



INTERFACES MULTIPONTES

Géry Casiez
M2 IVI - NIHM

<http://cristal.univ-lille.fr/~casiez>

CS Education Week

**Recognizing the
Transformative Role
of Computing**

Certified Software

A Conversation with
Pixar's Ed Catmull

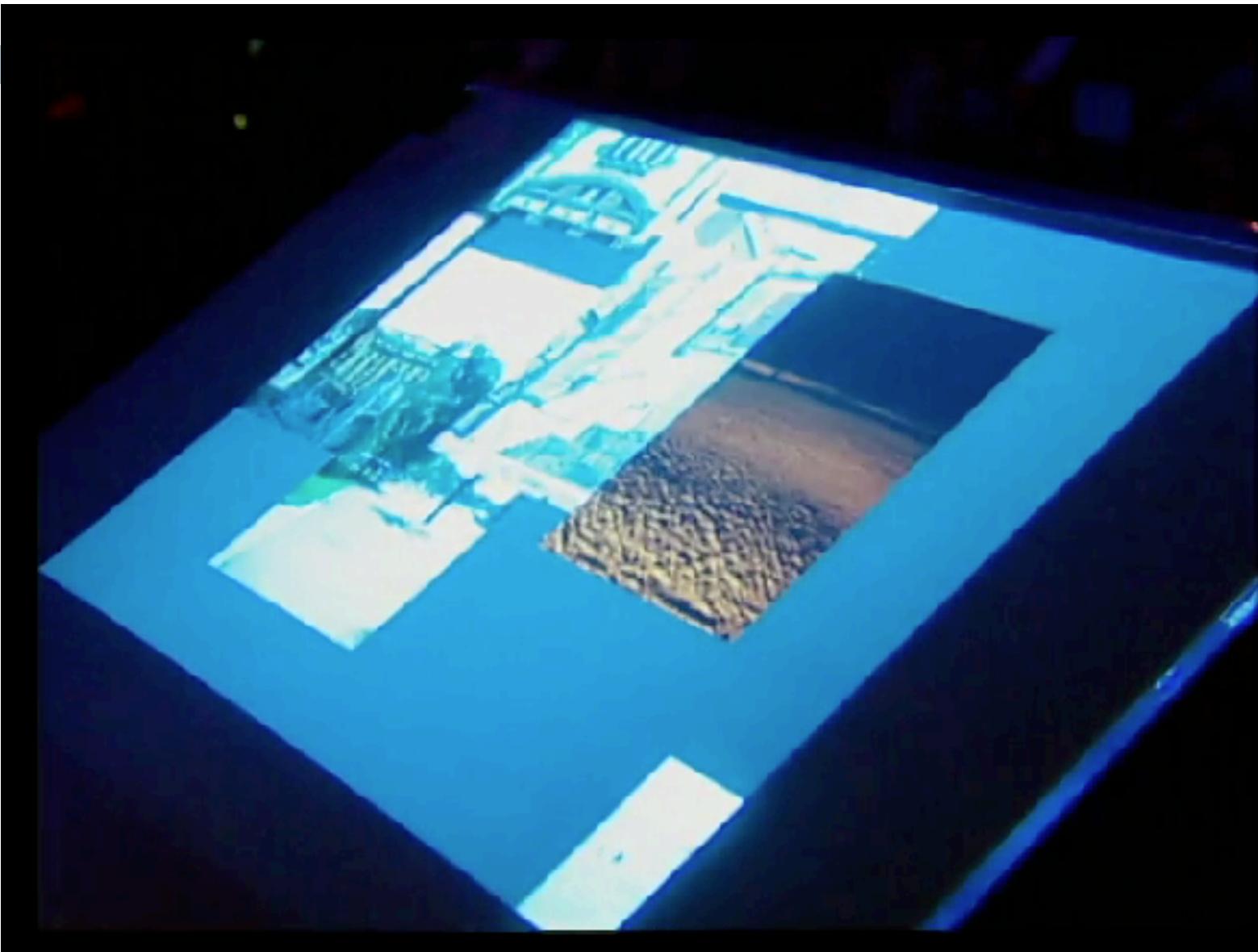
Bayesian Networks

The Theft of
Business Innovation



Jeff Han 2006

3



RST [Krueger et al. 1985]

4



Technologie

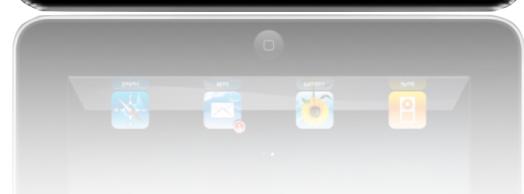
5



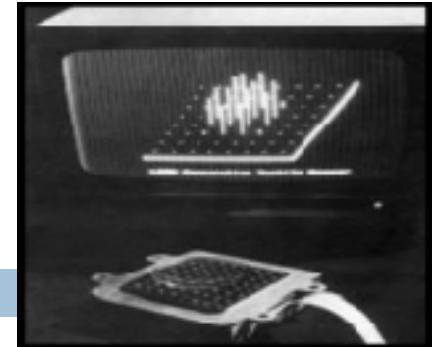
1972
PlatoIV



2007
iPhone



2010
iPad



A MULTI-TOUCH THREE DIMENSIONAL TOUCH-SENSITIVE TABLET

S.K. Lee, W. Buxton, K.C. Smith
Computer Systems Research Institute
University of Toronto
Toronto, Ontario
Canada, M5S 1A4

(416)-978-6320

ABSTRACT

A prototype touch-sensitive tablet is presented. The tablet's main innovation is that it is capable of sensing more than one point of contact at a time. In addition to being able to provide position coordinates, the tablet also gives a measure of degree of contact, independently for each point of contact. In order to enable multi-touch sensing, the tablet surface is divided into a grid of discrete points. The points are scanned using a recursive area subdivision algorithm. In order to minimize the resolution lost due to the discrete nature of the grid, a novel interpolation scheme has been developed. Finally, the paper briefly discusses how multi-touch sensing, interpolation, and degree of contact sensing can be combined to expand our vocabulary in human-computer interaction.

1. INTRODUCTION

Rapid advancement of computer technology has opened a variety of new applications. New applications and users mean demands for new modes of interaction. One consequence of this is a growing appreciation of the importance of using appropriate input technologies (Buxton, 1982). Positioning devices are seen to be essential to graphics applications, image transducers are required for pattern

recognition, and so on. In this paper we present a new type of input device which is a three dimensional touch-sensitive tablet. This tablet combines two additional features. First, it can sense the degree of contact in a continuous manner. Second, it can sense the amount and location of a number of simultaneous points of contact. These two features, when combined with touch sensing, are very important in respect to the types of interaction that we can support. Some of these are discussed below, but see Buxton, Hill, and Rowley (1985) and Brown, Buxton and Murtagh (1985) for more detail. The tablet which we present is a continuation of work done in our lab by Sasaki et al (1981) and Metha (1982).

In the presentation which follows, we focus mainly on issues relating to the transducer's implementation. Two important contributions discussed are our method of scanning the tablet surface, and our method of maintaining high resolution despite the surface being partitioned into a discrete grid. Additional technical details can be found in Lee (1984).

2. WHY MULTI-TOUCH?

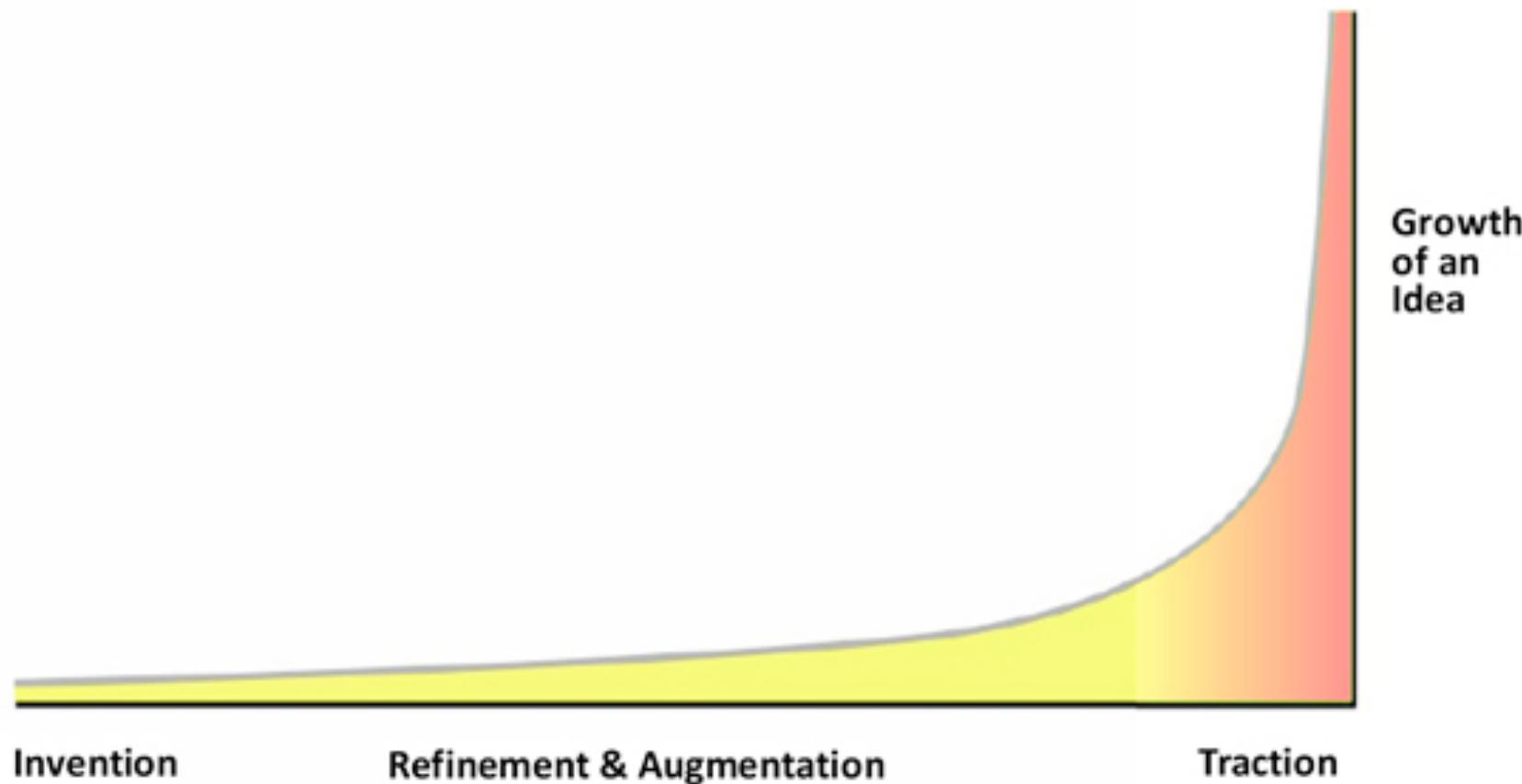
Touch sensing has a number of important characteristics. There is no physical stylus or puck to get lost, broken, or vibrate out of position. Touch tablets can be molded so as to make them easy to clean (therefore making them useful in clean environments like hospitals, or dirty environments like factories). Since there is no



The long nose of innovation

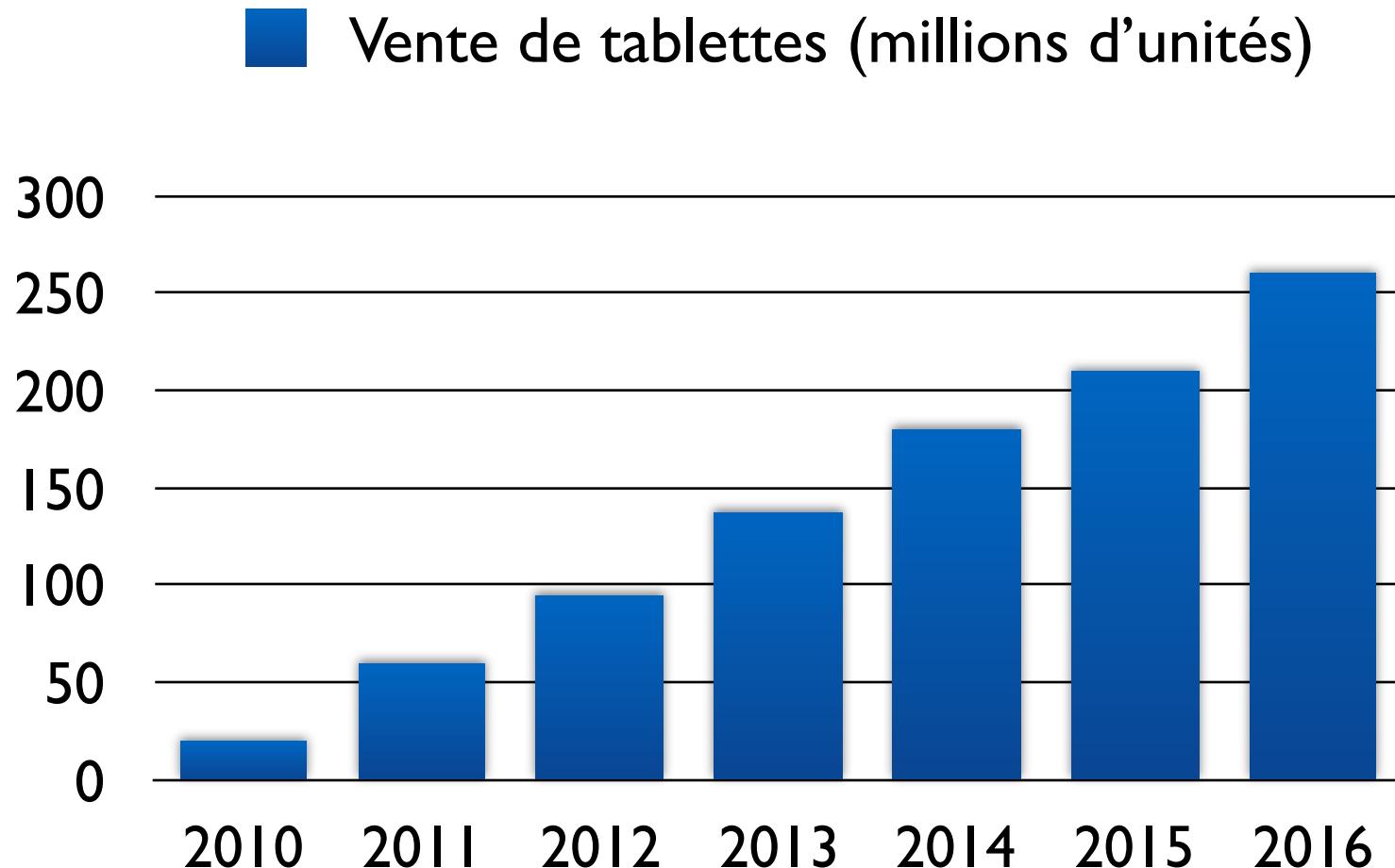
8

□ Bill Buxton



Technologie

9



Source : Rapport DisplaySearch 2011

Technologie tactile

10

- Nombre de périphériques tactiles dans les espaces publics



2007

2008

2009

2010

2011

2012

- “*Everything is best for something and worst for something else. The trick is knowing what is what, for what, when, for whom, where, and most importantly, why.*” Bill Buxton

Avantages

12



Barrière d'entrée



Robuste



Plaisir



Sans fil

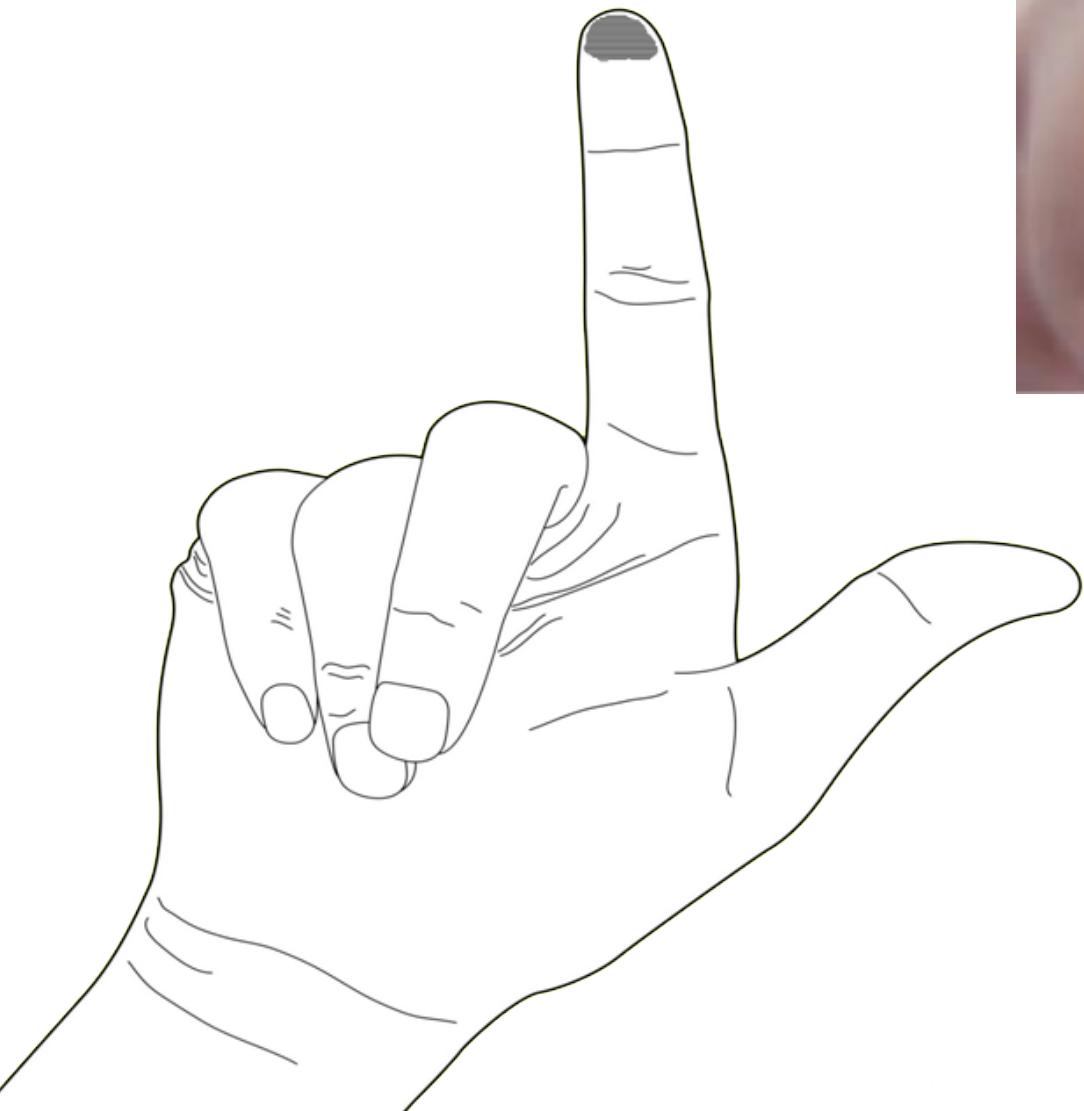
Fat finger problem

13



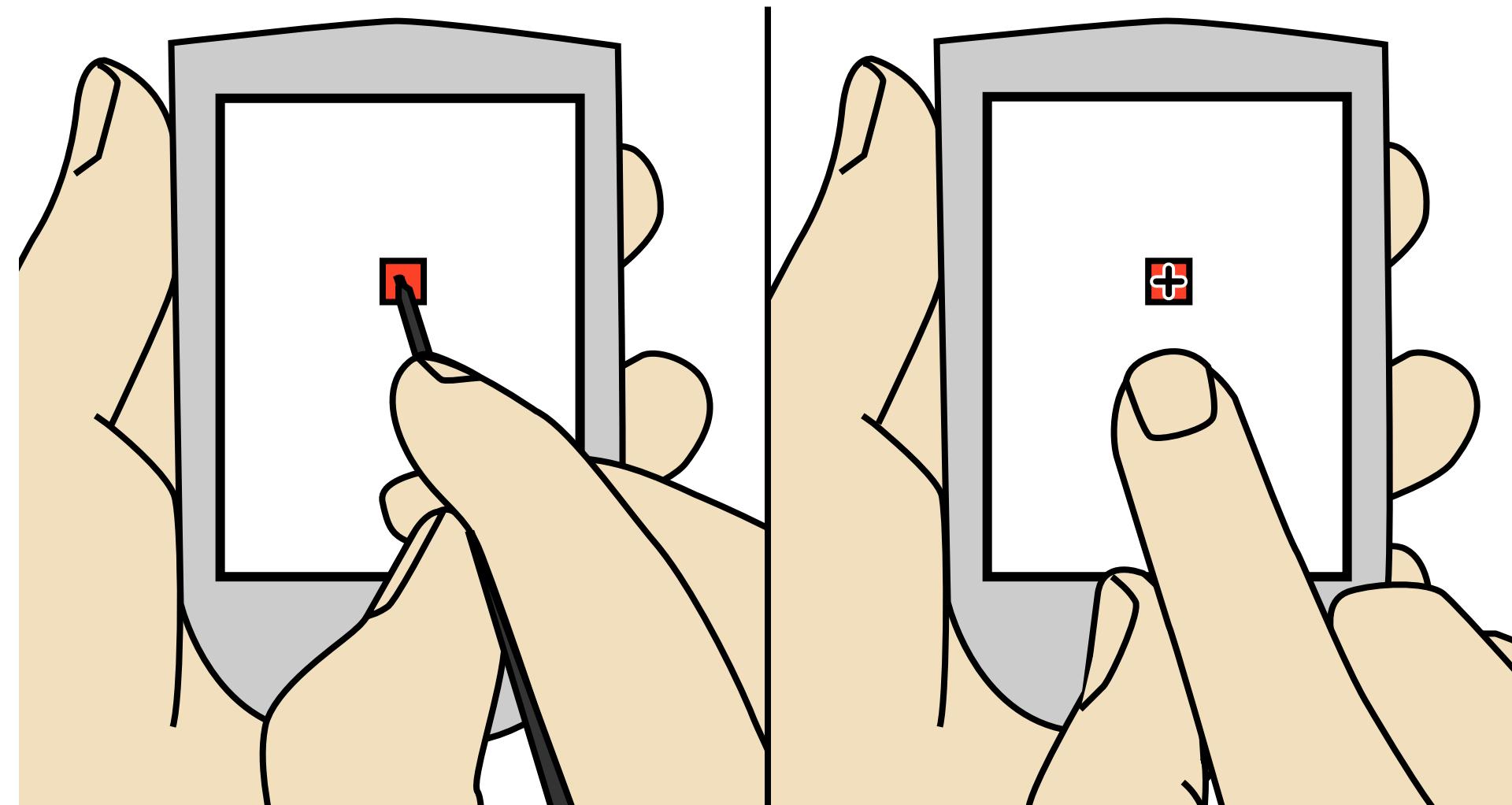
Fat finger problem

14



Precision

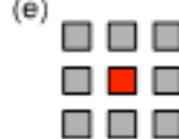
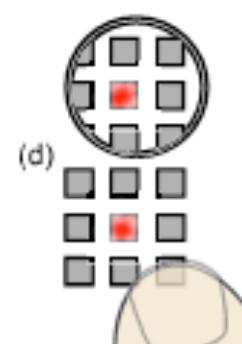
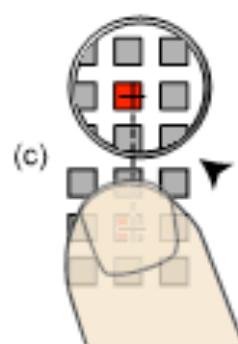
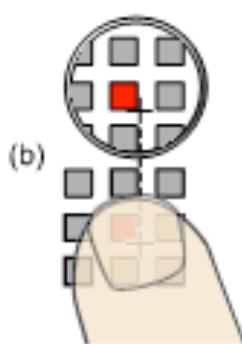
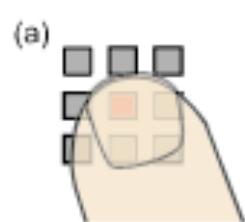
15



