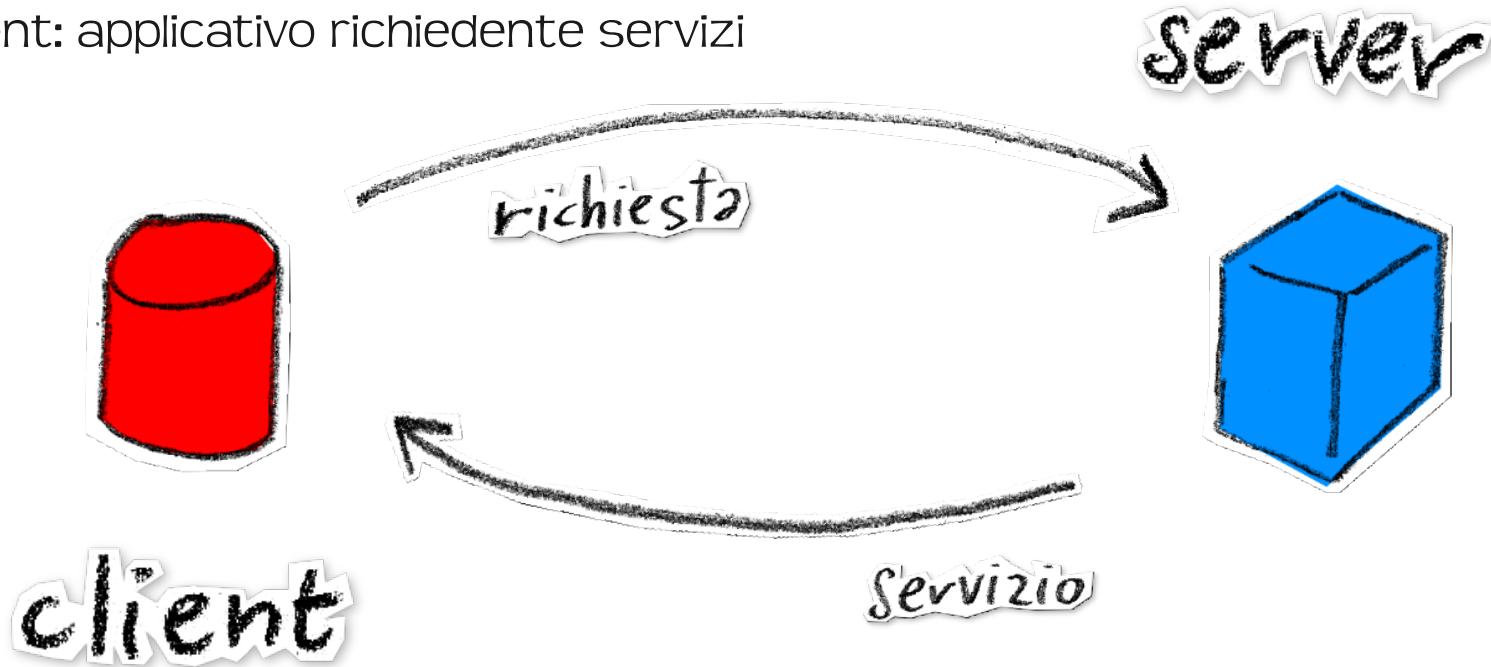
Client

Server

# Architettura software client - server

Server: gestore di servizi

Client: applicativo richiedente servizi



Il client richiede un servizio al server → Il server riconosce il client e lo eroga → il client invia al server la conferma di ricezione del servizio

# Architettura software protocolli e formattazione

## TCP (Transmission Control Protocol)

Per stabilire, mantenere e chiudere una connessione, è necessario inviare pacchetti di servizio

È affidabile ma lento

## UDP (User Datagram Protocol)

non gestisce il riordinamento dei pacchetti né la ritrasmissione di quelli persi

È molto rapido ma non sicuro

# Architettura software protocolli e formattazione

Applicazione	Protocollo strato applicazione	Protocollo strato trasporto
Posta elettronica	SMTP	TCP
Accesso a terminale remoto	telnet	TCP
Trasferimento file	FTP	TCP
Web	HTTP	TCP
Streaming Audio/Video	RTSP/RTP	TCP (comandi) + UDP (flusso)
Server di file remoto	NFS	tipicamente UDP
Telefonia su internet (VoIP)	SIP, H.323, altri	tipicamente UDP
Gestione della rete	SNMP	tipicamente UDP
Protocollo di routing	RIP	tipicamente UDP
Risoluzione dei nomi	DNS	tipicamente UDP

