ARCHIMEDES

SOFTWARE DEVELOPMENT FOR TABLETOPS

Ulrich von Zadow
Archimedes Solutions GmbH
uz@archimedes-exhibitions.de

AGENDA

- Multitouch APIs: Taxonomy of Features
- Overview of Some Current Multitouch APIs
- Interlude: Introduction to libavg
- Library Implementation
- Application-level Considerations

- Taxonomy: Classification of what the APIs provide
- Current APIs: Taxonomy gives us a tool for the evaluation. Too many APIs to evaluate all!
- libavg Intro: Perspective for the following parts – we implemented everything from scratch
- Library Implementation: Most of the time for the tutorial
- App-Level Considerations: Typical app lifecycle walkthrough

Multitouch API Taxonomy

Multitouch APIs - Low-Level Features

	TUIO			Microsoft						
	1	1.1	2	Native Win32 Touch	WPF Touch (4.0)	Surface SP1	Linux Kernel MT, libmtdev	PyMT	CCV	libavg
Basic Multitouch Detection	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Additional per-touch Data	N	ellipse data	ellipse data, geometry, pressure	only size	N	N	Y	Y	Y	Y
Tangible Support	Y	Y	Y	N	N	Y	N	(Y) - routes TUIO events	N	N
Raw Image	N	N	Y	N	N	Y	N	N	N	Y
Hover	N	N	negative pressure	N	N	N	N	N	N	Y
OS Independent	Y	Y	Y	N	N	N	N	Y	Y	Y
Open Source	Y	Y	Y	N	N	N	Y	Y	Y	Y

Multitouch APIs - High-Level Features

	Microsoft					
	Native Win32 Touch	WPF Touch (4.0)	Surface SP1	PyMT	GEIS	libavg
Event Bubbling & Capture	N	Y	Y	Y	N	Y
Enter/Leave Events	N	Y	Y	N	N	Y
Hand-Finger Correlation	N	N	N	N	N	Y
Gestures						
Basic Zoom/Pan/Rotate	Y				Y	
Integrated Zoom/Pan/Rotate	N	Y	Y	Y	N	Y
Other Gesture Support	Y	N	only tap & hold	?	Y	N
Inertia	only pan	Y	Y	N	N	Y
Widget Support	N	basic widgets, not multiuser	Y	Y	N	basic

-Too much for one slide.

-Things are in flux. New APIs are appearing etc.

-Distinction between Low- and High-Level features.

-From Bottom to Top:

- Closer -> further away from HW
- Older -> newer developments
- Clearer -> less clear requirements

Multitouch API Taxonomy

Multitouch APIs - Low-Level Features					
		TUIO			Micro
	1	1.1	2	Native Win32 Touch	W To (4
Basic Multitouch Detection	Y	Y	Y	Y	
Additional per-touch Data	N	ellipse data	ellipse data, geometry, pressure	only size	N
Tangible Support	Y	Y	Y	N	N
Raw Image	N	N	Y	N	N
Hover	N	N	negative pressure	N	N
OS Independent	Y	Y	Y	N	N
Open Source	Y	Y	Y	N	N

Multitouch API Taxonomy

Multitouch APIs - High-Level Features					
		Microsoft			
	Native Win32 Touch	WPF Touch (4.0)	Surface SP1	PyMT	GEI
Event Bubbling & Capture	N	Y	Y	Y	N
Enter/Leave Events	N	Y	Y	N	N
Hand-Finger Correlation	N	N	N	N	N
Gestures					
Basic Zoom/Pan/Rotate	Y				Y
Integrated Zoom/Pan/Rotate	N	Y	Y	Y	N
Other Gesture Support	Y	N	only tap & hold	?	Y
Inertia	only pan	Y	Y	N	N
Widget Support	N	basic widgets, not multiuser	Y	Y	N

-HW dependence: Wigdor's HW Taxonomy at last year's tutorial

Current Multitouch APIs

-Can't evaluate all APIs – too many available

-Selection: Industry standards + interesting developments

Current Multitouch APIs: TUIO

	TUIO		
	1	1.1	2
Basic Multitouch Detection	Y	Y	Y
Additional per-touch Data	N	ellipse data	ellipse data, geometry, pressure
Tangible Support	Y	Y	Y
Raw Image	N	N	Y
Hover	N	N	negative pressure
OS Independent	Y	Y	Y
Open Source	Y	Y	Y

-De-Facto industry standard.

-Network protocol, not library

-Originally for the reactable (Martin Kaltenbrunner, Ross Bencina et. al.), main feature was marker tracking.

-Based on OSC

-1.0: Basic touch data. Supported by/Wrappers exist for virtually all MT devices and tracking libraries.

-1.1: Reference implementation exists, but little library support. Adds ellipse data

-2.0: Looks very nice, but no implementation yet.