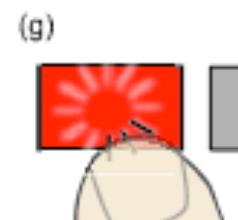
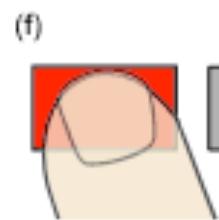
Shift [Vogel et al. 2007]

16

scenario 1:
ambiguous target
due to occlusion



scenario 2:
occlusion not a
problem



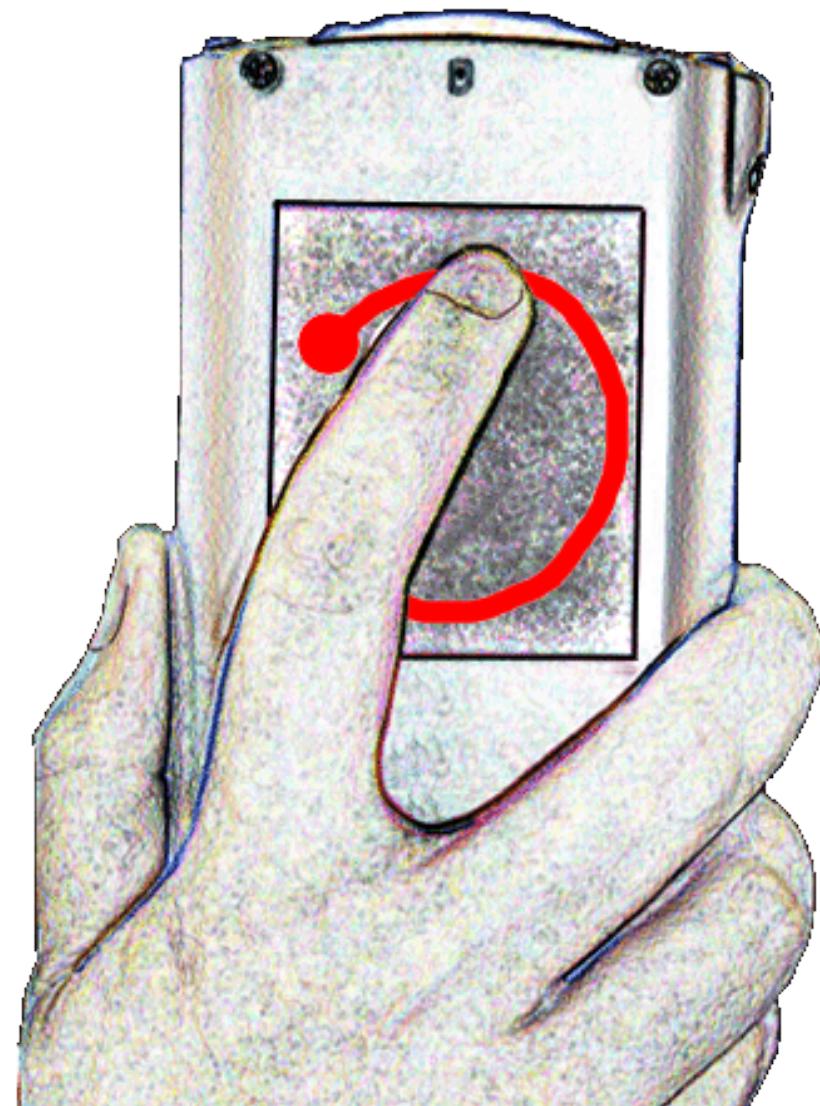
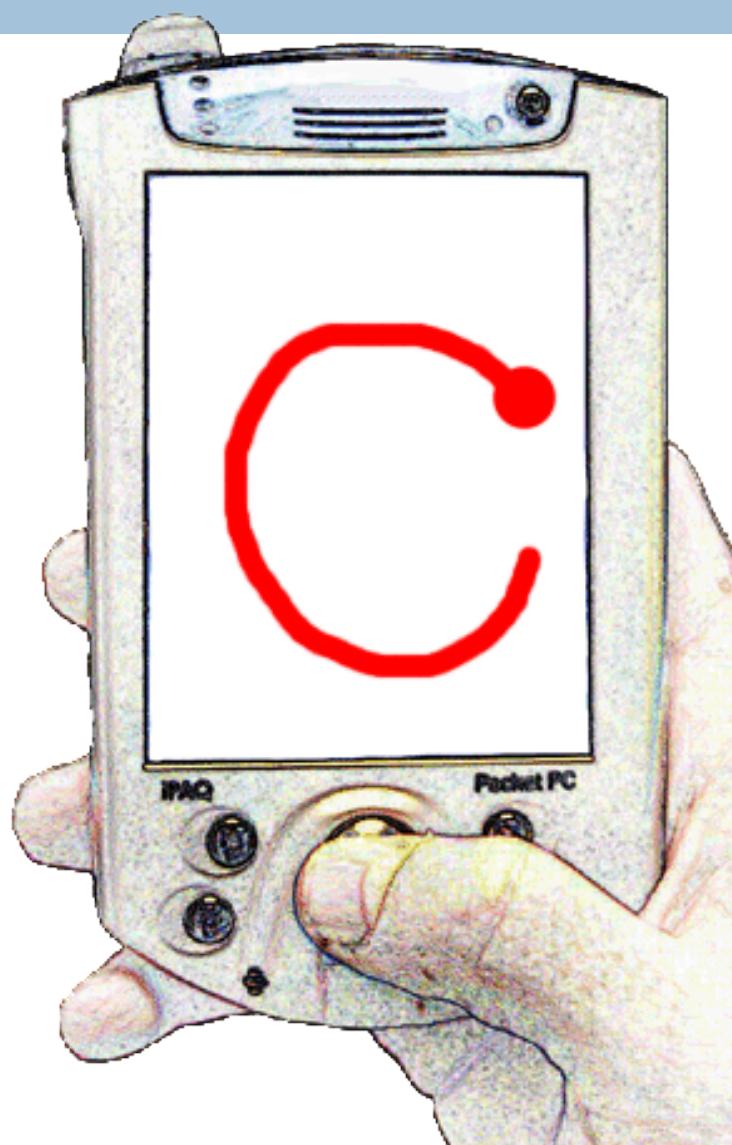
Shift

17



Behind touch [Wobbrock et al.]