Fonte: wikipedia

# Architettura software

## OSC e TUIO

### **OSC (Open Sound Control)**

Formato di trasmissione di dati tramite invio di pacchetti di messaggi (tra computer, sintetizzatori, strumenti multimediali)

I messaggi sono generalmente inviati e ricevuti su internet, sottoreti o LAN (UDP/IP, Ethernet).

### **TUIO**

Protocollo di trasmissione dati su rete (locale[LAN] o intenet [Ethernet]).

Utilizza il protocollo di trasmissione UDP/IP.

Si basa sulla formattazione dei messaggi OSC, di cui rappresenta una particolarizzazione.

Si basa su architettura Client – Server

# Architettura software

## TUIO v1.1

Anatomia di un messaggio TUIO

[src]/[alive]/[set]/[fseq]

**Src** = sorgente TUIO

**Alive** = lista degli oggetti presenti sulla superficie

**Set** = informazioni sullo stato degli oggetti

**Fseq** = identifica un determinato stadio di update



Oggetto = un qualunque oggetto interagente con la superficie.

Esempi di oggetti: oggetto fisico, un dito, ...

# Architettura software

## il protocollo TUIO

Protocollo inventato da M. Kaltenbrunner, T. Bovermann, R. Bencina, E. Costanza

Ispirato dai lavori presso Universitat Pempeu Fabra (Barcellona) sul Reactable

Implementato quasi esclusivamente per applicativi basati sul riconoscimento di fiducial marker tramite computer vision.

TUIO è implementato in Reactivision, un applicativo che funge da server e che semplifica la generazione e l'invio di messaggi correttamente formattati.



# Architettura software

## I fiducial marker

Marker visivi univocamente riconoscibili da un sistema di computer vision

Evoluzione dei fiducial:

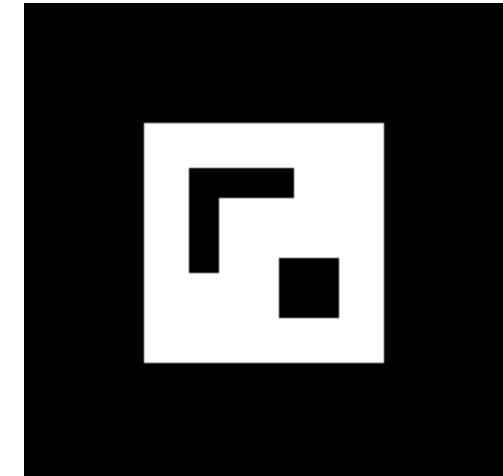
[ARToolKit](#)

[d-touch](#)

[Amoeba](#) (miglioramento del dtouch → algoritmo genetico)