18



Occultation

19

Occlusion-Aware Interfaces

Daniel Vogel^{1,2} and Ravin Balakrishnan¹

¹Dept. of Computer Science
University of Toronto, CANADA

²Dept. of Math & Computer Science
Mount Allison University, CANADA

Multipoint vs souris

20



Multipoint vs souris

21

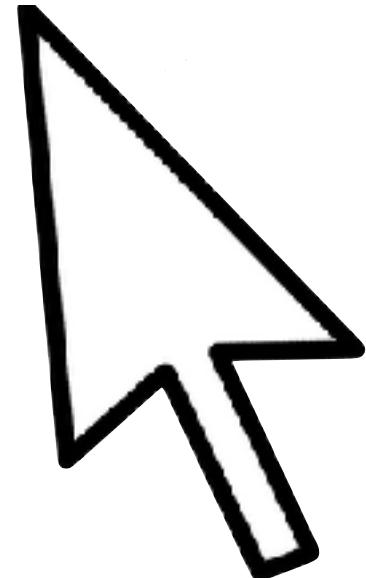


(2) Degrés de Liberté (DDL) intégrés

(1) pointeur

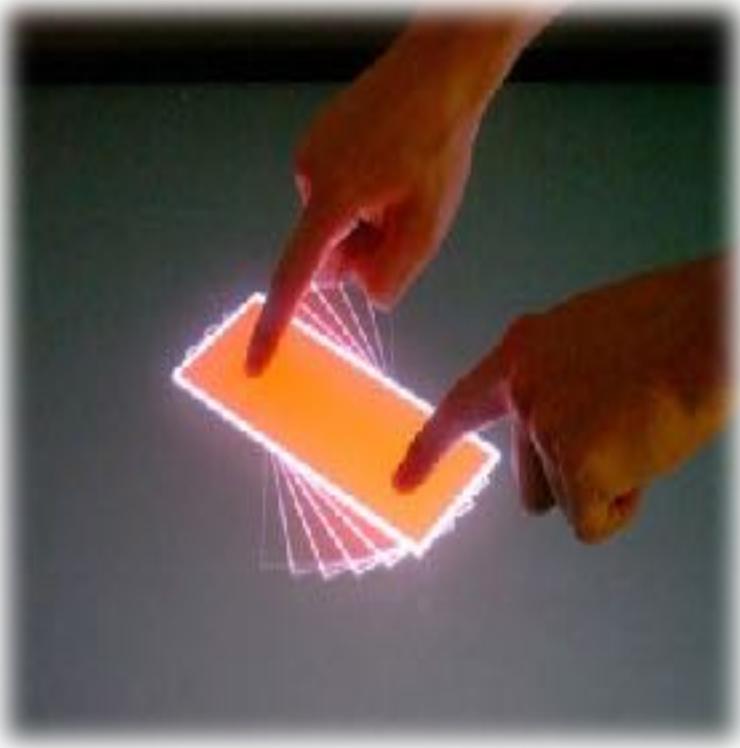
interaction *indirecte*

+ 1 DDL séparé



Directivité

22



Direct

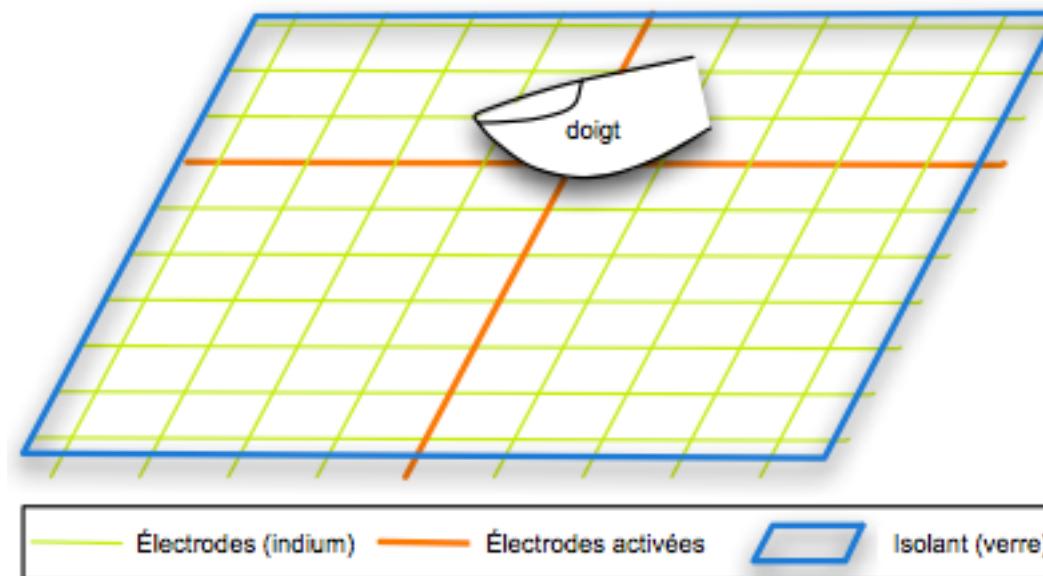


Indirect

Technologies

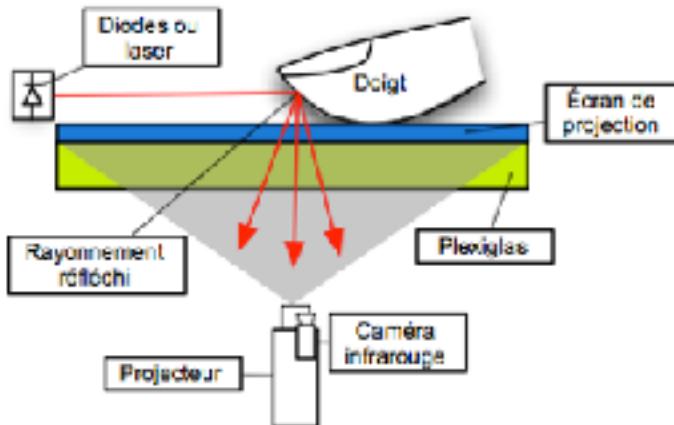
23

- Acoustique
- Résistive
- Capacitive
- Optique

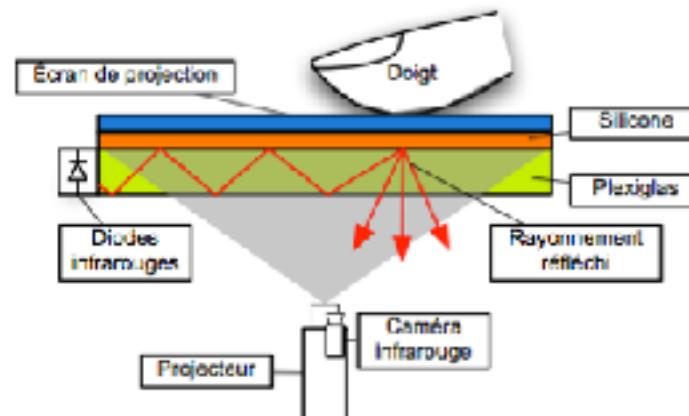


Technologie optique

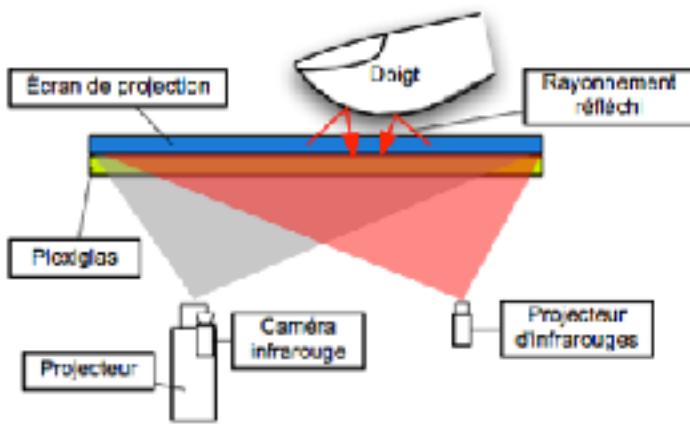
24



(a) *Occlusion Sensing.*



(b) *FTIR.*



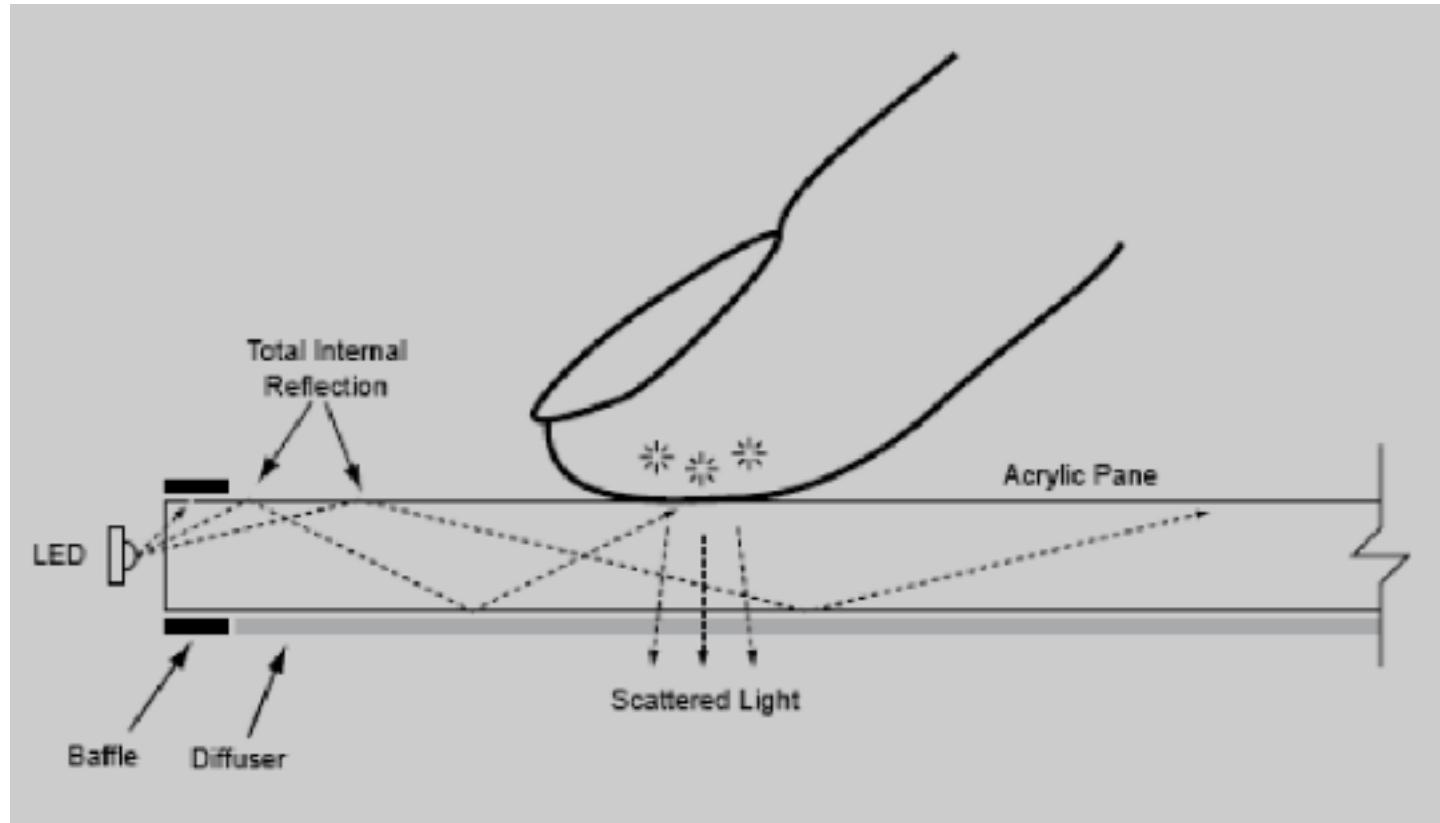
(c) *DI.*



(d) *ThinShight.*

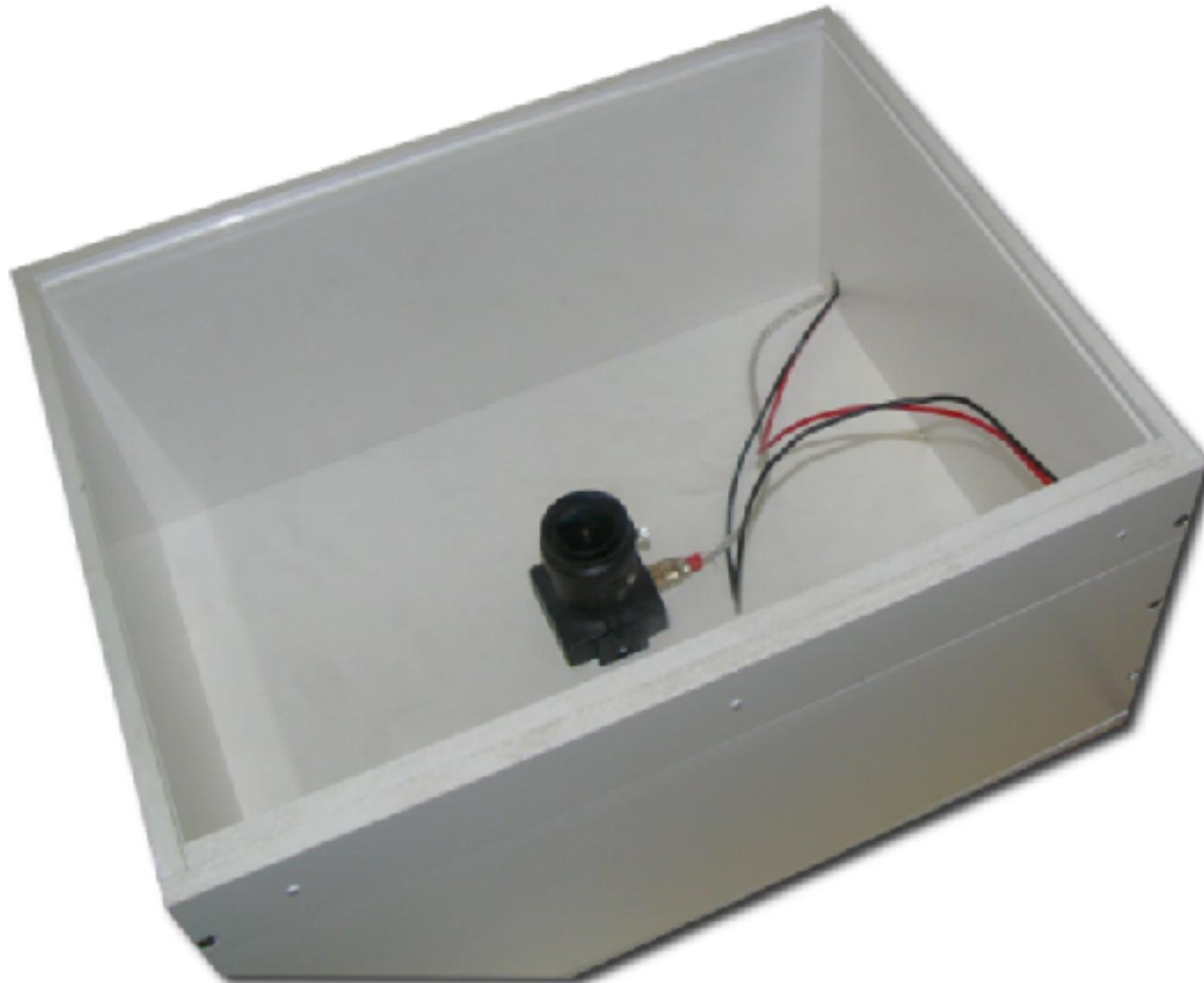
FTIR

25



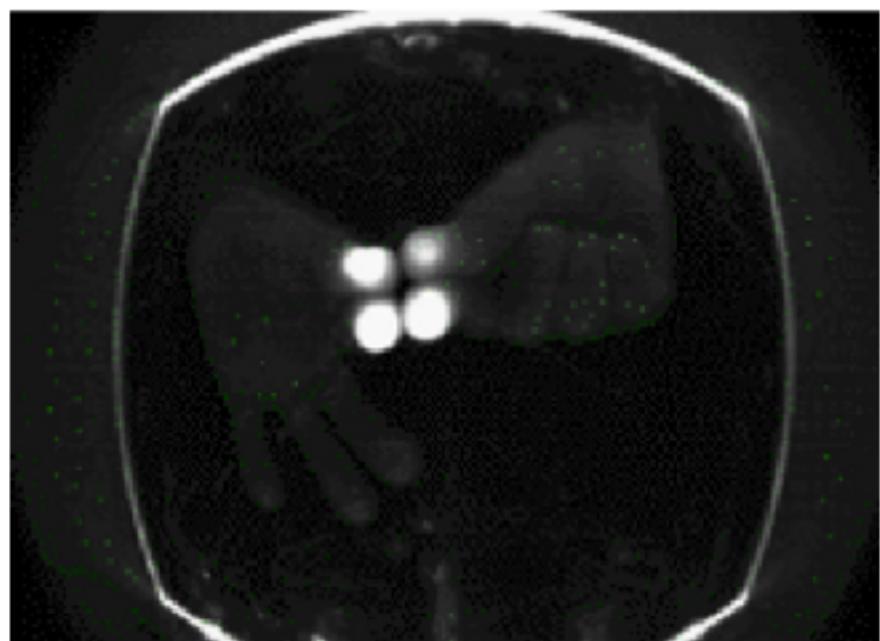
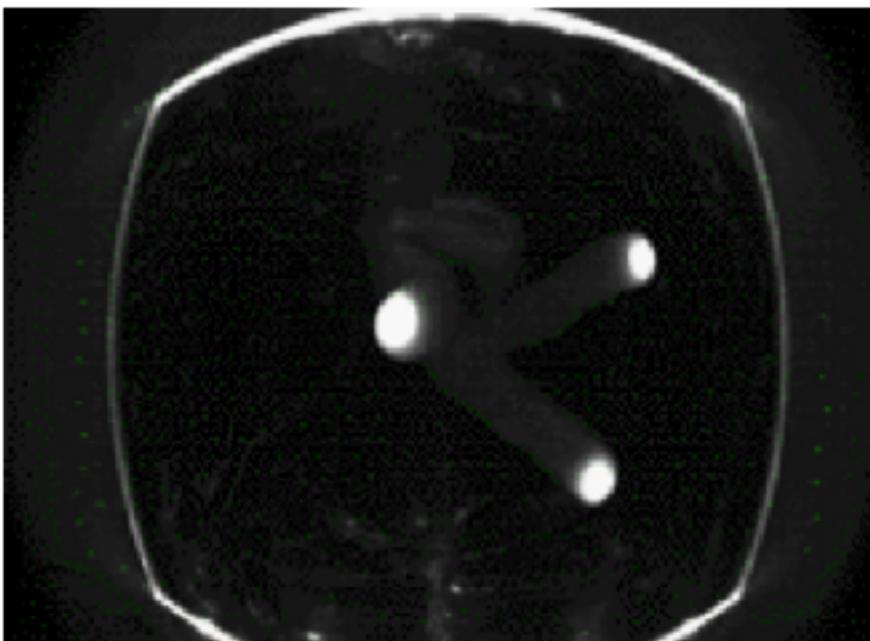
FTIR

26



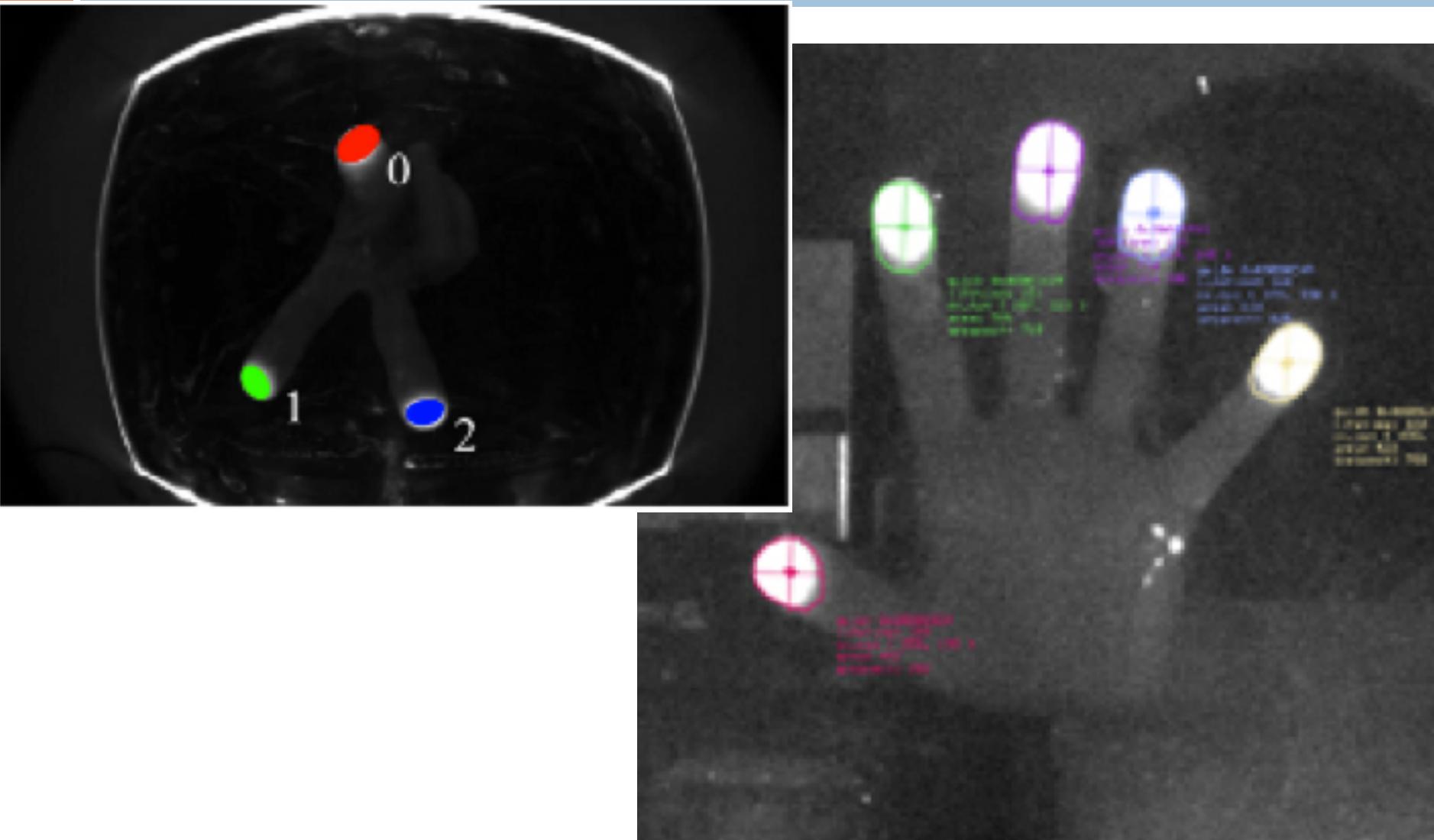
FTIR

27



FTIR

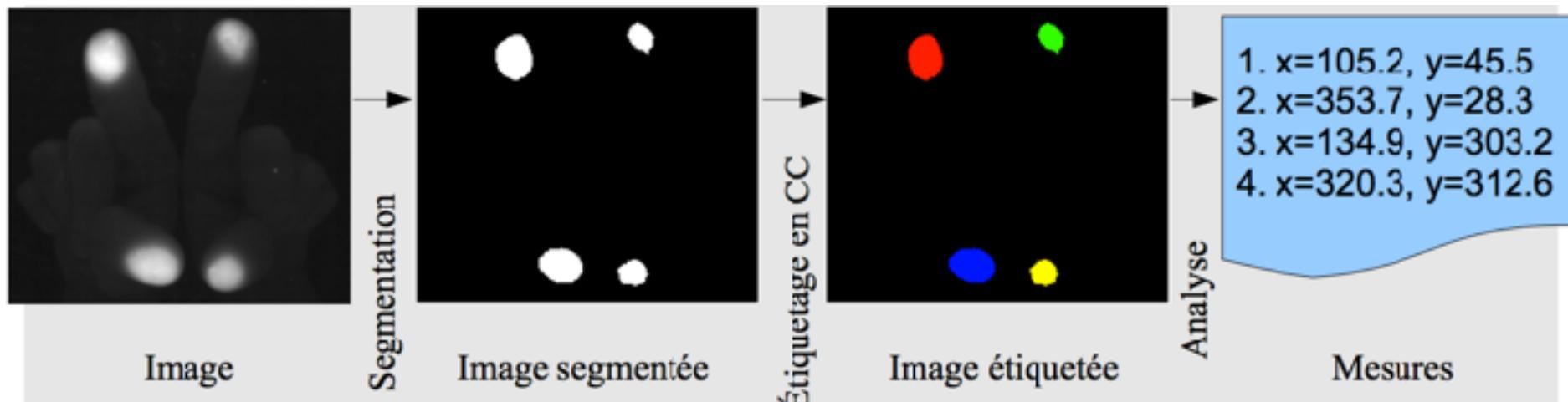
28



Segmentation

29

□ Binarisation

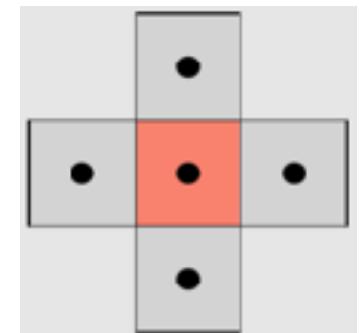


Détection des régions connexes

30

0	0	0	0	0	0	0	0	0
0	1	1	0	0	0	1	1	1
0	1	1	0	0	1	0	1	1
0	1	1	0	0	0	0	0	0
0	0	0	1	1	0	0	0	0
0	0	0	1	1	0	0	0	0
0	0	0	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0

0	0	0	0	0	0	0	0	0
0	1	1	0	0	0	1	1	1
0	1	1	0	0	1	0	1	1
0	1	1	0	0	0	0	0	0
0	0	0	1	1	0	0	0	0
0	0	0	1	1	0	0	0	0
0	0	0	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0

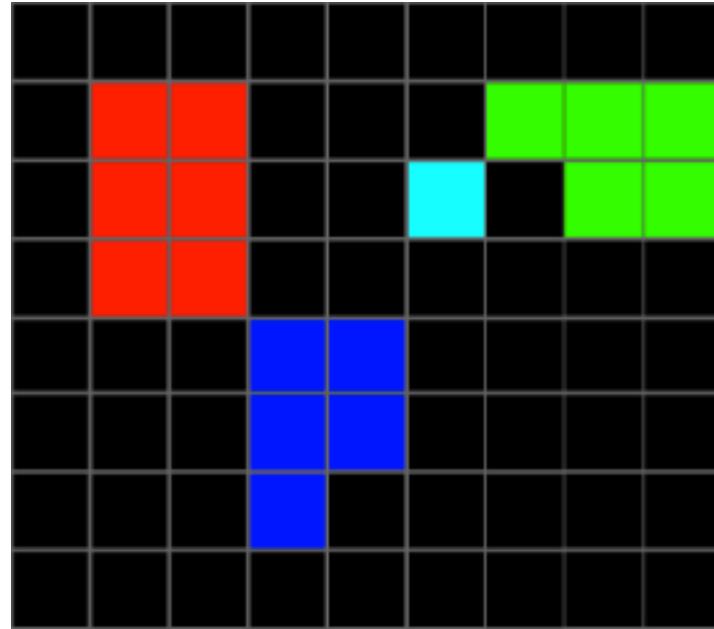


Etiquetage

31

0	0	0	0	0	0	0	0	0
0	1	1	0	0	0	1	1	1
0	1	1	0	0	1	0	1	1
0	1	1	0	0	0	0	0	0
0	0	0	1	1	0	0	0	0
0	0	0	1	1	0	0	0	0
0	0	0	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0

0	0	0	0	0	0	0	0	0
0	1	1	0	0	0	2	2	2
0	1	1	0	0	3	0	2	2
0	1	1	0	0	0	0	0	0
0	0	0	4	4	0	0	0	0
0	0	0	4	4	0	0	0	0
0	0	0	4	0	0	0	0	0
0	0	0	0	0	0	0	0	0



Double parcours

32

→ E/S

- Paramètre d'entrée : image binaire B
- Résultat de sortie : matrice (ou *carte*) d'étiquettes L

→ Algorithme

- Premier parcours de l'image, dans le sens classique (*raster-scan order*) :
À chaque pixel à 1 dans B , on affecte
 - la plus petite étiquette parmi celles de ses voisins **haut** et **gauche**
ou
 - une nouvelle étiquette si aucun de ces 2 voisins n'est encore étiqueté
- Second parcours de l'image, dans le sens inverse :
À chaque pixel précédemment étiqueté, on affecte la plus petite étiquette parmi la **sienne** et celles de ses voisins **bas** et **droite**.

Double parcours (1er parcours)

33

Exemple-1^{er} parcours (1/3)

- Initialisation
 $nbLabels=0$
 - Étape 1
les voisins haut et gauche du premier pixel à 1 dans B ne sont pas encore étiquetés
⇒ nouvelle étiquette ($nbLabels=1$)
 - Étape 2
le voisin gauche du pixel suivant à 1 dans B est déjà étiqueté à 1 ⇒ affecter cette même étiquette au pixel
 - Étape 3
nouvelle étiquette ($nbLabels=2$)

D	0	0	0	0	0	0	0
D	0	1	1	0	1		1
D	1	1	0	0	0		1
D	1	0	0	1	1		1
D	0	0	0	0	0		0

0	0	0	0	0	0	0
0	0	1	1	0	1	1
0	1	1	0	0	0	1
0	1	0	0	1	1	1
0	0	0	0	0	0	0

0	0	0	0	0	0	0
0	0	1	1	0	1	1
0	1	1	0	0	0	1
0	1	0	0	1	1	1
0	0	0	0	0	0	0

0	0	0	0	0	0	0
0	0	1	1	0	1	1
0	1	1	0	0	0	1
0	1	0	0	1	1	1
0	0	0	0	0	0	0

$$L = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

0	0	0	0	0	0	0
0	0	1	1	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

0	0	0	0	0	0	0
0	0	1	1	0	2	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

Double parcours (1er parcours)

34

Exemple-1^{er} parcours (2/3)

- Étape 4
affecter au pixel l'étiquette de son voisin gauche
 - Étape 5
nouvelle étiquette ($nbLabels=3$)
 - Étape 6
les voisins haut et gauche du pixel suivant à 1 dans B sont étiquetés différemment \Rightarrow affecter au pixel l'étiquette minimale
 - Étape 7
affecter au pixel l'étiquette de son voisin haut

	0	0	0	0	0	0	0
	0	0	1	1	0	1	1
	0	1	1	0	0	0	1
	0	1	0	0	1	1	1
	0	0	0	0	0	0	0

0	0	0	0	0	0	0
0	0	1	1	0	1	1
0	1	1	0	0	0	1
0	1	0	0	1	1	1
0	0	0	0	0	0	0

0	0	0	0	0	0	0
0	0	1	1	0	1	1
0	1	1	0	0	0	1
0	1	0	0	1	1	1
0	0	0	0	0	0	0

0	0	0	0	0	0	0
0	0	1	1	0	1	1
0	1	1	0	0	0	1
0	1	0	0	1	1	1
0	0	0	0	0	0	0

L	0	0	0	0	0	0	0
	0	0	1	1	0	2	2
	0	0	0	0	0	0	0
	0	0	0	0	0	0	0
	0	0	0	0	0	0	0

0	0	0	0	0	0	0
0	0	1	1	0	2	2
0	3	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

0	0	0	0	0	0	0
0	0	1	1	0	2	2
0	3	1	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

0	0	0	0	0	0	0
0	0	1	1	0	2	2
0	3	1	0	0	0	2
0	0	0	0	0	0	0
0	0	0	0	0	0	0

Double parcours (1er parcours)

Exemple-1^{er} parcours (3/3)

- Étape 8
affecter au pixel l'étiquette de son voisin haut
- Étape 9
nouvelle étiquette ($nbLabels=4$)
- Étape 10
affecter au pixel l'étiquette de son voisin gauche
- Étape 11
les voisins haut et gauche du pixel suivant à 1 dans B sont étiquetés différemment \Rightarrow affecter au pixel l'étiquette minimale

 B

0	0	0	0	0	0	0
0	0	1	1	0	1	1
0	1	1	0	0	0	1
0	1	0	0	1	1	1
0	0	0	0	0	0	0

 L

0	0	0	0	0	0	0
0	0	1	1	0	2	2
0	3	1	0	0	0	2
0	3	0	0	0	0	0
0	0	0	0	0	0	0

0	0	0	0	0	0	0
0	0	1	1	0	1	1
0	1	1	0	0	0	1
0	1	0	0	1	1	1
0	0	0	0	0	0	0

0	0	0	0	0	0	0
0	0	1	1	0	2	2
0	3	1	0	0	0	2
0	3	0	0	4	0	0
0	0	0	0	0	0	0

0	0	0	0	0	0	0
0	0	1	1	0	1	1
0	1	1	0	0	0	1
0	1	0	0	1	1	1
0	0	0	0	0	0	0

0	0	0	0	0	0	0
0	0	1	1	0	2	2
0	3	1	0	0	0	2
0	3	0	0	4	4	0
0	0	0	0	0	0	0

0	0	0	0	0	0	0
0	0	1	1	0	1	1
0	1	1	0	0	0	1
0	1	0	0	1	1	1
0	0	0	0	0	0	0

0	0	0	0	0	0	0
0	0	1	1	0	2	2
0	3	1	0	0	0	2
0	3	0	0	4	4	2
0	0	0	0	0	0	0

Double parcours (2e parcours)

36

Exemple-2^{ème} parcours (1/2) B

- Image et carte initiales

0	0	0	0	0	0	0
0	0	1	1	0	1	1
0	1	1	0	0	0	1
0	1	0	0	1	1	1
0	0	0	0	0	0	0

L

0	0	0	0	0	0	0	0
0	0	1	1	0	2	2	2
0	3	1	0	0	0	2	2
0	3	0	0	4	4	2	2
0	0	0	0	0	0	0	0

- 1^{ère} étiquette modifiée

Affecter au pixel l'étiquette de son voisin droit car elle est inférieure à la sienne

0	0	0	0	0	0	0
0	0	1	1	0	1	1
0	1	1	0	0	0	1
0	1	0	0	1	1	1
0	0	0	0	0	0	0

0	0	0	0	0	0	0	0
0	0	1	1	0	2	2	2
0	3	1	0	0	0	2	2
0	3	0	0	4	2	2	2
0	0	0	0	0	0	0	0

- 2^{ème} étiquette modifiée

Affecter au pixel l'étiquette de son voisin droit car elle est inférieure à la sienne

0	0	0	0	0	0	0
0	0	1	1	0	1	1
0	1	1	0	0	0	1
0	1	0	0	1	1	1
0	0	0	0	0	0	0

0	0	0	0	0	0	0	0
0	0	1	1	0	2	2	2
0	3	1	0	0	0	2	2
0	3	0	0	2	2	2	2
0	0	0	0	0	0	0	0

- (étiquette non modifiée)

0	0	0	0	0	0	0
0	0	1	1	0	1	1
0	1	1	0	0	0	1
0	1	0	0	1	1	1
0	0	0	0	0	0	0

0	0	0	0	0	0	0	0
0	0	1	1	0	2	2	2
0	3	1	0	0	0	2	2
0	3	0	0	2	2	2	2
0	0	0	0	0	0	0	0

Double parcours (2e parcours)

37

Exemple-2^{ème} parcours (2/2)

- #### ■ (étiquette non modifiée)

■ 3^{ème} étiquette modifiée

Affecter au pixel l'étiquette de son voisin droit car elle est inférieure à la sienne et à celle de son voisin bas

- #### ■ Carte à l'issue du 2^{ème} parcours

Aucune autre étiquette n'est plus modifiée

- **Problème (collisions)** : d'autres parcours (jusqu'à ce qu'il n'y ait plus de changement) sont nécessaires pour obtenir la carte finale. Ex. après 3^{ème} parcours en sens classique :

0	0	0	0	0	0	0
0	0	1	1	0	1	1
0	1	1	0	0	0	1
0	1	0	0	1	1	1
0	0	0	0	0	0	0

0	0	0	0	0	0	0
0	0	1	1	0	1	1
0	1	1	0	0	0	1
0	1	0	0	1	1	1
0	0	0	0	0	0	0

0	0	0	0	0	0	0
0	0	1	1	0	1	1
0	1	1	0	0	0	1
0	1	0	0	1	1	1
0	0	0	0	0	0	0

L	0	0	0	0	0	0	0
	0	0	1	1	0	2	2
	0	3	1	0	0	0	2
	0	3	0	0	2	2	2
	0	0	0	0	0	0	0

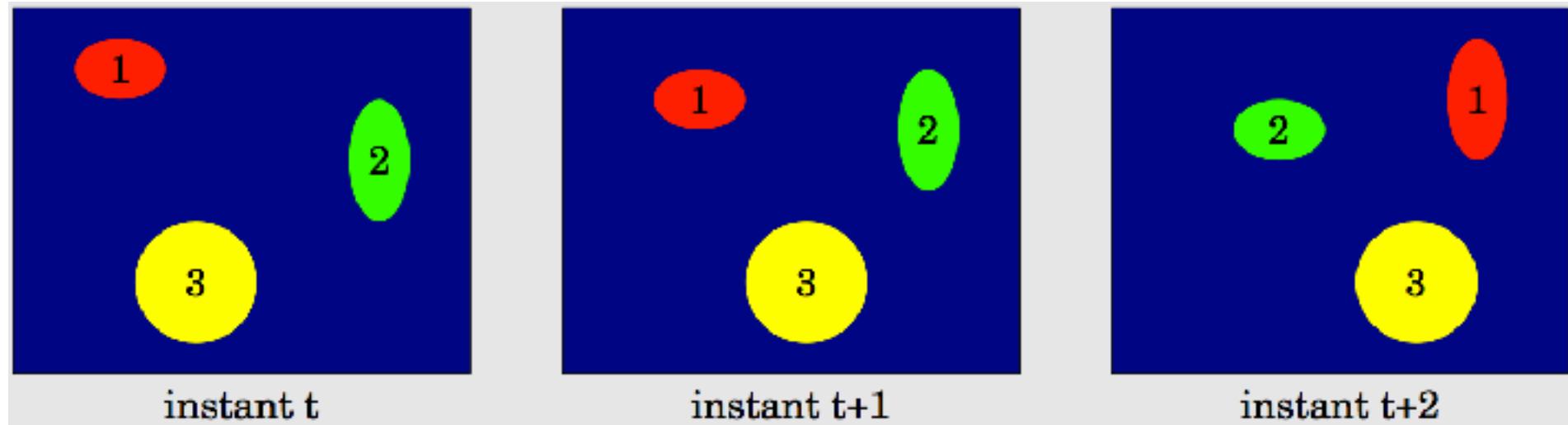
0	0	0	0	0	0	0
0	0	1	1	0	2	2
0	1	1	0	0	0	2
0	3	0	0	2	2	2
0	0	0	0	0	0	0

0	0	0	0	0	0	0
0	0	1	1	0	2	2
0	1	1	0	0	0	2
0	3	0	0	2	2	2
0	0	0	0	0	0	0

0	0	0	0	0	0	0
0	0	1	1	0	2	2
0	1	1	0	0	0	2
0	1	0	0	2	2	2
0	0	0	0	0	0	0

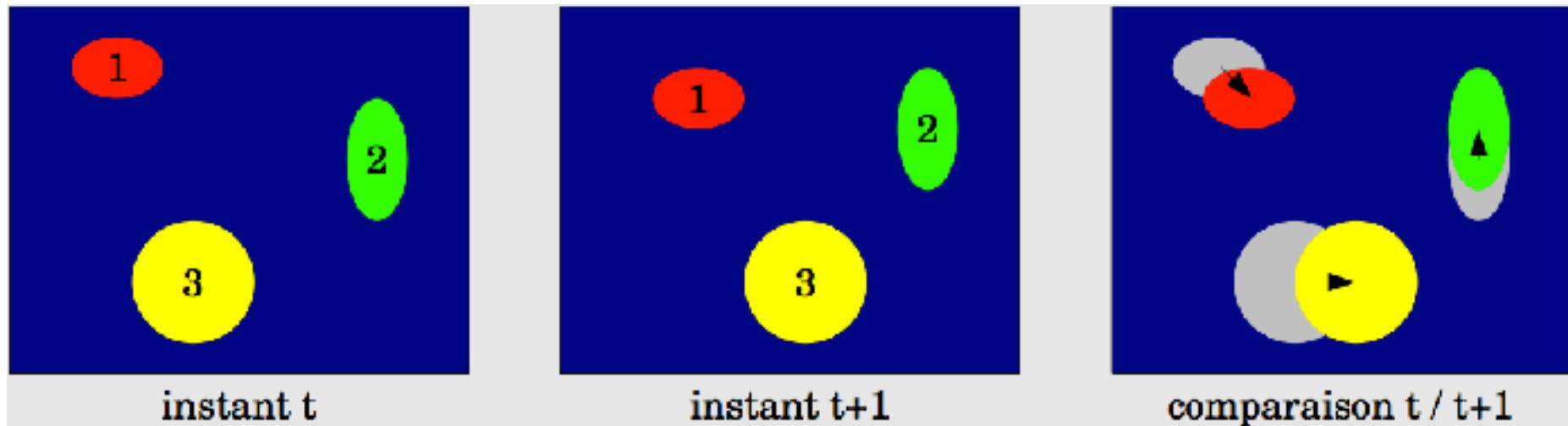
Suivi des blobs

38



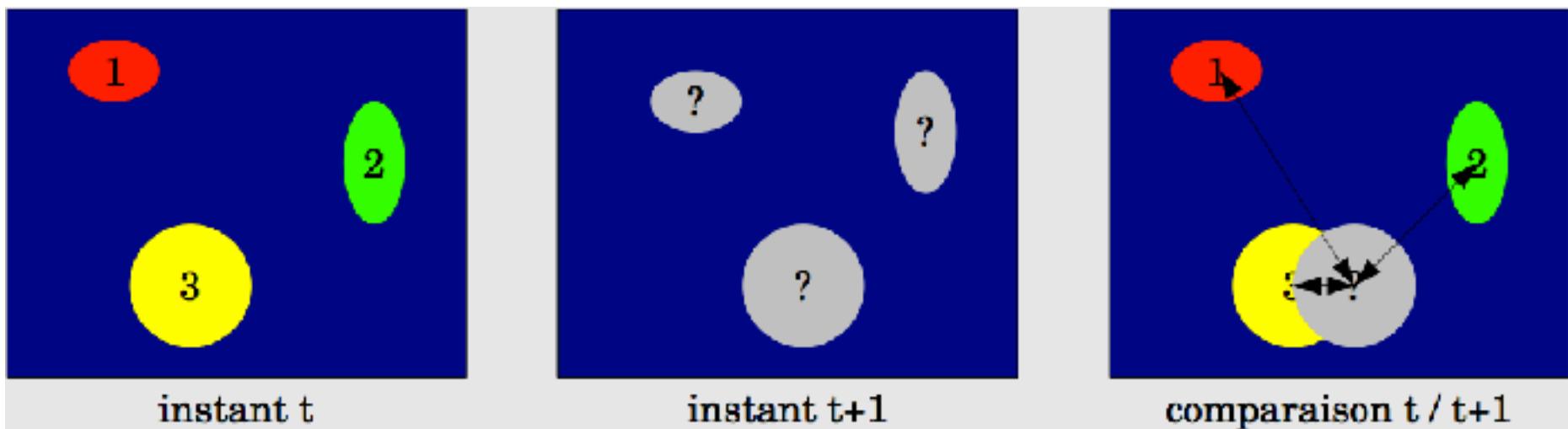
Distances inter-blobs

39



Distances inter-blobs

40



Distances inter-blobs

41

t	$t+1$?	?	?
1	25.3	1.9	40.3	
2	3.2	27	36.4	
3	41.6	33.2	4.2	

t	$t+1$?	?	?
1	1		1.9	
2	2	3.2		
3	3			4.2

t	$t+1$	2	1	3
1	1		1.9	
2	2	3.2		
3	3			4.2

Disparition d'un objet

42

A diagram illustrating the disappearance of object 1 from a sequence. It shows three stages: at time t , object 1 is present in the first slot; at time $t+1$, object 1 has disappeared, replaced by two question marks.

1	25.3	1.9
2	3.2	27
3	41.6	33.2

A diagram illustrating the disappearance of object 2 from a sequence. It shows three stages: at time t , object 2 is present in the second slot; at time $t+1$, object 2 has disappeared, replaced by two question marks.

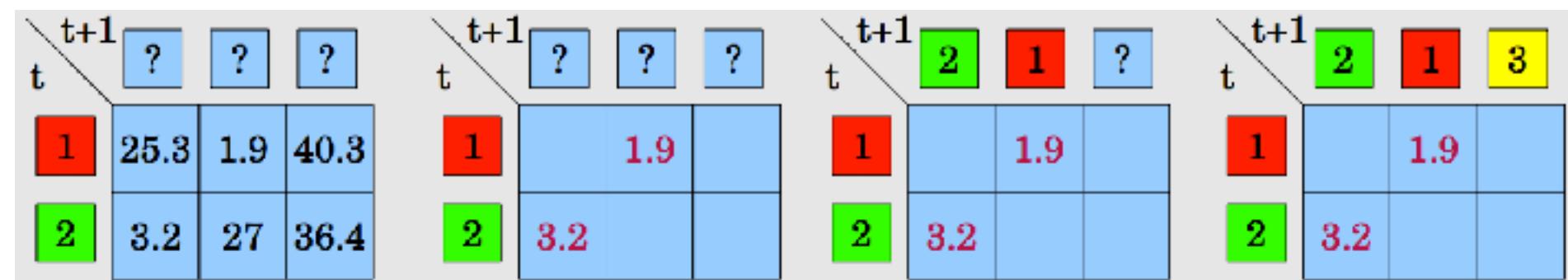
1	?	?
2	?	?
3	?	?

A diagram illustrating the disappearance of object 3 from a sequence. It shows three stages: at time t , object 3 is present in the third slot; at time $t+1$, object 3 has disappeared, replaced by two question marks.

2	1	
1		1.9
2		3.2
3		

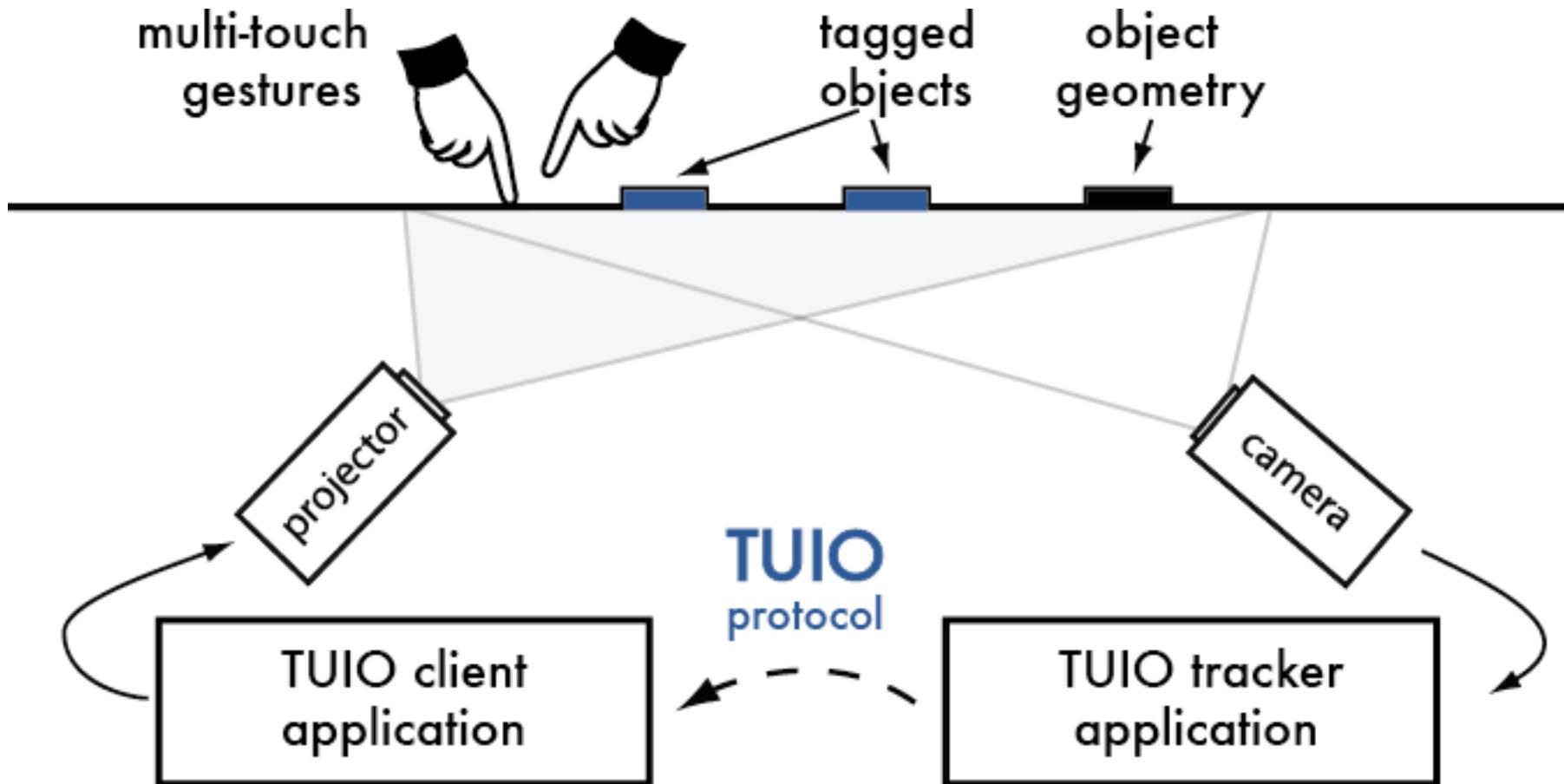
Apparition d'un objet

43



TUIO

44



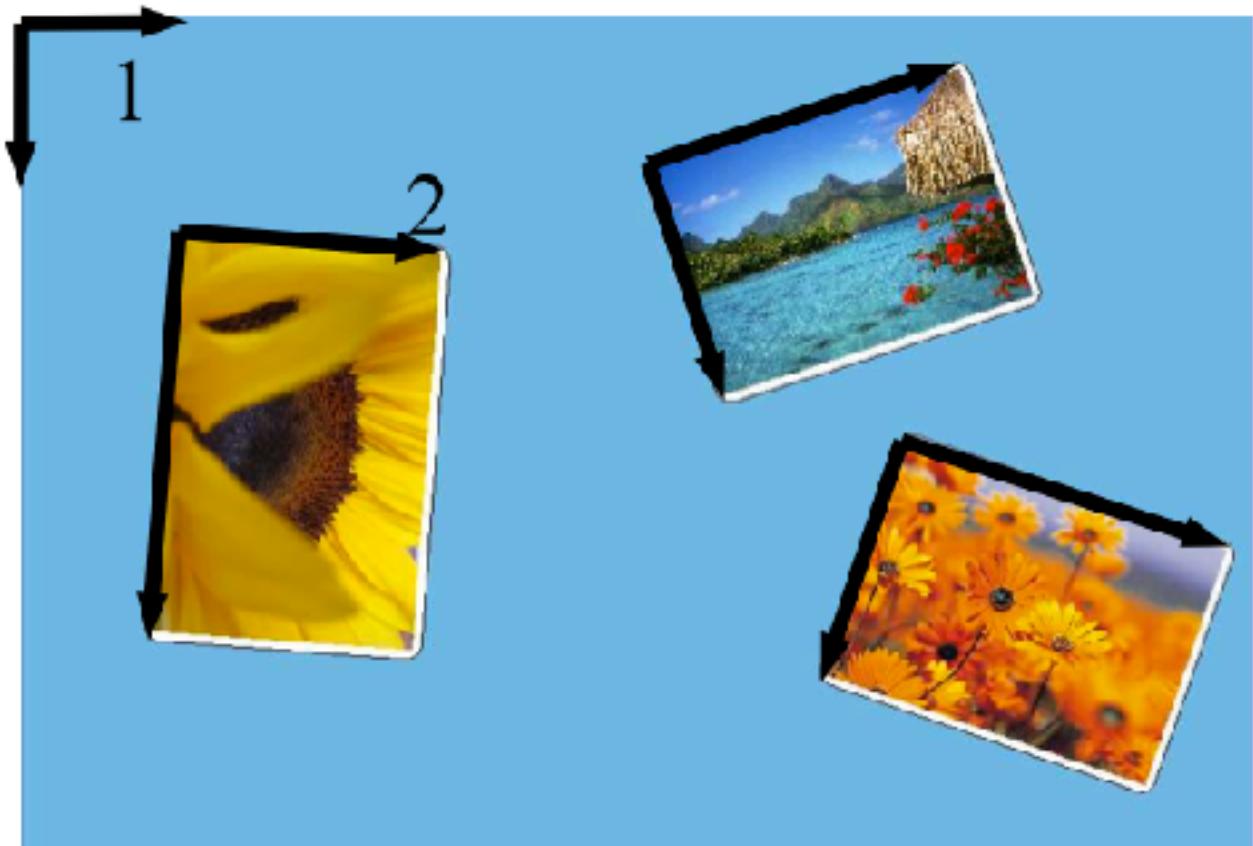
Interaction

45



OBB

46



Le repère 2 est placé dans le repère 1 en déplaçant 1 vers 2 (on dit qu'on passe du repère 1 au repère 2, et sera noté $M_{1 \rightarrow 2}$) :

- ➊ En faisant une translation (donnée par le champs `position`) notée T
- ➋ Puis une rotation autour de l'origine actuelle (l'angle est donné par `angle`) notée R
- ➌ Puis un changement d'échelle (l'axe `x` actuel est multiplié par `width`, l'axe actuel `y` par `height`) notée S .
- ➍ → on note ainsi le passage de 1 vers 2 : $M_{1 \rightarrow 2} = TRS$.

Expression d'un point dans des repères différents

47

Soit un point P connu dans un repère 2 (noté P_2). Soit un repère 1 dont on connaît le passage $M_{1 \rightarrow 2}$.

Alors les coordonnées de P dans 1 sont donnés par la relation $P_1 = M_{1 \rightarrow 2}P_2$.

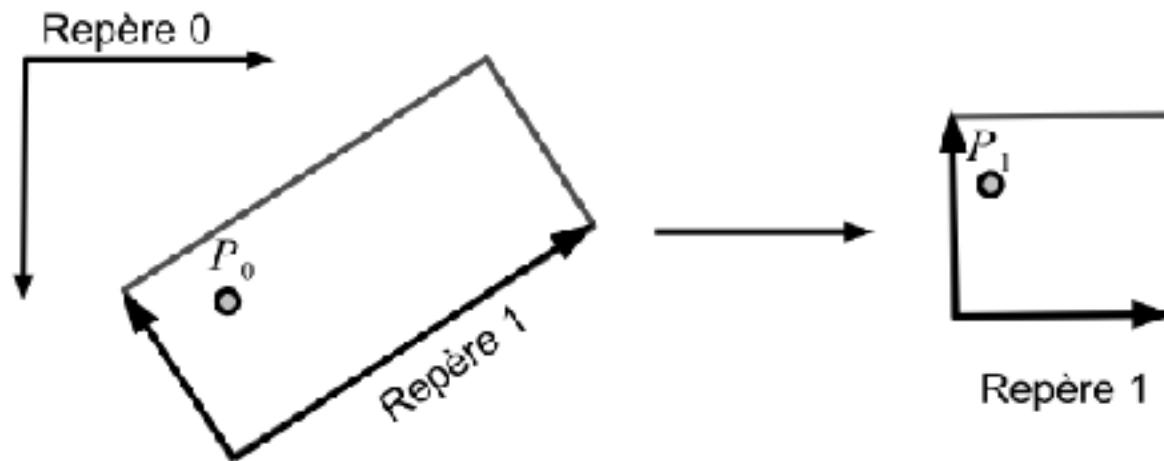
Exemple : $M_{1 \rightarrow 2} = TRS$, et $P_2(x_2, y_2)$, alors $P_1(x_1, y_1) = TRSP_2(x_2, y_2)$:

- $S(k_x, k_y)$: $x' = k_x \times x_2$ et $y' = k_y \times y_2$.
- puis $R(\theta)$: $x'' = x' \cos \theta - y' \sin \theta$ et $y'' = x' \sin \theta + y' \cos \theta$
- et finalement $T(t_x, t_y)$: $x_1 = x'' + t_x$ et $y_1 = y'' + t_y$.