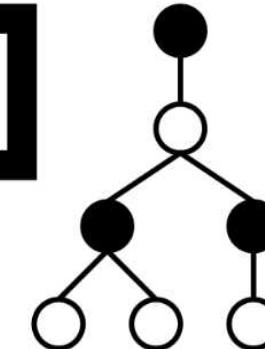
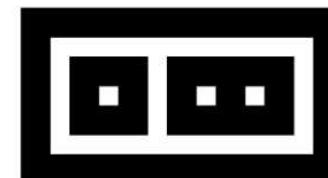
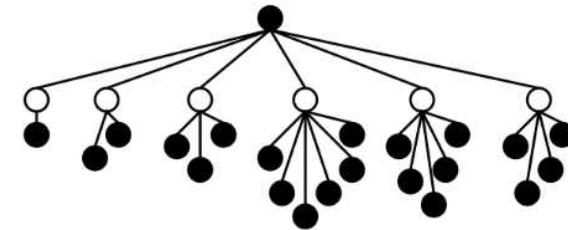
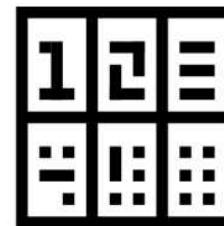
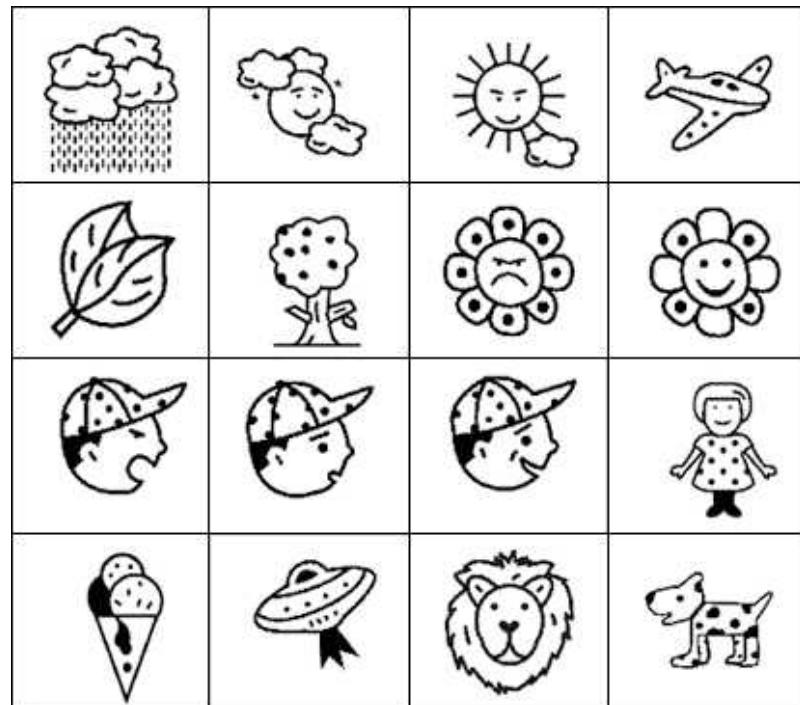
# Architettura software

## I fiducial marker: ARToolKit



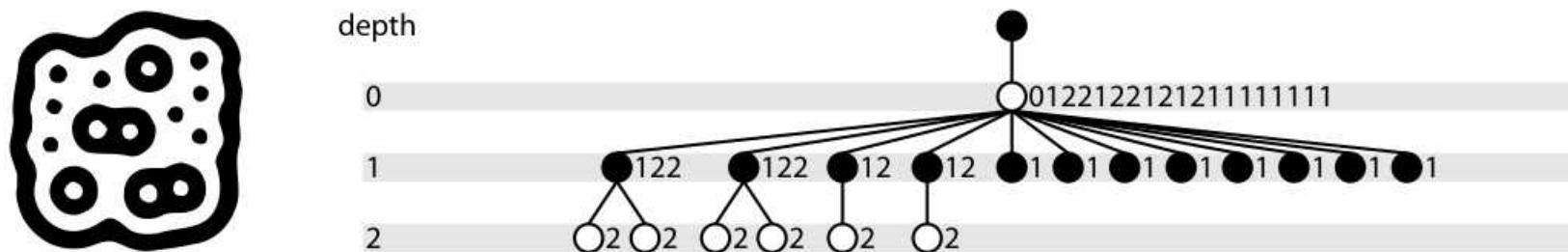
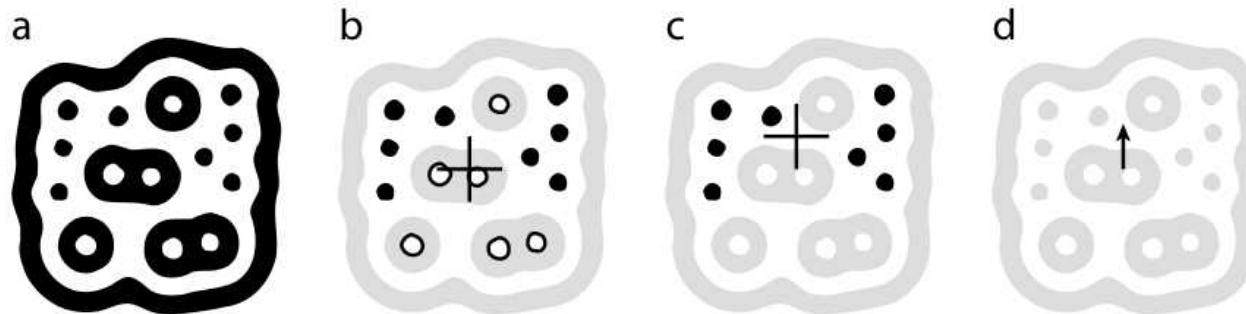
# Architettura software

I fiducial marker: d-touch



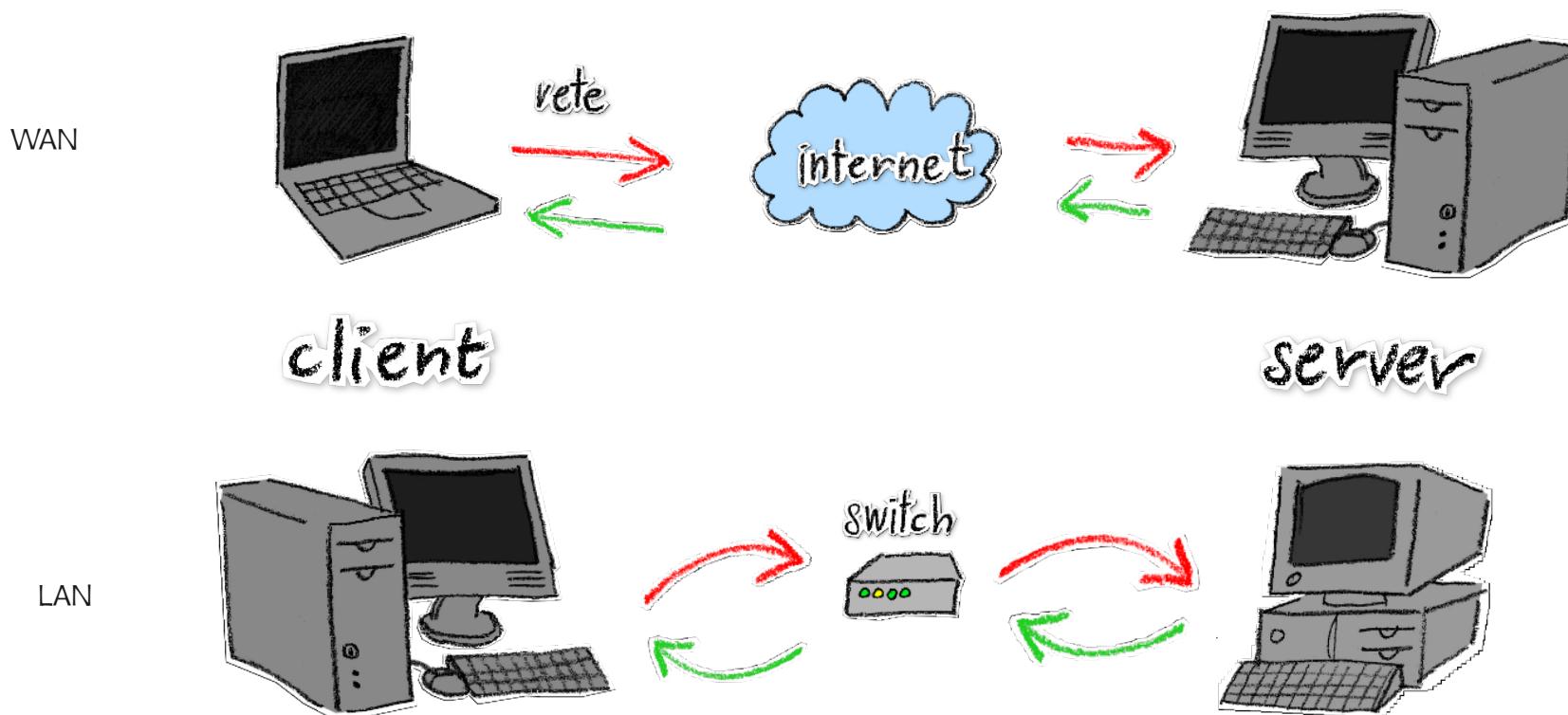
# Architettura software

I fiducial marker: amoeba



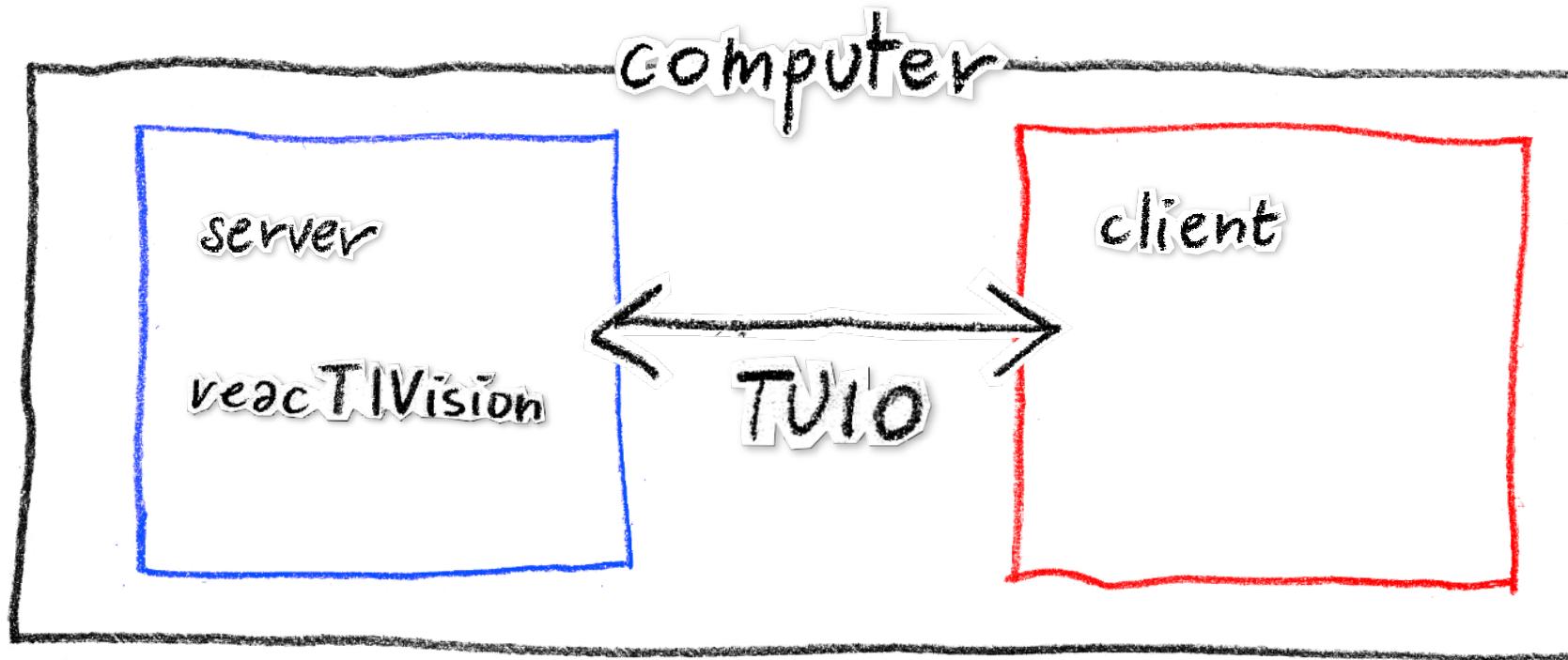
# Architettura software

## Il nostro setup



# Architettura software

## Il nostro setup



# Architettura software

## Il nostro client

Quali strumenti utilizzare?