Localiser un point dans une OBB

48

- ▶ Sélection d'un composant = lors d'un "addCursor(x,y)" on détermine à quel composant correspond (x,y) .
- ▶ Donnée : un point $P(x,y)$ dans le repère initial 0 (curseur en coordonnée pixel).
- ▶ Problème : à quel composant appartient P ?



Localiser un point dans une OBB

49

- ▶ une OBB est un carré défini dans un repère local 1 dont on a fait subir $M_{0 \rightarrow 1} = TRS$ pour le placer par rapport au repère 0 ($T = \text{origin}$, $R = \text{angle}$, $S = (\text{width}, \text{height})$).
- ▶ il est plus aisé de tester l'appartenance du point P dans le carré \Rightarrow exprimer P dans 1 par $P_1 = M_{1 \rightarrow 0}P_0$.
- ▶ Cas général : il faut inverser $M_{p \rightarrow q}$ pour obtenir $M_{q \rightarrow p}$.
- ▶ Cas d'une Composition : l'inverse de $M_{p \rightarrow q}M_{q \rightarrow r}$ est $M_{r \rightarrow q}M_{q \rightarrow p}$
- ▶ Par ailleurs $T^{-1} = -T$, $R^{-1}(\theta) = R(-\theta)$ et $S^{-1}(k_x, k_y) = S(\frac{1}{k_x}, \frac{1}{k_y})$.

Superposition de composants

50

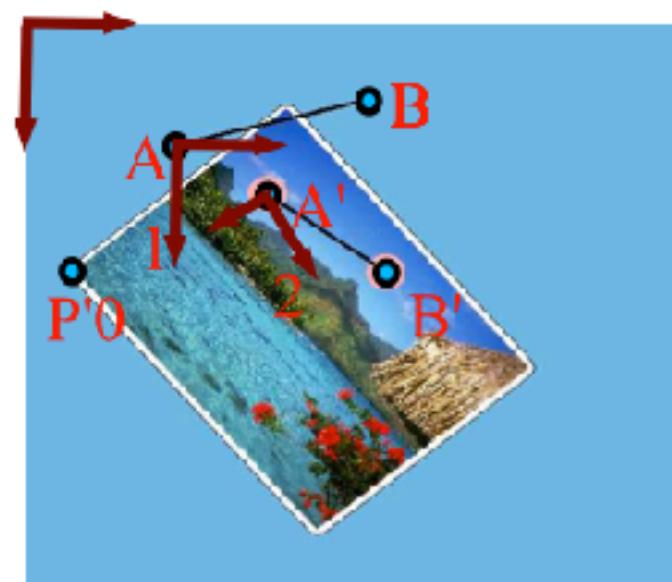
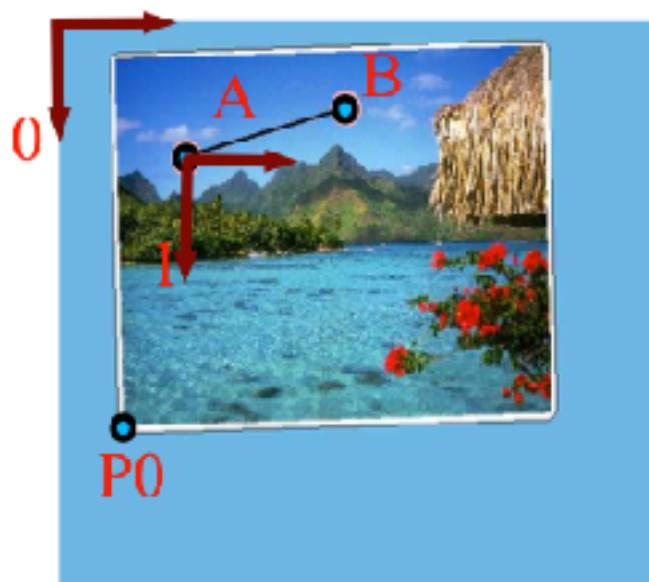
- ▶ L'ordre dans la liste des composants définit l'ordre d'affichage (le premier à l'arrière-plan), ou profondeur d'affichage.
- ▶ Si il y a superposition de composants lors d'une sélection ⇒ le composant sélectionné est celui qui est visible (i.e. le moins profond).
- ▶ Une fois sélectionné, le composant passe à l'avant plan.



Mouvement d'un objet avec 2 doigts

51

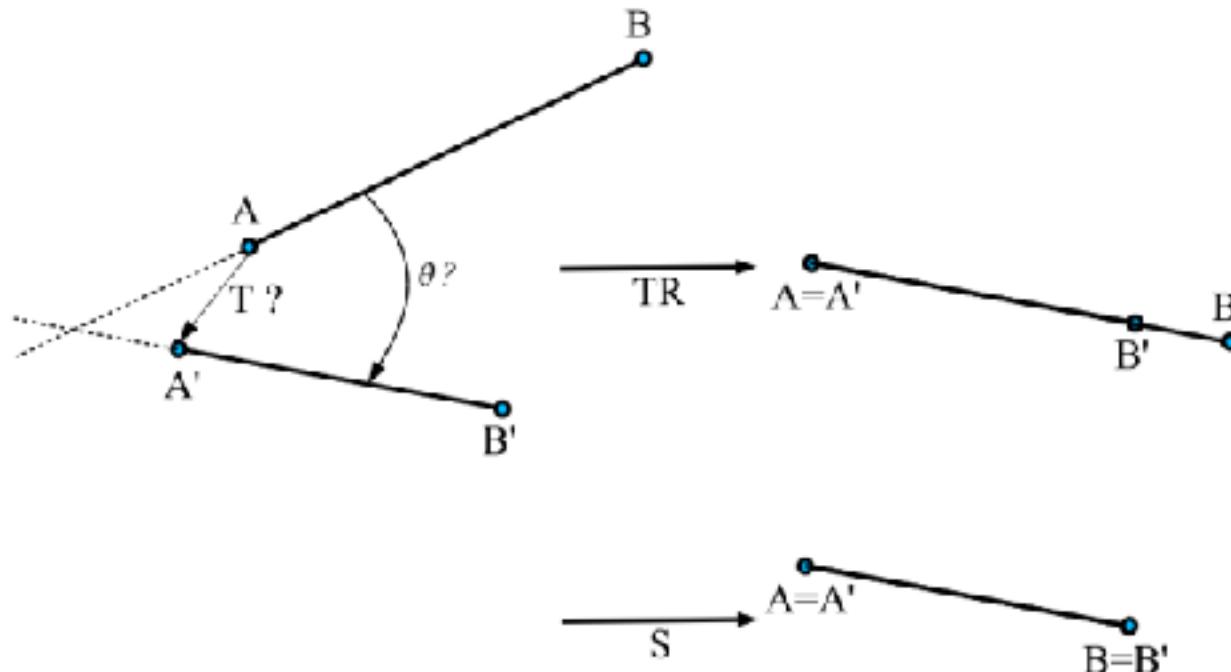
- ▶ 2 curseurs sur l'image : *A* et *B*.
- ▶ Attacher l'image aux curseurs.
- ▶ Mouvement des curseurs : *A* déplacé en *A'* et *B* en *B'*.



Transformation du segment AB

52

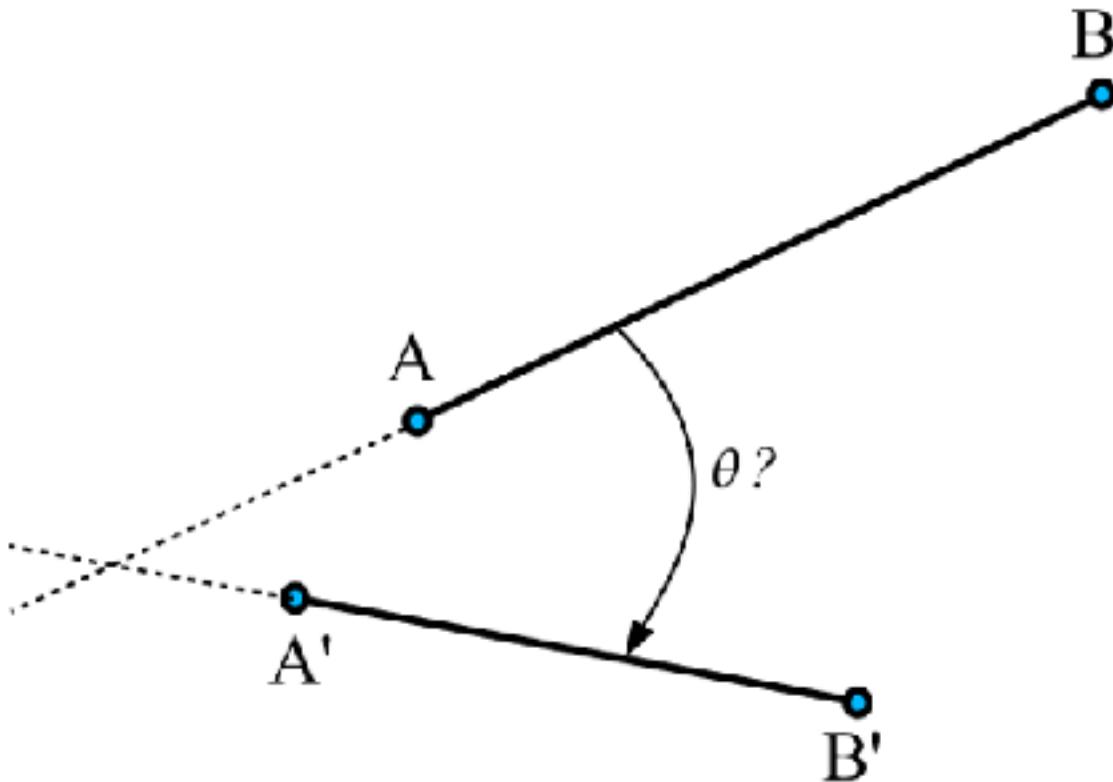
- ▶ On "attache" AB à un repère 1 (l'origine de 1 est A) et on exprime le changement de repère $M_1 \rightarrow M_2$ pour aller en $A'B'$ (l'origine de 2 est A') :



- ▶ $T =$ translation de vecteur AA' .
- ▶ puis $R =$ rotation d'angle ?
- ▶ puis $S =$ rapport des longueurs entre AB et $A'B'$.

Calcul de l'angle de rotation

53



- On utilise le produit scalaire entre AB et $A'B'$, car $AB \cdot A'B' = \|AB\| \|A'B'\| \cos\theta$.

Produit scalaire

54

Soient $u = \begin{pmatrix} u_x \\ u_y \end{pmatrix}$, $v = \begin{pmatrix} v_x \\ v_y \end{pmatrix}$, le produit scalaire $u \cdot v$ (appelé dot en anglais) est le nombre donné par :

- ▶ $u \cdot v = u_x v_x + u_y v_y$ (Calcul par coordonnées)
- ▶ $u \cdot v = \|u\| \|v\| \cos(u, v)$ (Calcul par cosinus)

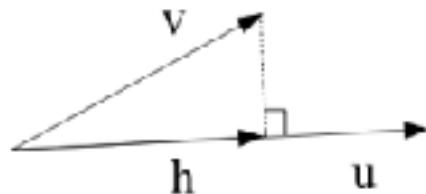
Relation avec la norme (la « longueur ») du vecteur u :

- ▶ Norme (euclidienne) : $\|u\| = \sqrt{u \cdot u} = \sqrt{u_x u_x + u_y u_y}$
- ▶ Remarque : Normer u consiste à « rendre » le vecteur u de norme 1 : $u' = \frac{u}{\|u\|}$

Produit scalaire

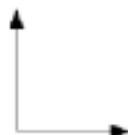
55

► Projection du vecteur v sur u :



- $u \cdot v = u \cdot h$
- $\|h\| = \frac{|u \cdot v|}{\|u\|}$
- Si u est unitaire : $h = (u \cdot v)u$

► Localisation relative des vecteurs u et v :



$u \cdot v = 0$ (orthonormal)



$u \cdot v > 0$ (aigu)



$u \cdot v < 0$ (obtus)

► Si u et v sont de norme 1 :

- $u \cdot v = \cos(u, v)$
- $\cos(u \cdot v)$ donne un angle dans $[0, \pi]$.
- \Rightarrow le produit scalaire ne suffit pas à lui seul :



Déterminant

56

► Calcul :

Soient $u = \begin{pmatrix} u_x \\ u_y \end{pmatrix}$, $v = \begin{pmatrix} v_x \\ v_y \end{pmatrix}$, le déterminant est donné par le nombre :

$$\det(u, v) = u_x v_y - u_y v_x$$

Remarque : il s'agit de la 3ième coordonnée du produit vectoriel entre $\begin{pmatrix} u_x \\ u_y \\ u_z = 0 \end{pmatrix}$ et $\begin{pmatrix} v_x \\ v_y \\ v_z = 0 \end{pmatrix}$

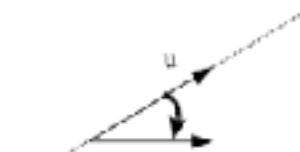
$$\begin{pmatrix} v_x \\ v_y \\ v_z = 0 \end{pmatrix}$$

► Interprétation :



$$\det(u, v) > 0$$

u vers v : dans le sens direct
appliquer θ



$$\det(u, v) < 0$$

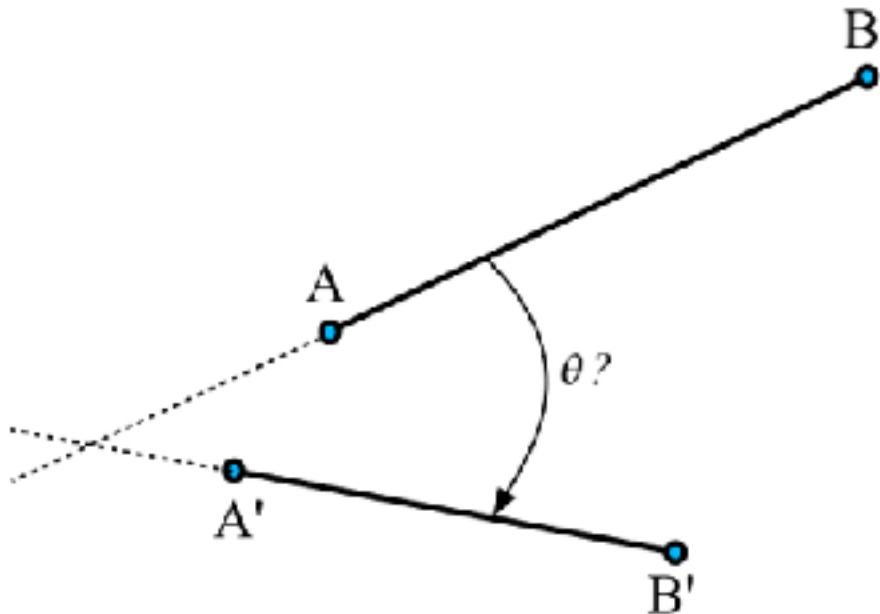
u vers v : dans le sens indirect
appliquer $-\theta$



$$\det(u, v) = 0$$

Résumé du calcul de l'angle de rotation

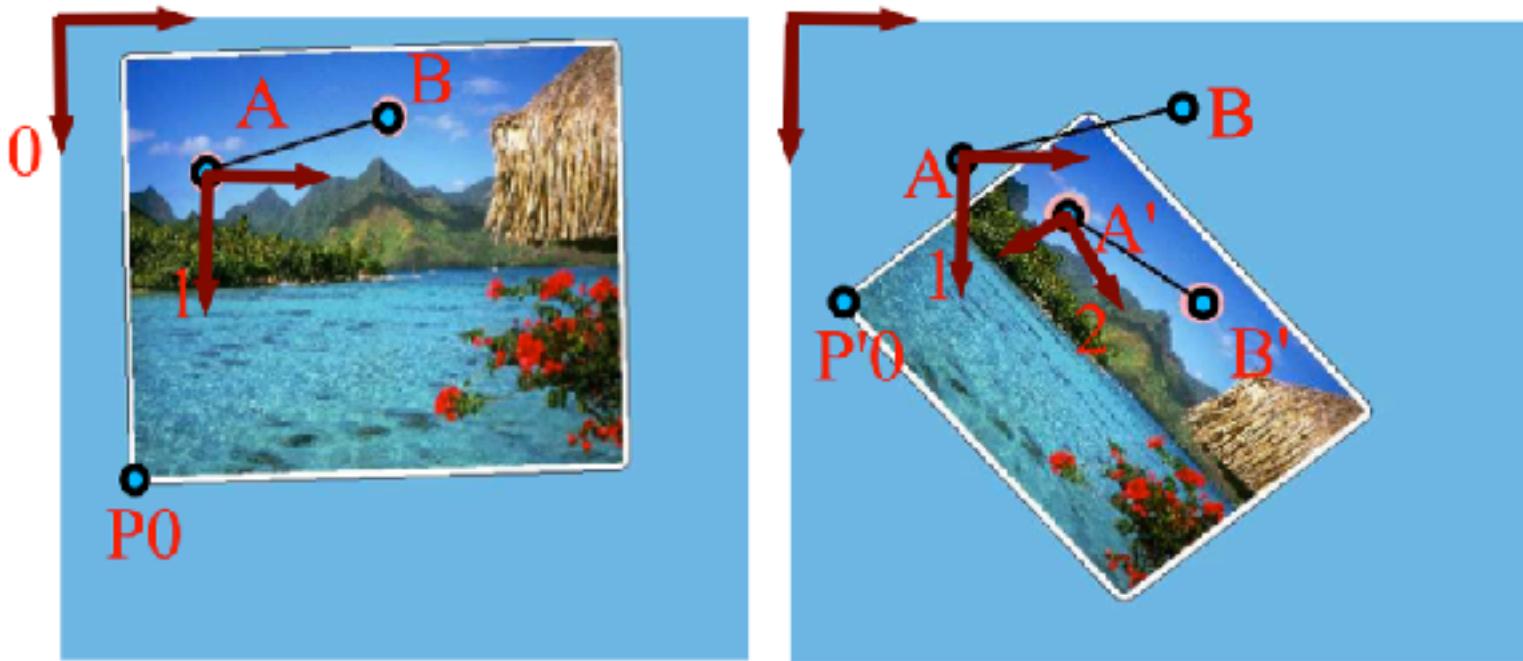
57



- ▶ $u = AB, v = A'B'$
- ▶ On norme u et $v \Rightarrow u \cdot v = \cos(\theta) \Rightarrow \theta = \arccos(u \cdot v)$
- ▶ Calculer $\det(u, v)$ pour savoir si θ est négatif ou positif (i.e. si négatif, prendre $-\theta$ comme angle de rotation).