✗ Processing



✗ Cinder

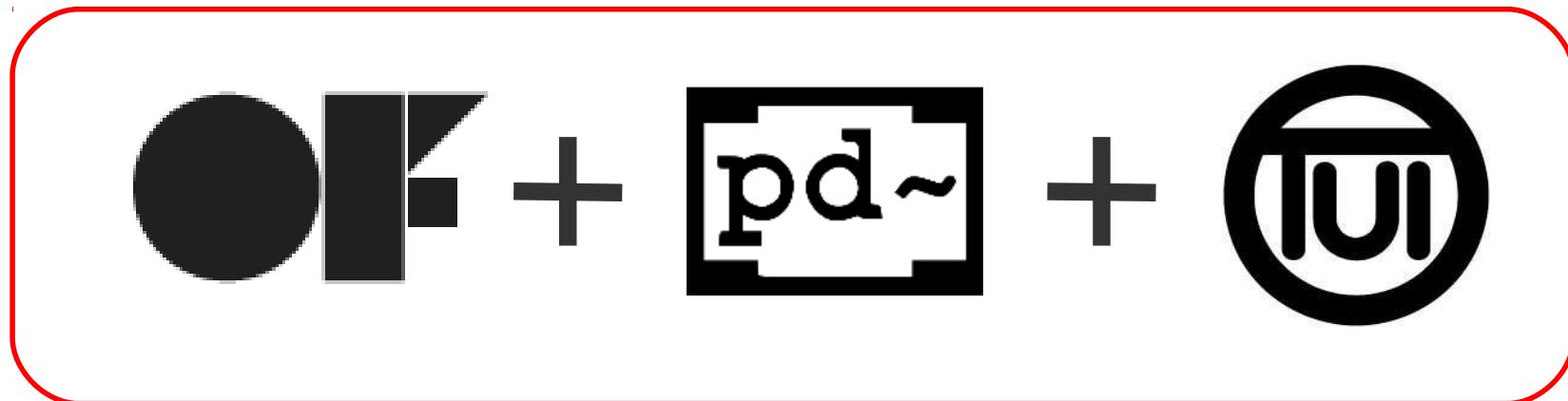


✗ openFrameworks



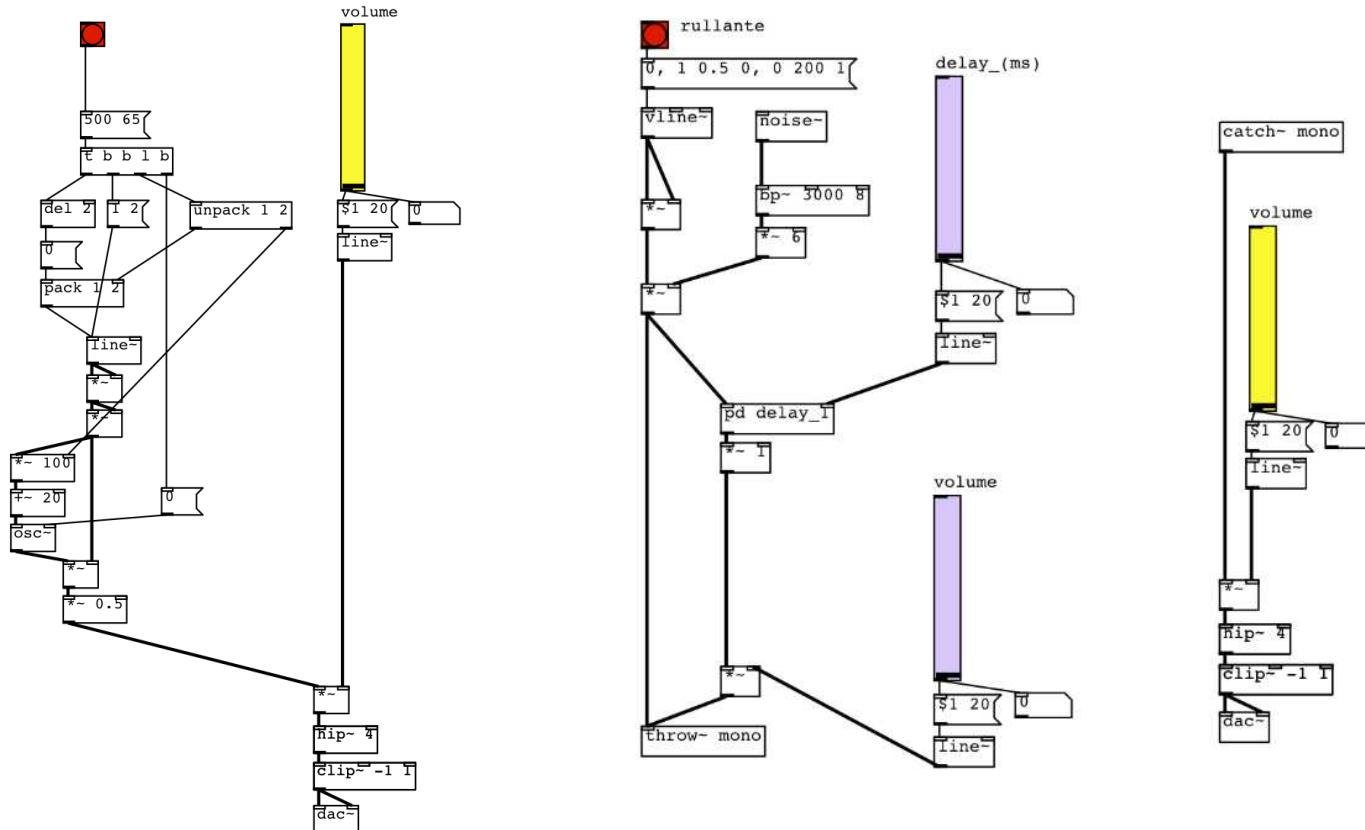
✗ PureData

✗ SuperCollider



# Architettura software

## Il nostro client - PureData



I suoni sono generati in tempo reale (audio procedurale)

# Architettura software

## Il nostro client - addons

Come mettere assieme openFrameworks, TUIO, PureData?

### OfAddons!!!

ofxPd: permette di utilizzare la libreria libpd che consente di embeddare il core di PureData in programmi sviluppati in C, C++, java,...

ofxTuio: permette di creare e gestire direttamente nell'ambiente di sviluppo di openFrameworks un server e un client che comunicano tramite protocollo TUIO

Libpd è sviluppato da [Peter Brinkmann](#)

ofxPd è sviluppato da [danomatika](#)

ofxTuio è sviluppato da [patriciogonzalezvivo](#)

# Architettura software

## Il nostro client - addons

Problemi	Soluzioni
Distorsione ottica della lente grandangolare	Ottimizzazione della posizione relativa della telecamera
Posizionamento angolato del proiettore	Keystone software
Distorsione sonora	Matrice di mappatura dei suoni
Riflessi degli illuminatori	Angolazione degli illuminatori e riposizionamento della telecamera
Surriscaldamento interno del tavolo	Applicazione di due ventola di raffreddamento

# Links

NUI group	Martin Kaltenbrunner	d-touch
Reactivision	Ross Bencina	Cinder
Reactable	OpenFrameworks	SuperCollider
TUIO	Processing	PureData
OSC	ARToolKit	LucidScience
Danomatika	patriciogonzalevivo	libpd