Résumé du TRS à deux doigts

58



- ▶ width et height subissent le rapport $\frac{\|A'B'\|}{\|AB\|}$
- ▶ angle subit l'angle entre AB et $A'B'$
- ▶ position subit le mouvement de P_0 à P'_0

Résumé du TRS à deux doigts

m_{01} : matrice de passage du repère 0 au repère 1

m_{02} : matrice de passage du repère 0 au repère 2

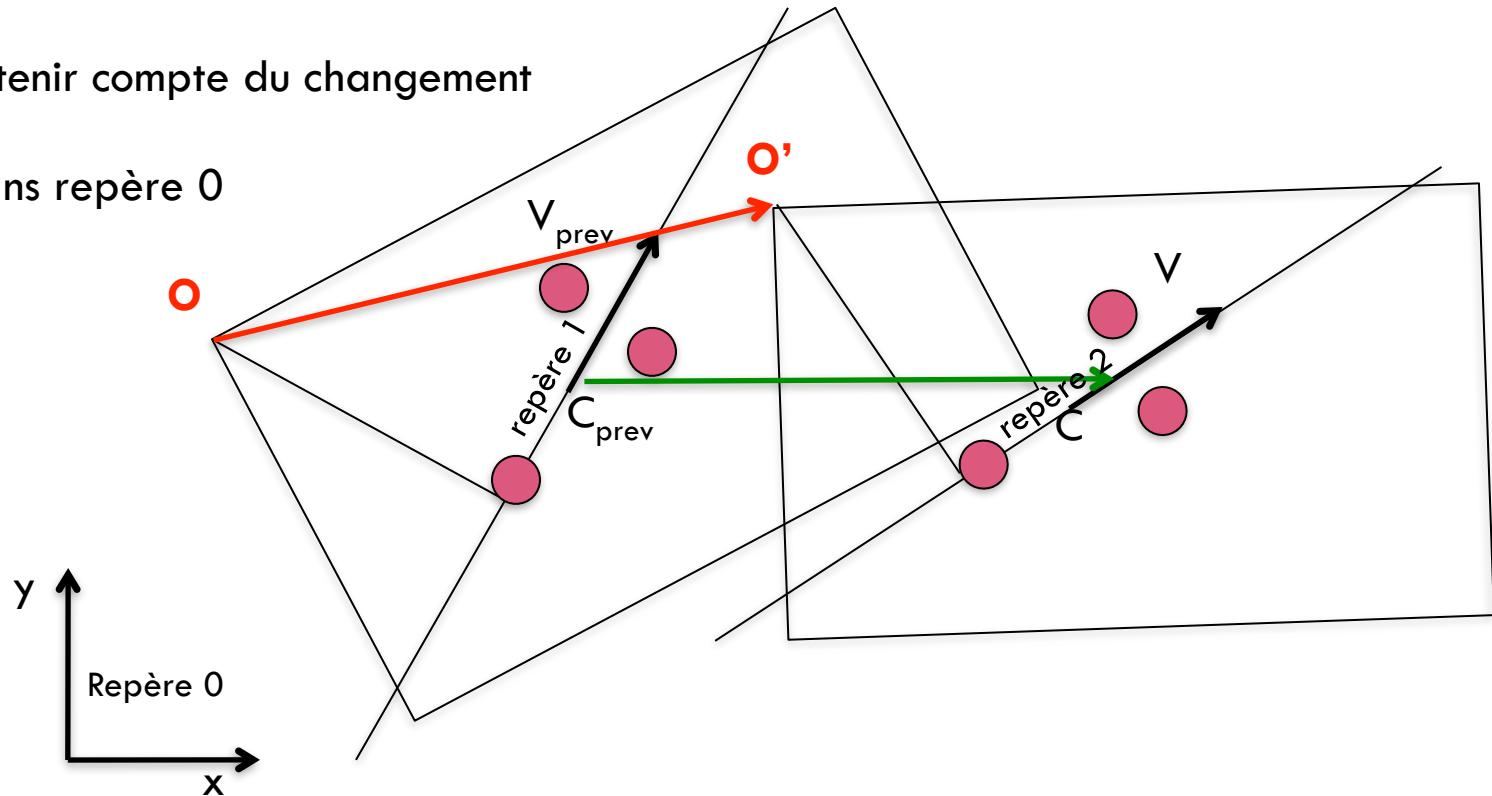
1) Calcul des coordonnées de 0 dans le repère 1:

$$O_1 = m_{01}^{-1} O_0$$

2) $O'_2 = O_1$ (sans tenir compte du changement d'échelle)

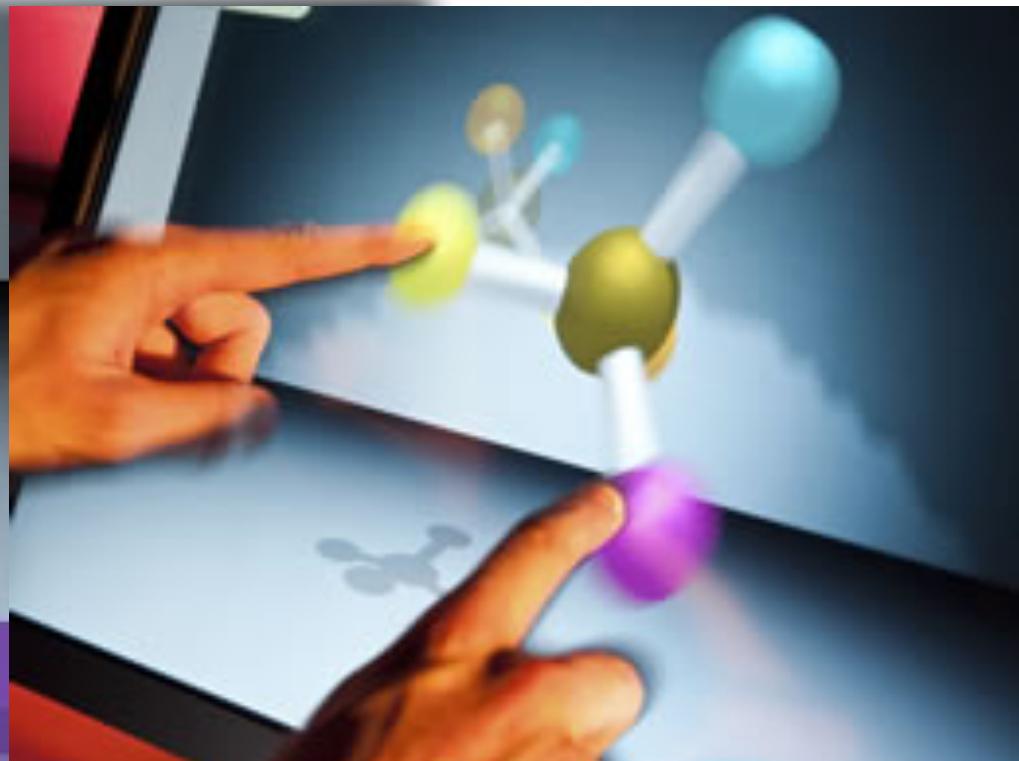
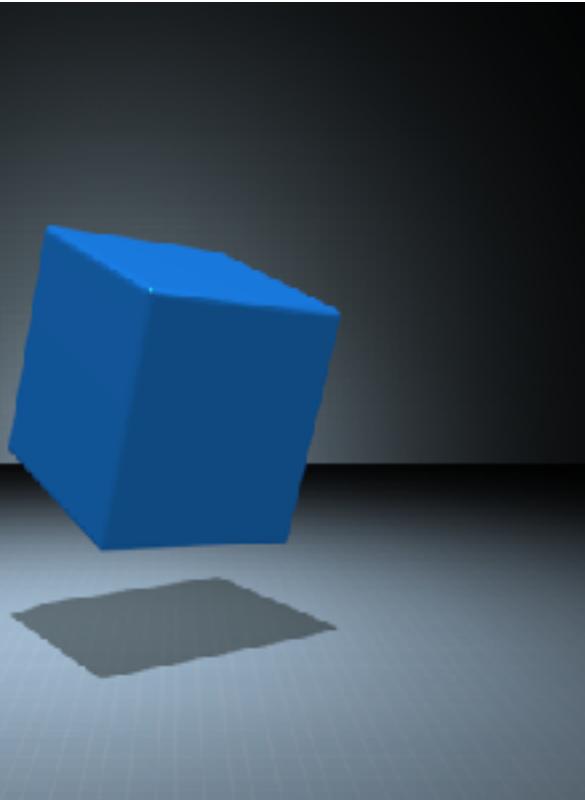
3) Calcul de O' dans repère 0

$$O'_0 = m_{02} O'_2$$



Manipulation 3D

60



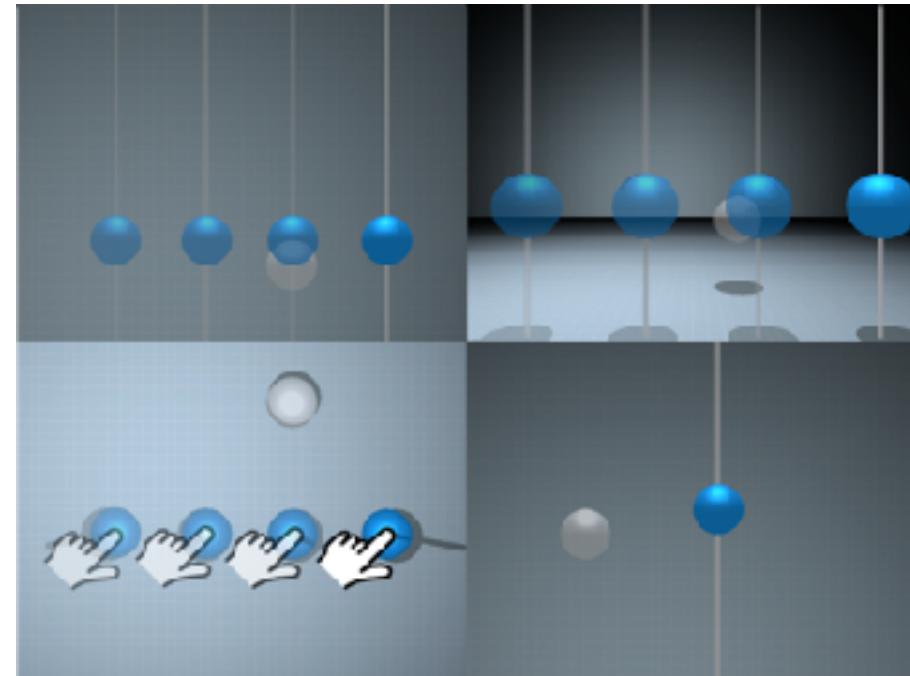
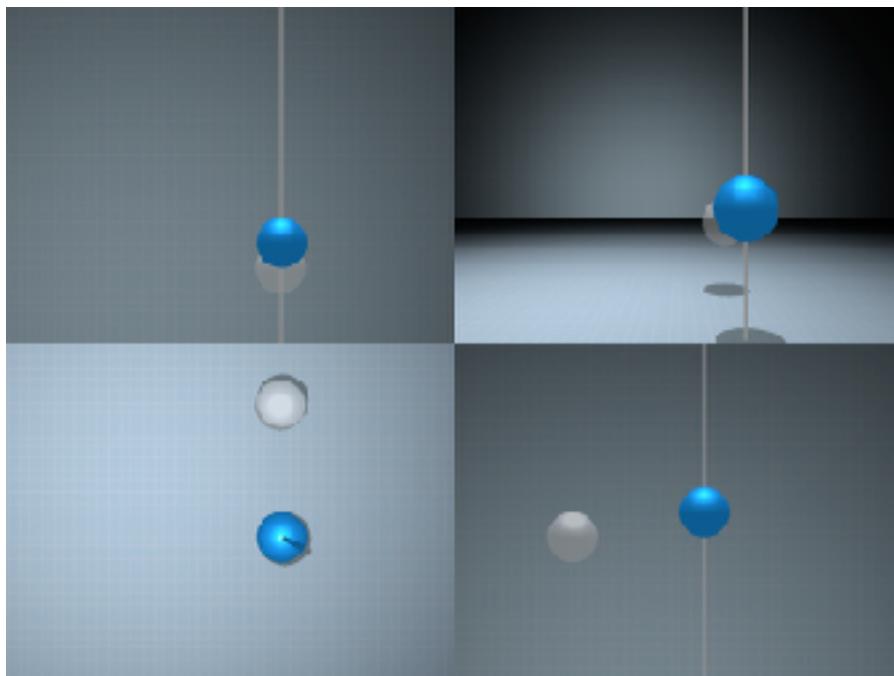


MT-Viewport

62

Amélioration d'une technique traditionnelle

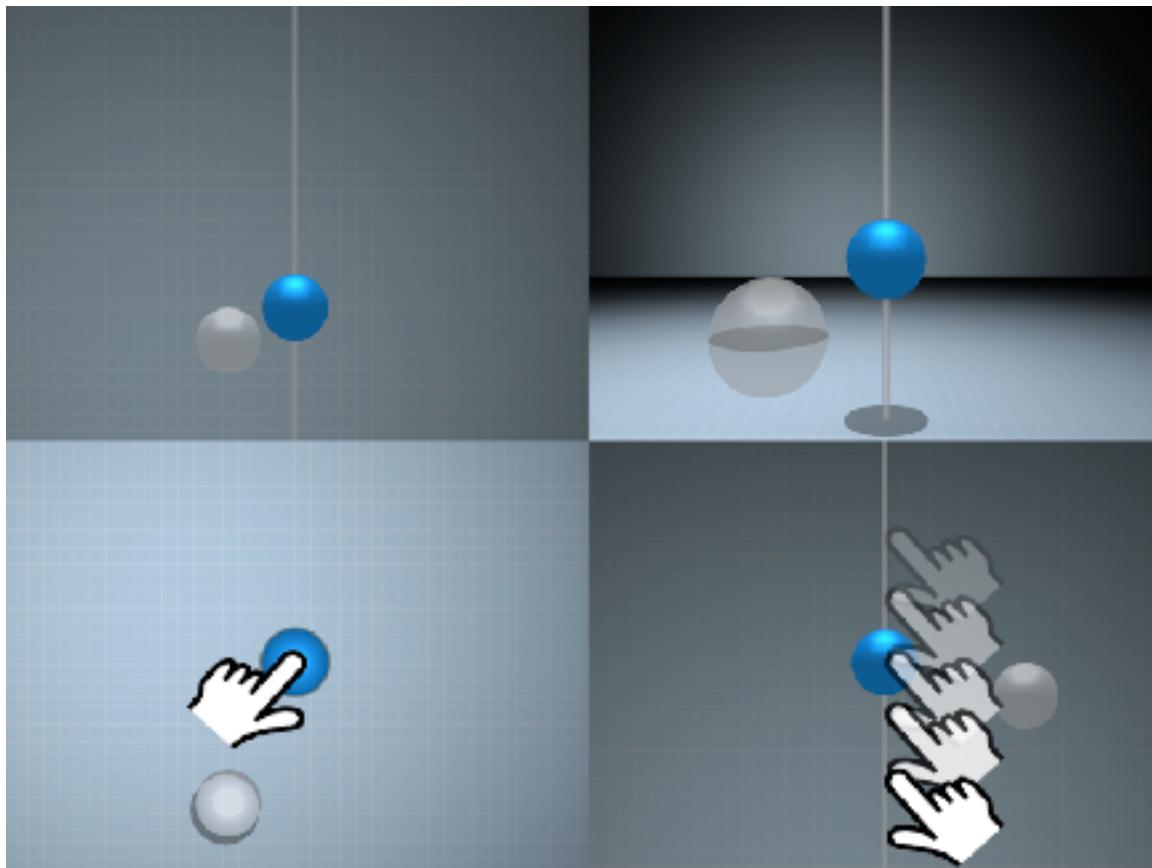
1 vue = 2 DDL



MT-Viewport

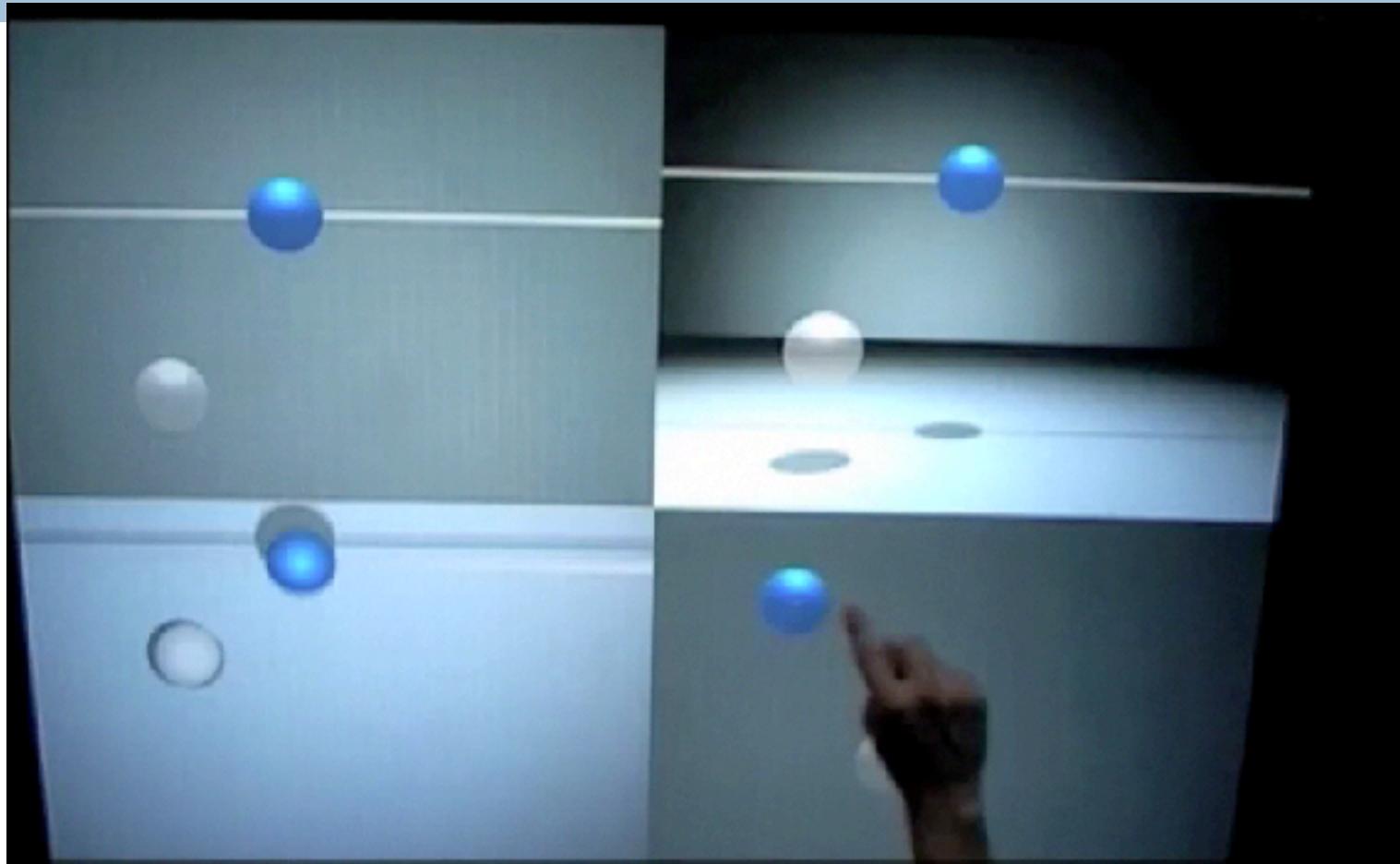
63

3ème DDL dans les autres vues



MT-Viewport

64



Control of 2 DOF on each viewport
like the 4-view technique

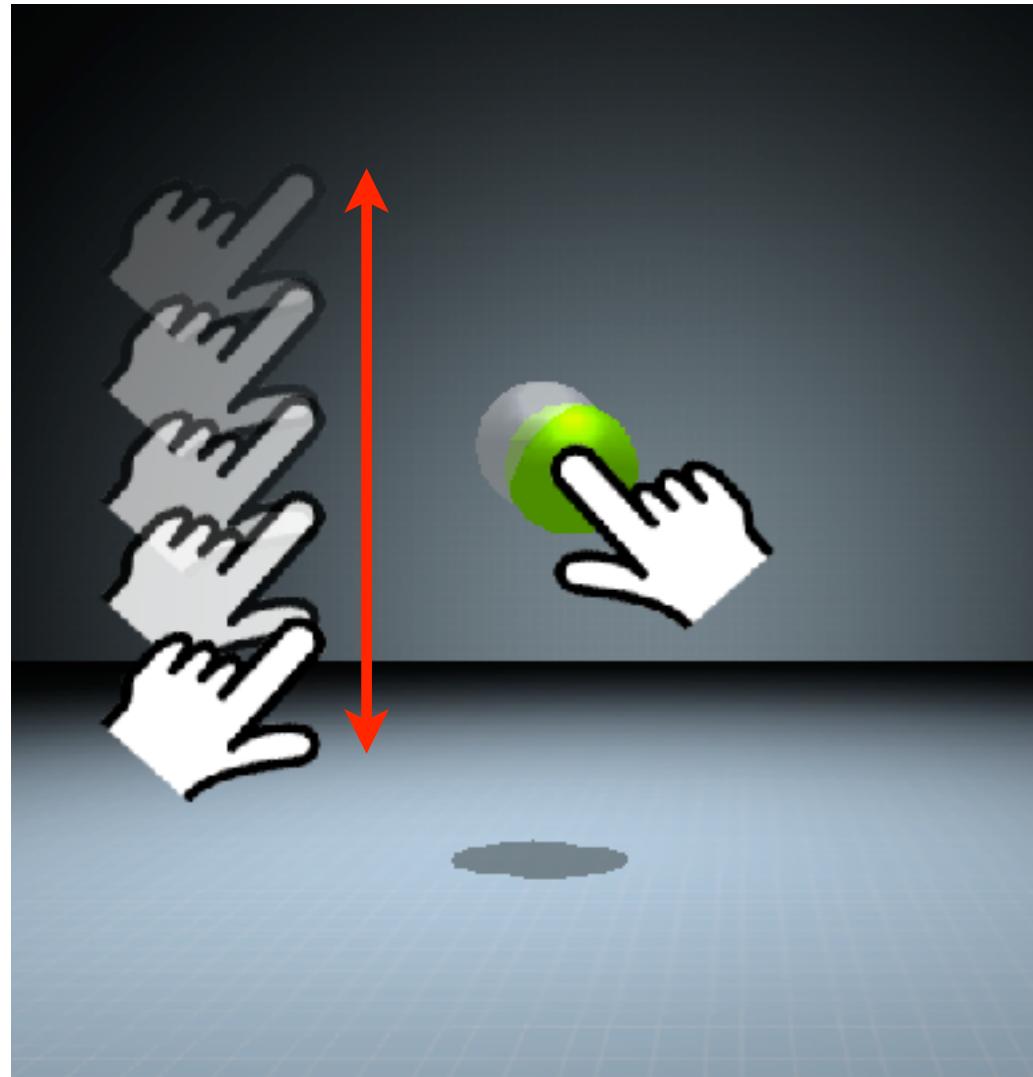
Z-technique [Martinet, Casiez, Grisoni 2010]

65

I doigt : 2 DDL

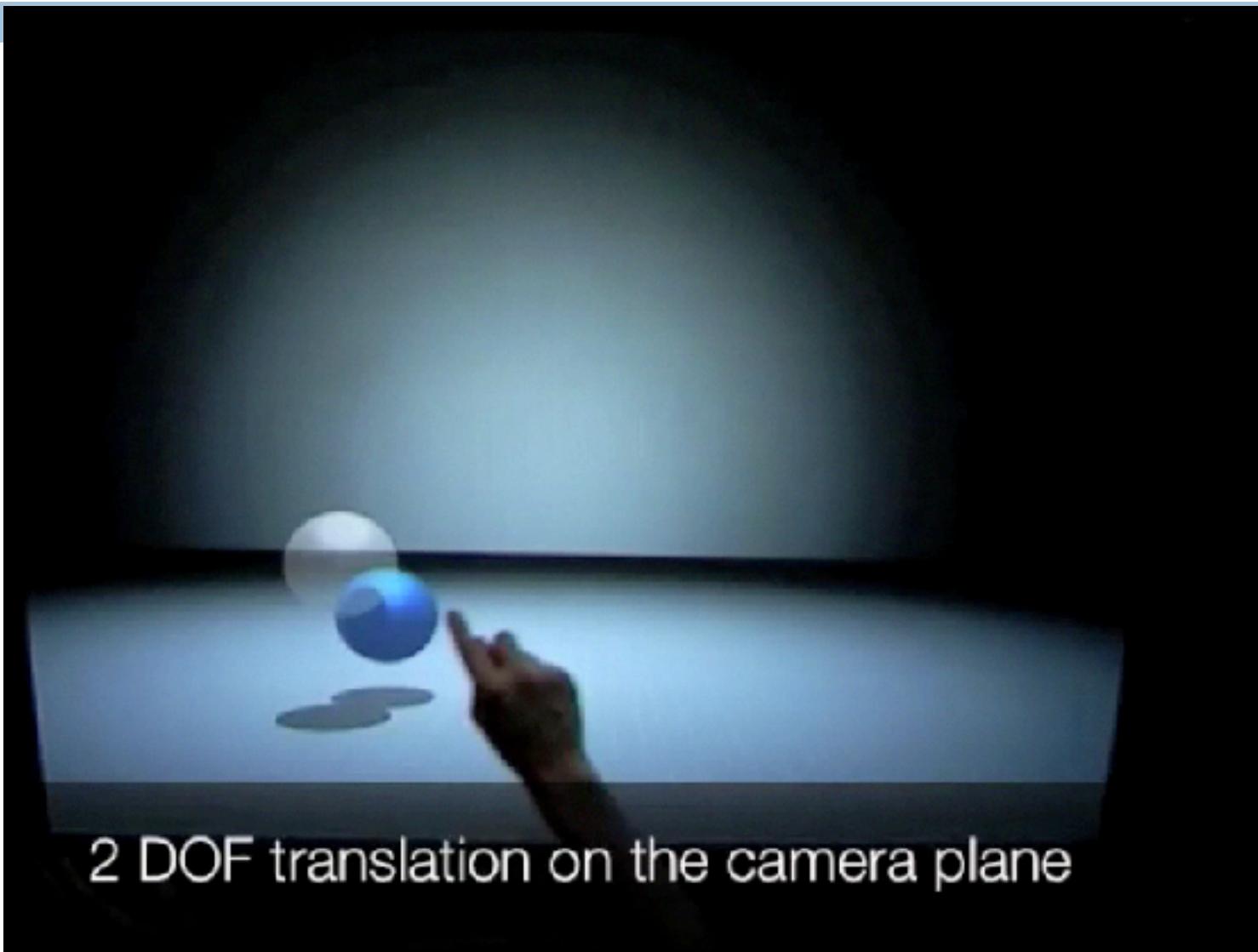
Doigt indirect :

dernier DDL



Z-technique

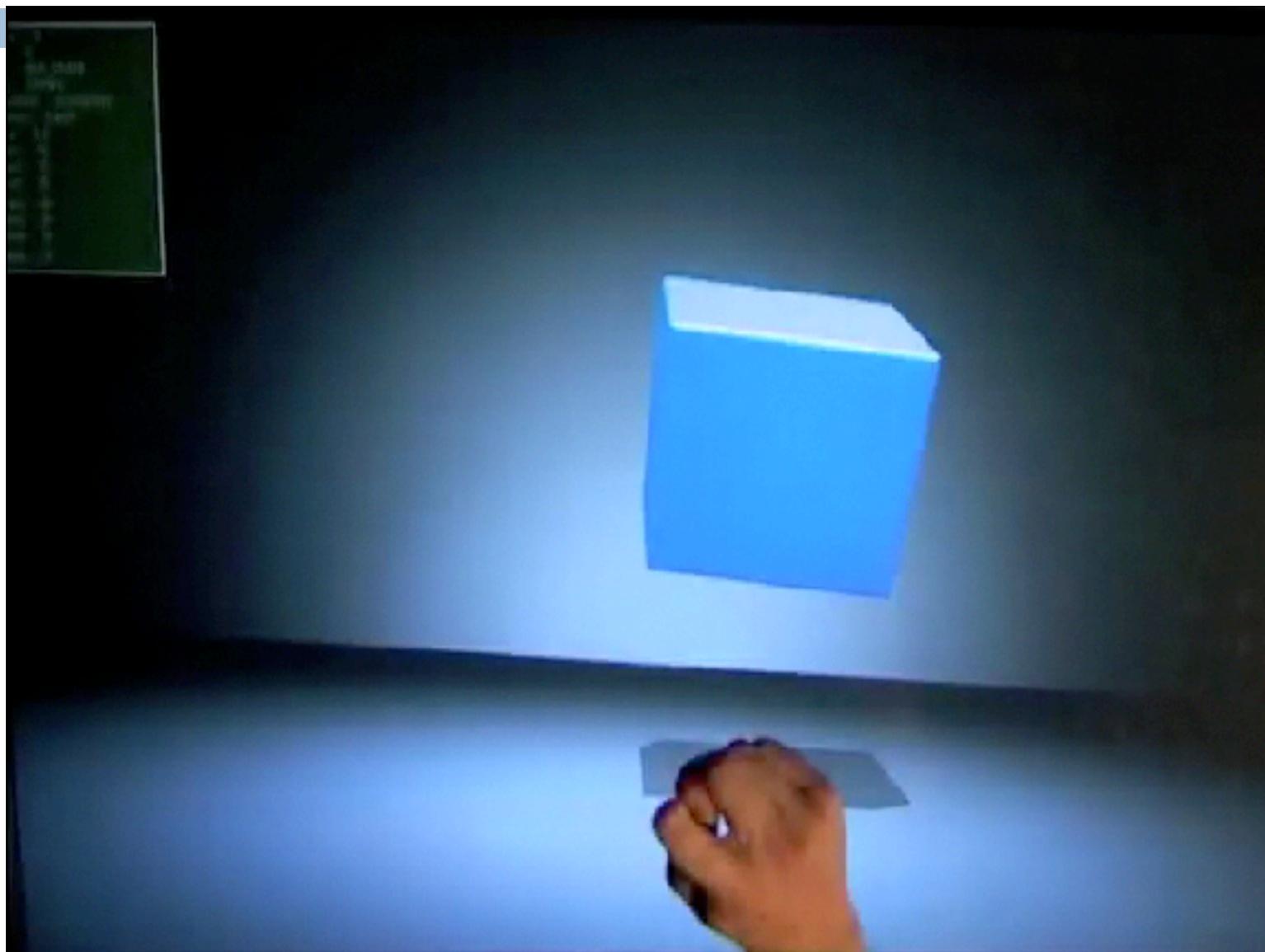
66



2 DOF translation on the camera plane

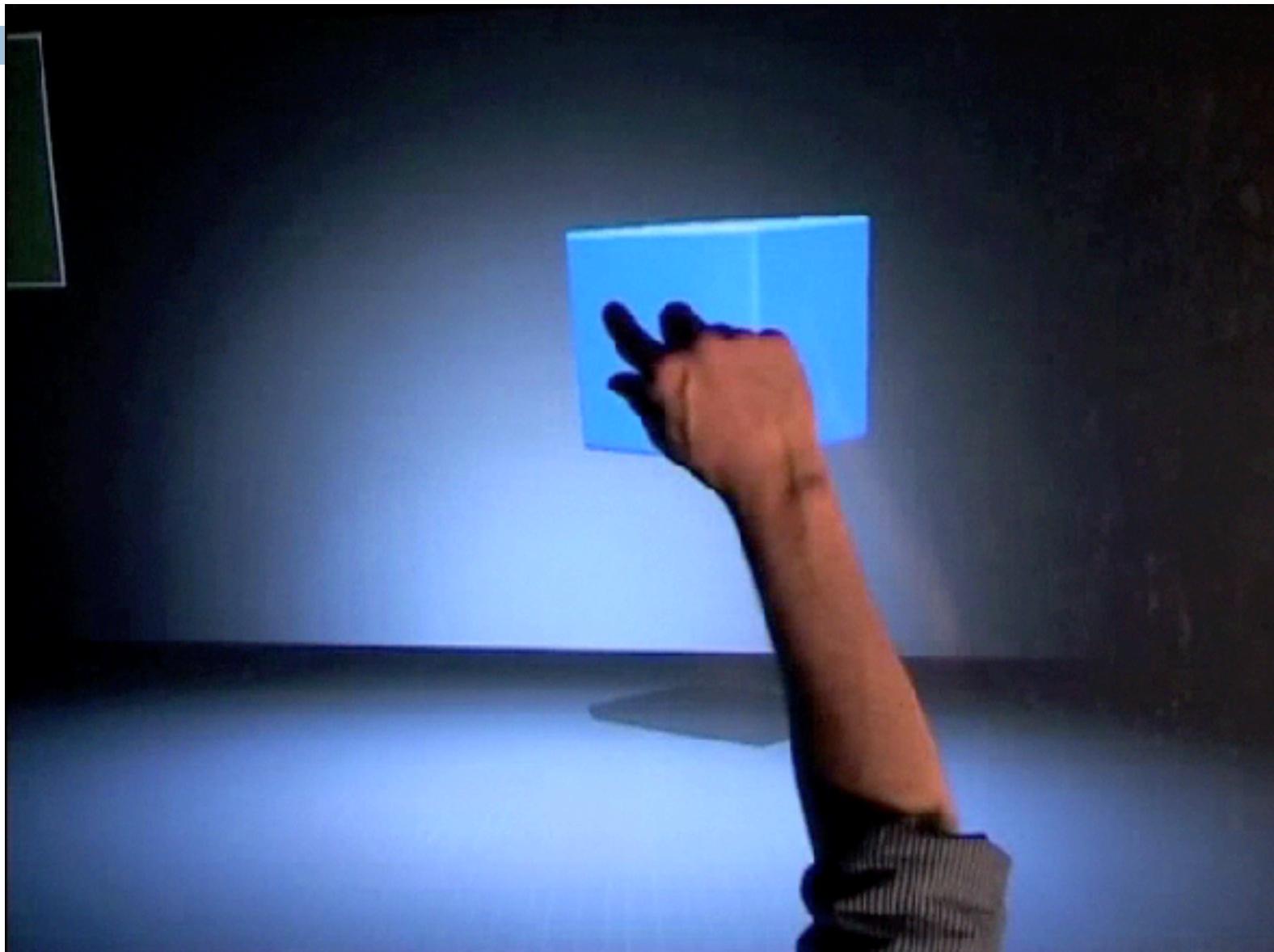
Sticky Tools [Hancock et al. 2009]

67



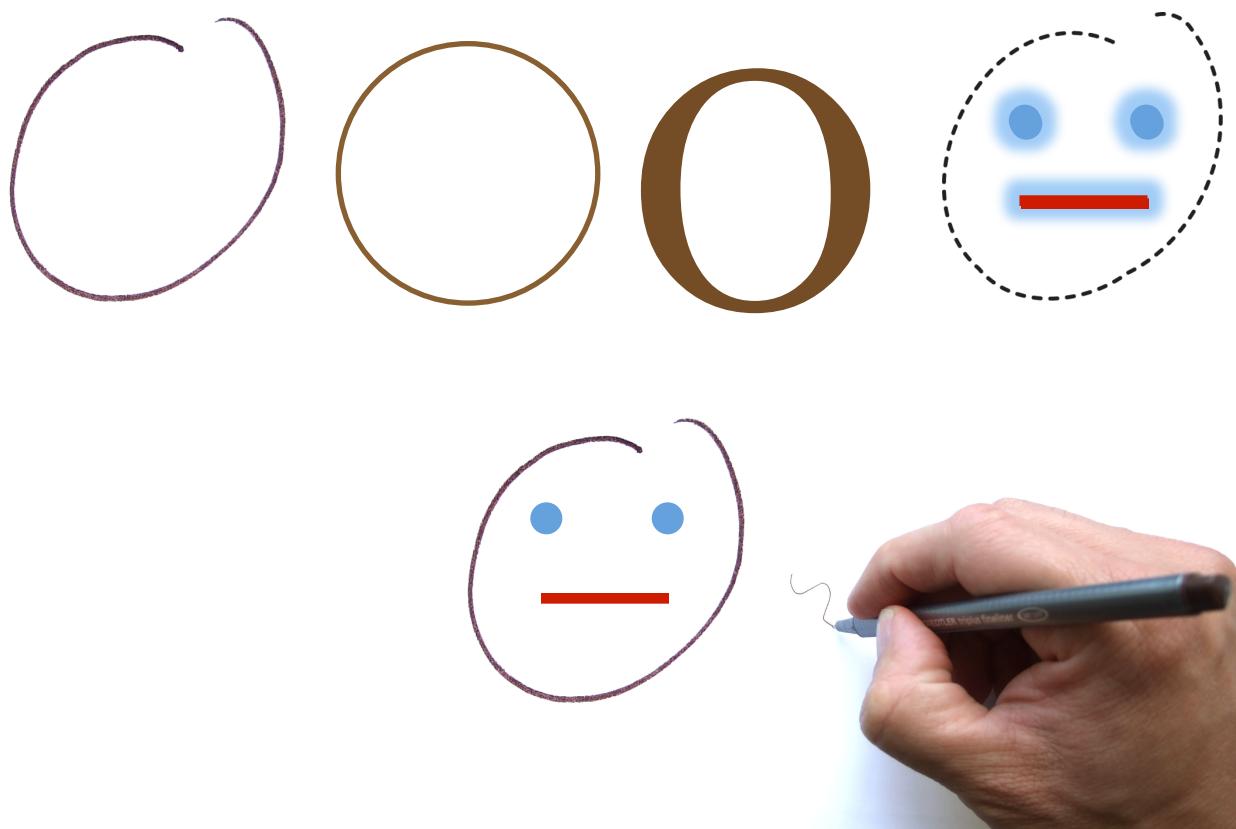
Screenspace [Reisman et al. 2009]

68



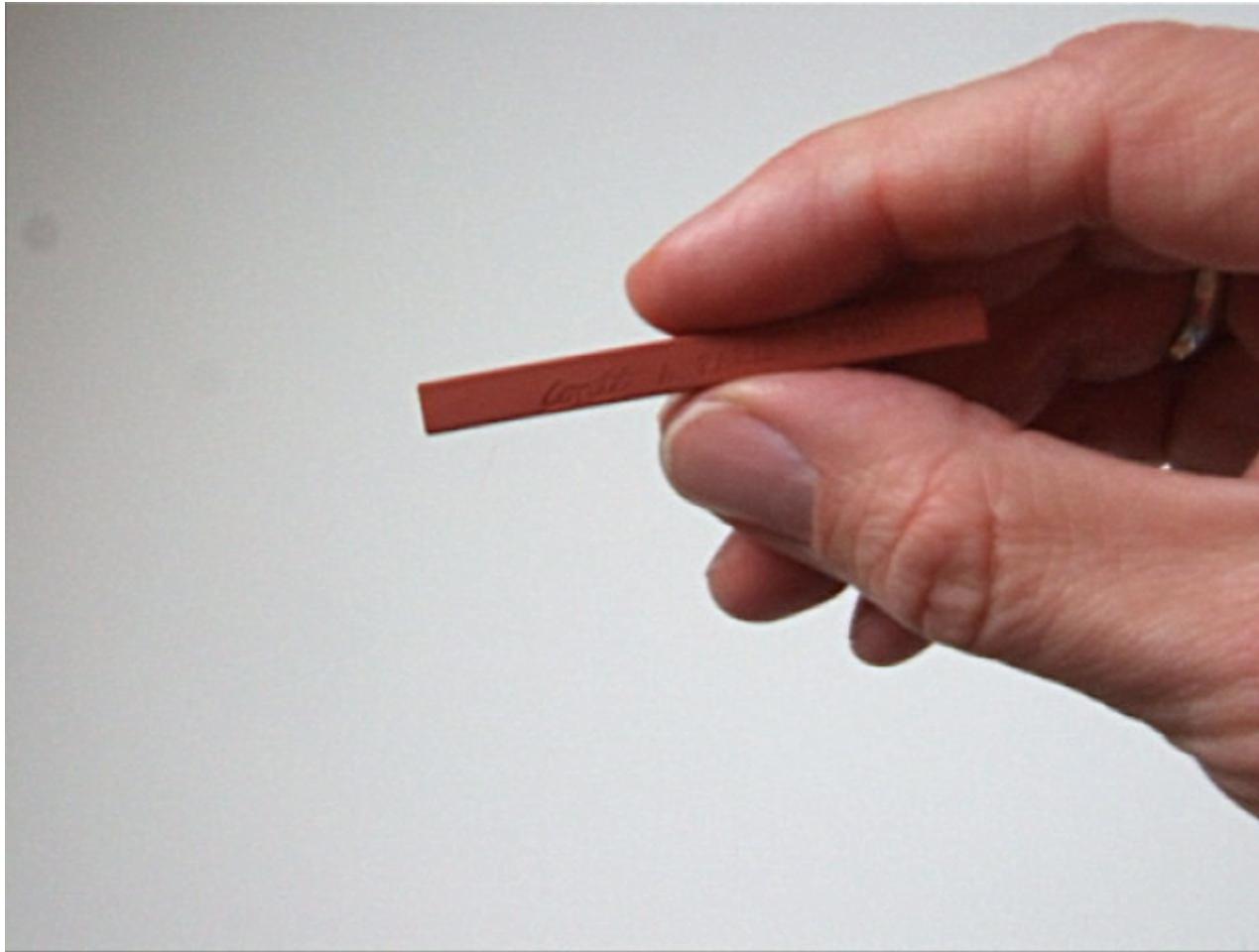
Conté [Vogel & Casiez 2011]

69



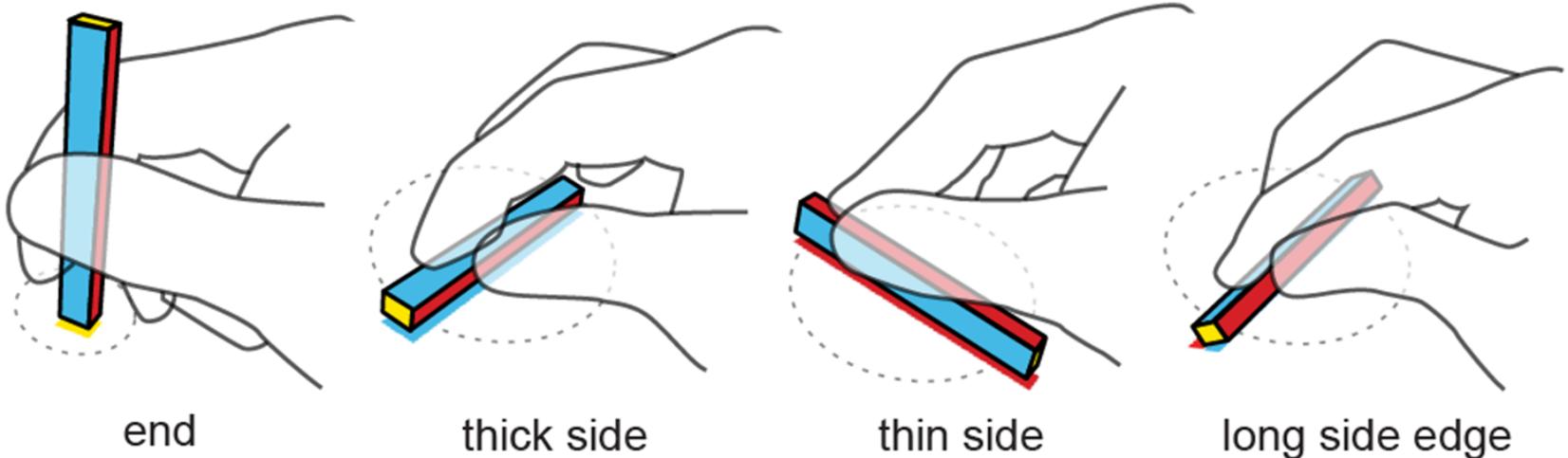
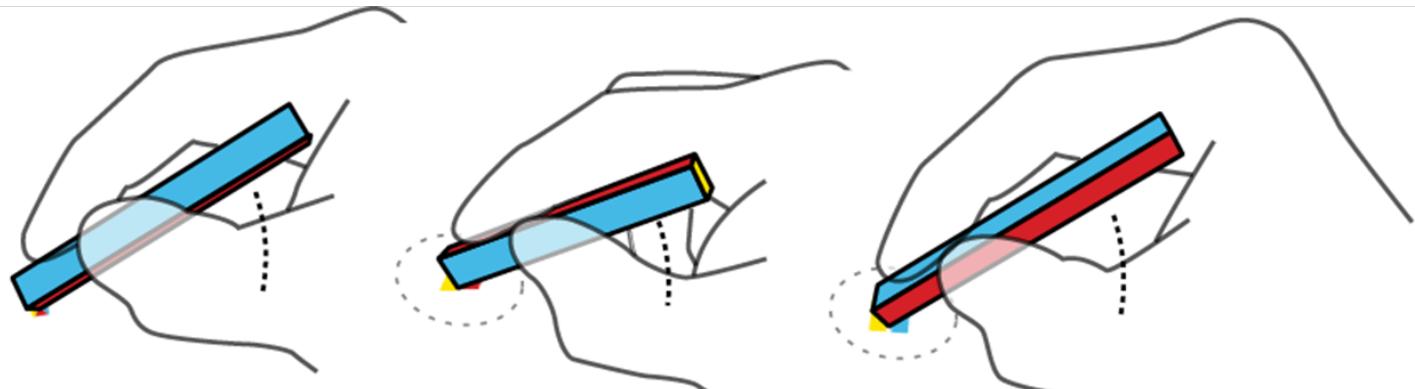
Conté