# Papers

- Kaltenbrunner, M., Bovermann, T., Bencina, R., Costanza, E.: "TUIO - A Protocol for Table-Top Tangible User Interfaces". Proceedings of the 6th International Workshop on Gesture in Human-Computer Interaction and Simulation (GW 2005), Vannes, France, 2005
- Kaltenbrunner, M., Bencina, R.: "reacTIVision: A Computer-Vision Framework for Table-Based Tangible Interaction". Proceedings of the first international conference on "Tangible and Embedded Interaction" (TEI07). Baton Rouge, Louisiana, 2007
- Wright, M., Freed, A., Momeni A.: "OpenSound Control: State of the Art 2003". Proceedings of the 3rd Conference on New Instruments for Musical Expression (NIME 03), Montreal, Canada, 2003.
- Kaltenbrunner, M. "reacTIVision and TUIO: A Tangible Tabletop Toolkit", Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces (ITS2009). Banff, Canada.
- Kaltenbrunner, M. & Bencina, R. "reacTIVision: A Computer-Vision Framework for Table-Based Tangible Interaction", Proceedings of the first international conference on "Tangible and Embedded Interaction (TEI07). Baton Rouge, Louisiana.
- Bencina, R. & Kaltenbrunner, M. "The Design and Evolution of Fiducials for the reacTIVision System", Proceedings of the 3rd International Conference on Generative Systems in the Electronic Arts (3rd Iteration 2005), Melbourne (Australia)

# Bibliografia e letture consigliate

- J. Kreidler, "loadbang", 2009, wolke
- A. Farnell, "Designing Sound", 2010, MIT Press
- Wilson, Cottle, Collins, "The SuperCollider Book", 2011, MIT Press
- H. Schildt, "The Complete Reference C++", 2003, McGraw Hill
- D. Shiffman, "Learning Processing", 2008, Morgan Kaufmann
- J. Noble, "Interactivity", 2009, O'Reilly