70



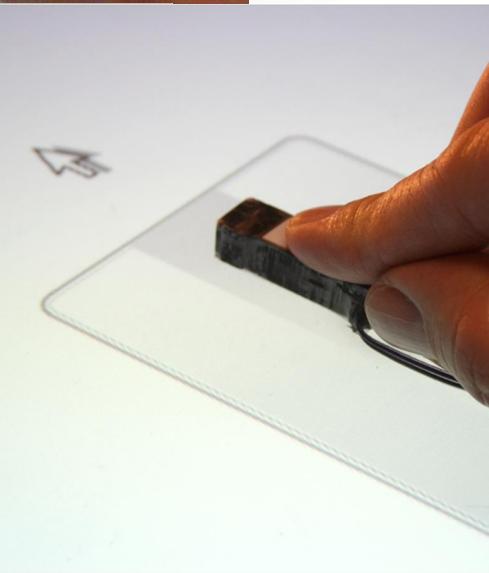
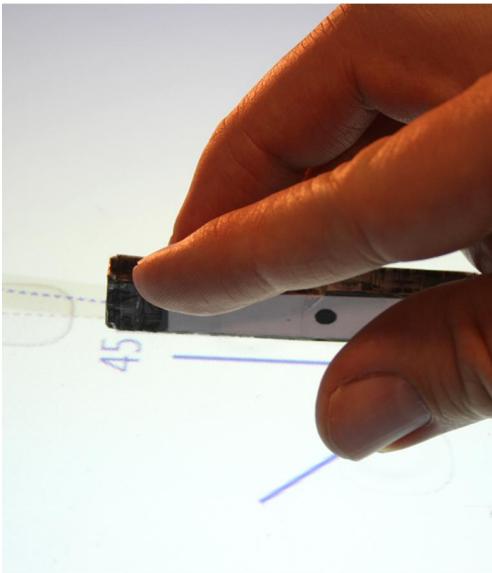
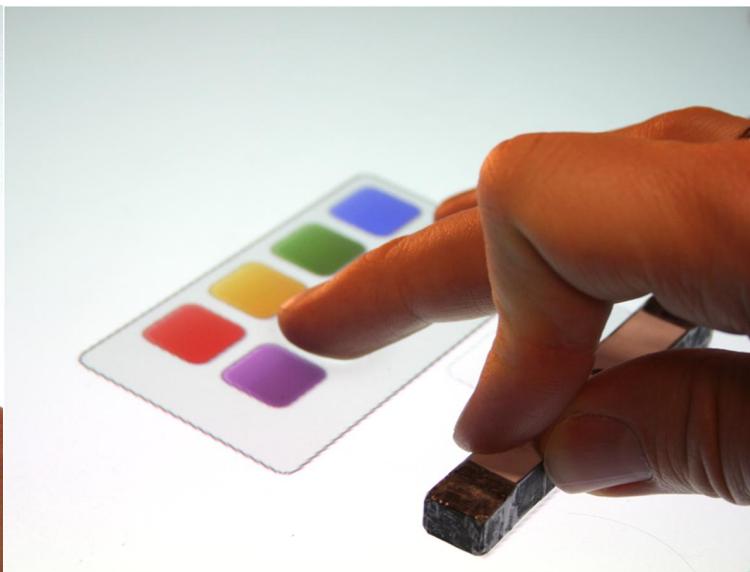
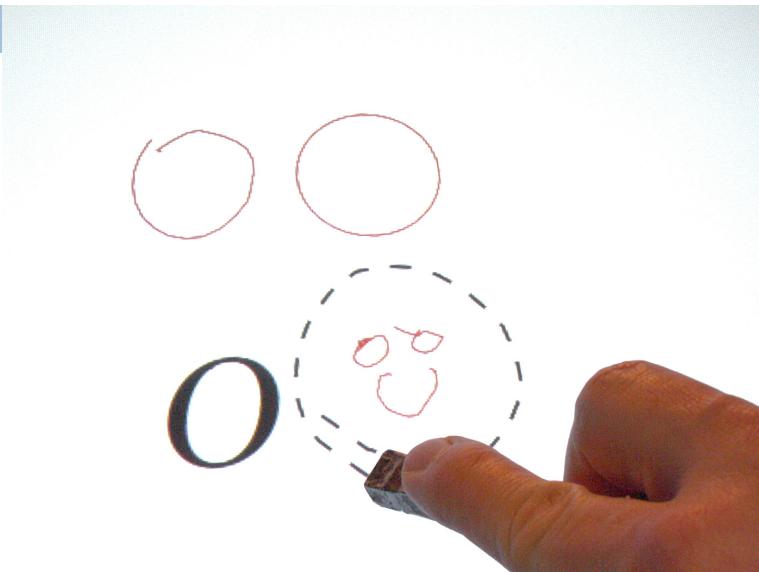
Conté

71



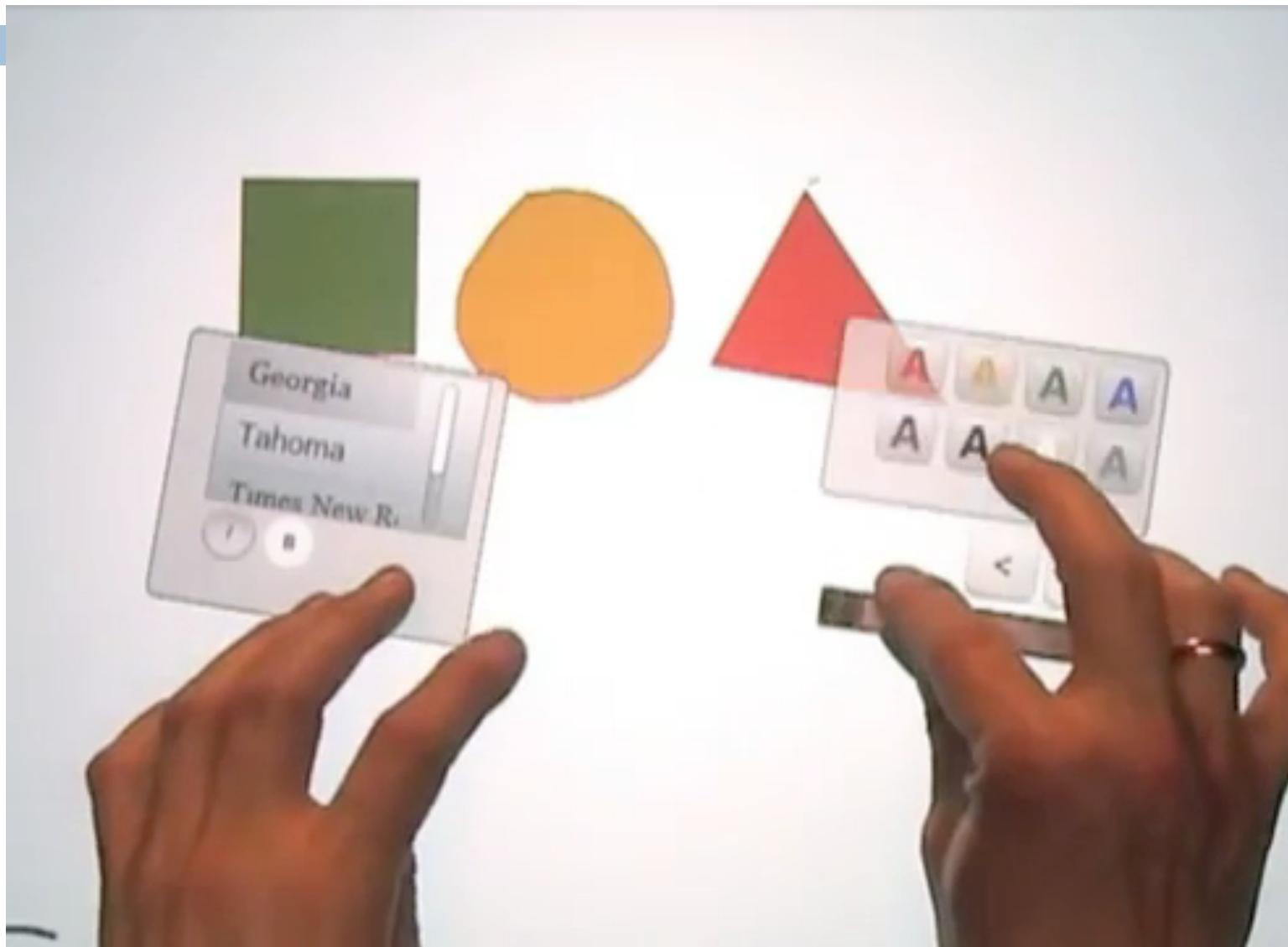
Conté

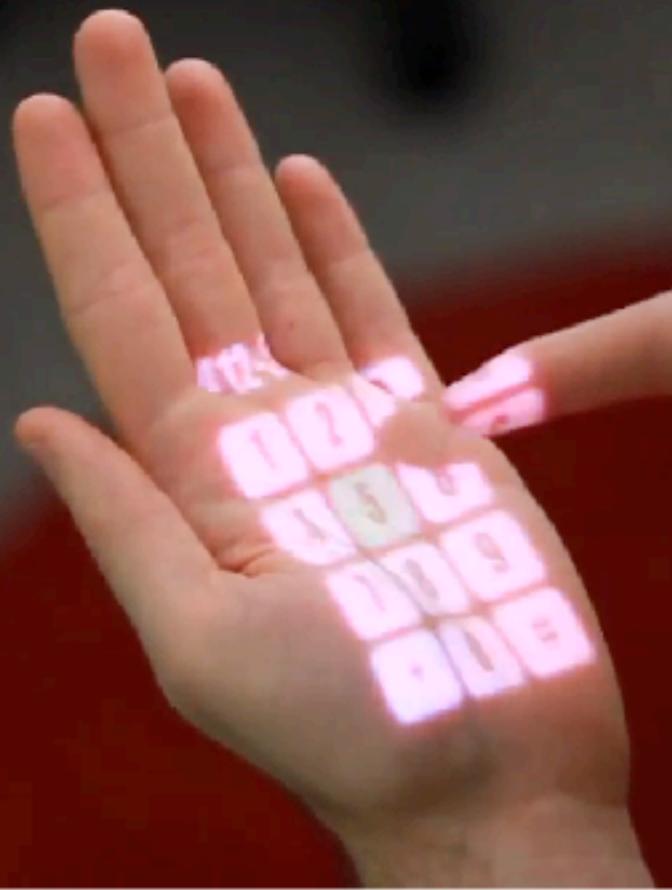
72

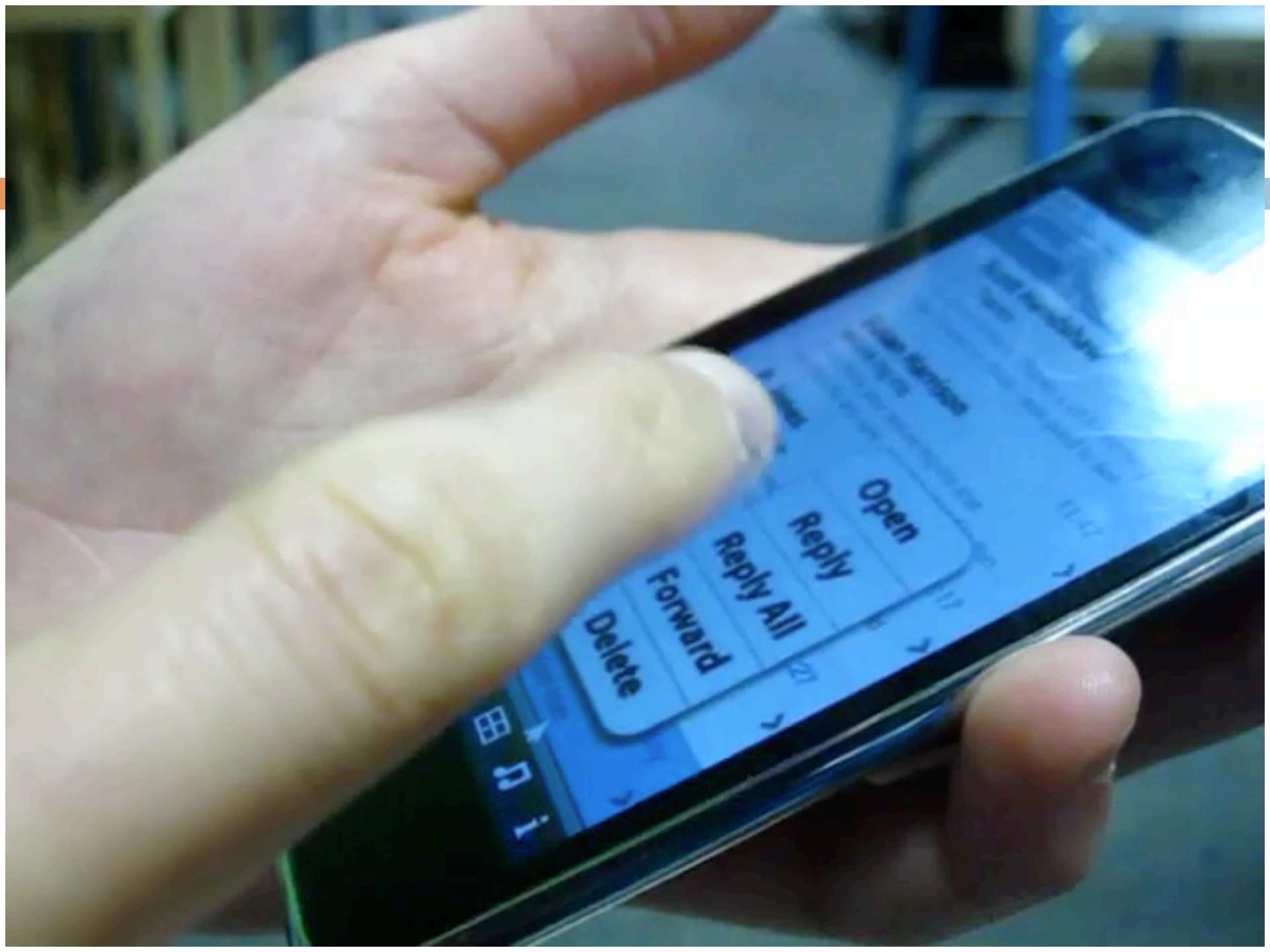


Conté

73







Open
Reply
Reply All
Forward
Delete

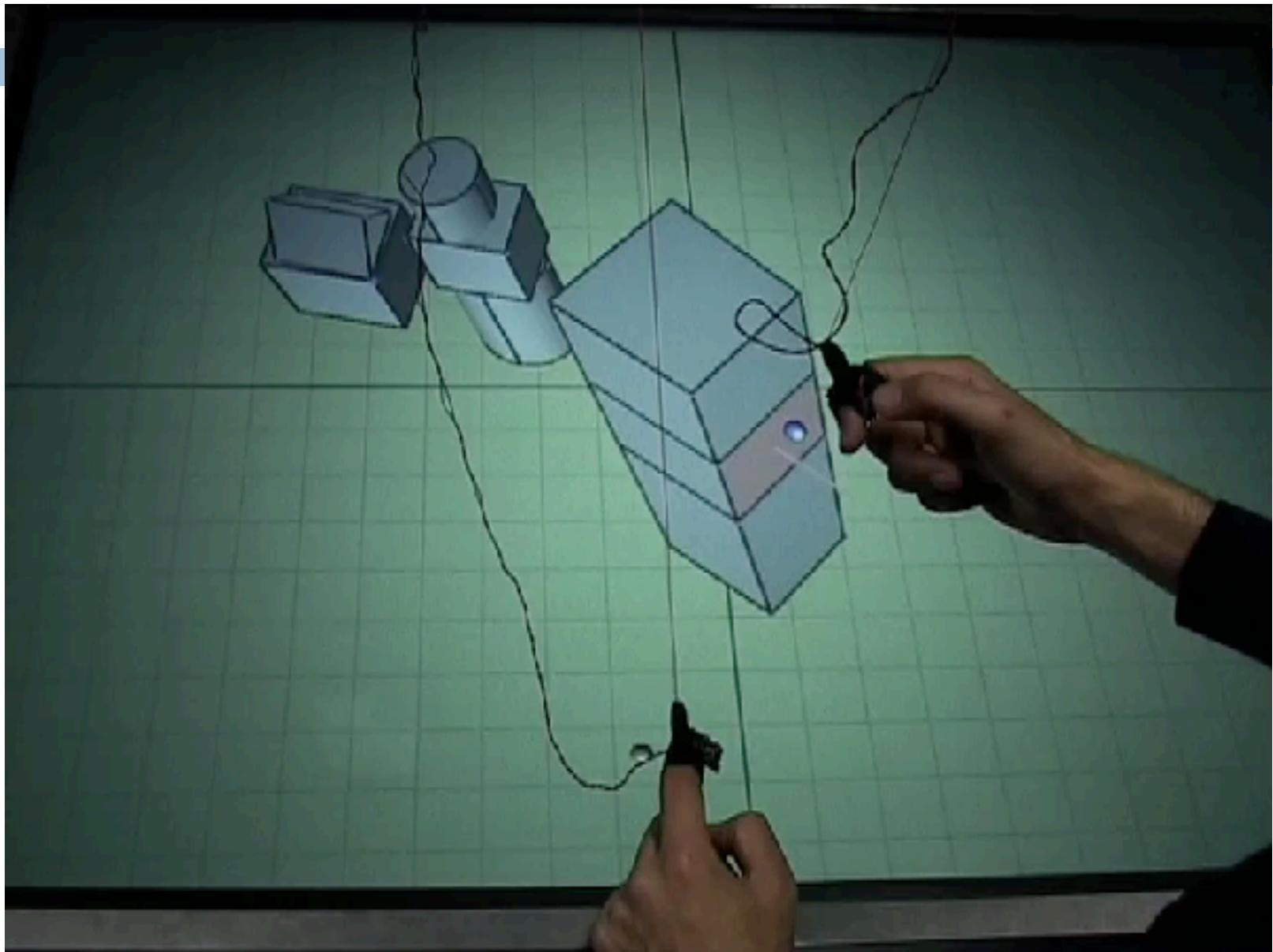
ForceEdge [Antoine et al. 2017]

76



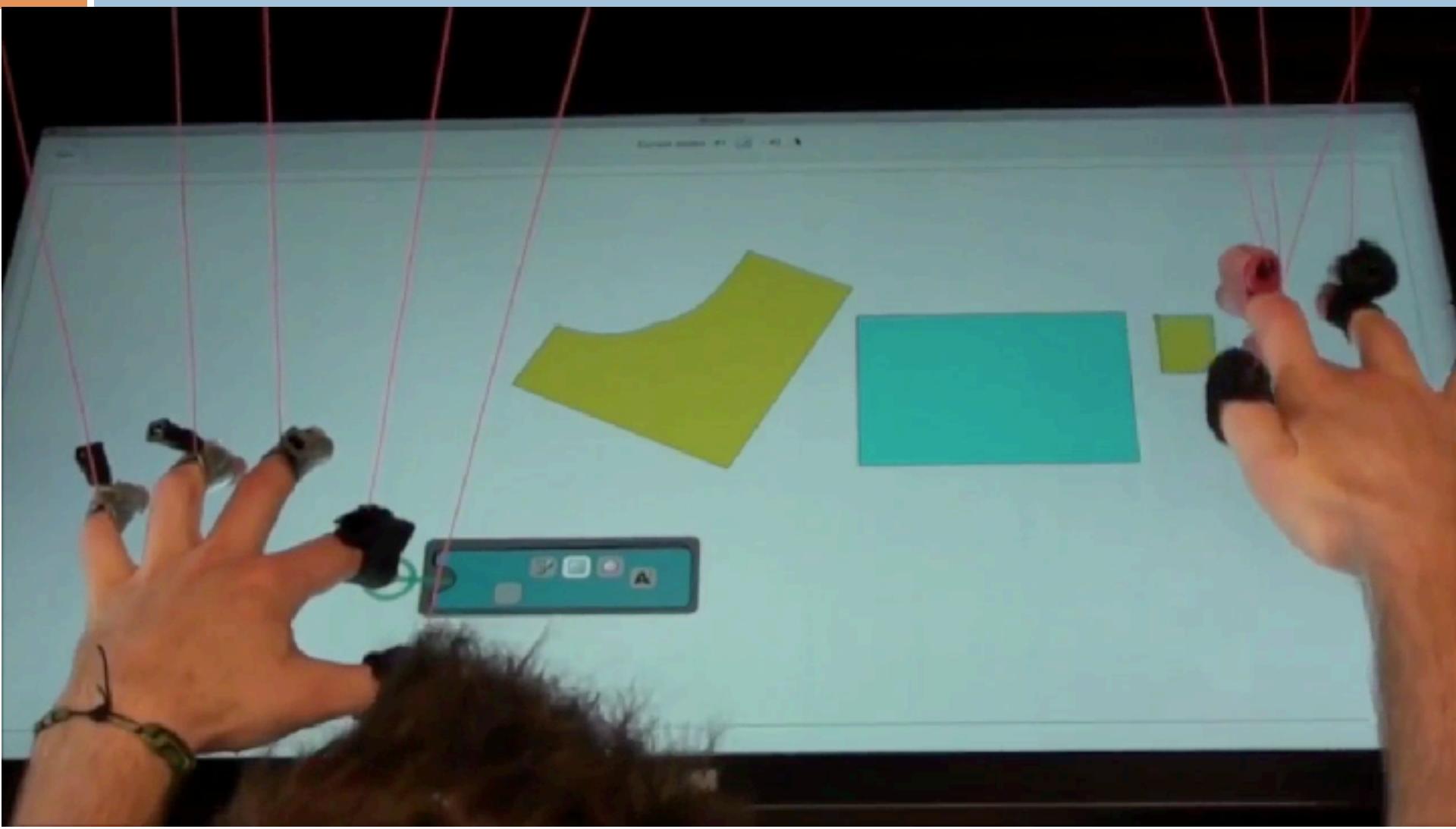
Mockup Builder [Araujo, Casiez et al. 2012]

77



FingerCuts [Goguey, Casiez et al.]

78



WhichFingers [Masson et al. 2017]

79