Current Multitouch APIs: TUIO

Some implementations

- iPhone
- Win7 Touch Wrapper
- CCV
- ...

Current Multitouch APIs: Microsoft

	Microsoft		
	Native Win32 Touch	WPF Touch (4.0)	Surface SP1
Basic Multitouch Detection	Y	Y	Y
Additional per-touch Data	N	N	N
Tangible Support	N	N	Y
Raw Image	N	N	Y
Hover	N	N	N
OS Independent	N	N	N
Open Source	N	N	N
Event Bubbling & Capture	N	Y	Y
Enter/Leave Events	N	Y	Y
Hand-Finger Correlation	N	N	N
Gestures			
Basic Zoom/Pan/Rotate	Y		
Integrated Zoom/Pan/Rotate	N	Y	Y
Other Gesture Support	Y	N	only tap & hold
Inertia	only pan	Y	Y
Widget Support	N	basic widgets, not multiuser	Y

- Native Win32: WM_TOUCH, WM_GESTURE.
  - Missing a good event model – no bubbling.
  - WM_GESTURE separates zoom, pan, rotate.
- WPF Touch:
  - Integrated into WPF event model
  - Very nice manipulation API (zoom/pan/rotate)
  - Single-user: Only one control can have focus

-Surface SP1: Similar features to WPF touch, adds multi-user widgets. Widgets are available for use with WPF touch, so usable with any HW.

Current Multitouch APIs: Linux

	Linux Kernel MT, libmtdev	GEIS
Basic Multitouch Detection	Y	N
Additional per-touch Data	Y	N
Tangible Support	N	N
Raw Image	N	N
Hover	N	N
OS Independent	N	N
Open Source	Y	Y
Event Bubbling & Capture	N	N
Enter/Leave Events	N	N
Hand-Finger Correlation	N	N
Gestures		
Basic Zoom/Pan/Rotate	N	Y
Integrated Zoom/Pan/Rotate	N	N
Other Gesture Support	N	Y
Inertia	N	N
Widget Support	N	N

- Kernel MT support:
  - Driver-level API (files + ioctls),
  - not easy to use,
  - ID tracking not guaranteed
  - not meant for direct use by applications.

-Libmtdev:
  - Layer above Kernel MT
  - Adds ID tracking
  - A little easier to use: function calls instead of ioctls.

-Xinput 2.1 extension:
  - Work in progress: 2 specs, 1 implementation

-GEIS: Gesture recognition framework. Heart of publicised Ubuntu 10.10 MT support.

Current Multitouch APIs: PyMT

Basic Multitouch Detection	Y
Additional per-touch Data	Y
Tangible Support	Y
Raw Image	N
Hover	N
OS Independent	Y
Open Source	Y
Event Bubbling & Capture	Y
Enter/Leave Events	N
Hand-Finger Correlation	N
Gestures	
Basic Zoom/Pan/Rotate	
Integrated Zoom/Pan/Rotate	Y
Other Gesture Support	?
Inertia	N
Widget Support	Y

-Python-based library

-Mostly implemented in python

-Interfaces to several low-level libraries.

-Large widget library – couldn't test the quality of the widgets.

Current Multitouch APIs: CCV

Basic Multitouch Detection	Y
Additional per-touch Data	Y
Tangible Support	Y
Raw Image	N
Hover	N
OS Independent	Y
Open Source	Y

-One of many tracking libraries that deliver TUIO 1.0 events, but:

-Mature & well maintained

-Nice configuration interface

-Large user base (NUIGroup)

Current Multitouch APIs:libavg

Basic Multitouch Detection	Y
Additional per-touch Data	Y
Tangible Support	N
Raw Image	Y
Hover	Y
OS Independent	Y
Open Source	Y
Event Bubbling & Capture	Y
Enter/Leave Events	Y
Hand-Finger Correlation	Y
Gestures	
Basic Zoom/Pan/Rotate	
Integrated Zoom/Pan/Rotate	Y
Other Gesture Support	N
Inertia	Y
Widget Support	basic

-Framework with a Python API

-Lots of green ☺

libavg

"high-level development plattform for media-centric applications"

libavg

"high-level development plattform for media-centric applications"

- Screen layout (xml) + interaction (python)

libavg

High-level development plattform for media-centric applications

- Screen layout (xml) + interaction (python)
- 2D Scenegraph

-Like a 3D engine, just for two dimensions.

libavg

- Open Source, LGPL License
- Runs on Mac, Linux, Windows

- Started in 2003
- Since 2007: Development sponsored by Archimedes

- LGPL: Usable for commercial applications. If you add things to the core library, please publish the changes.
- Development at Archimedes: Currently 10 developers across all plattforms
- Currently in the process of delivering a museum with > 70 different exhibits running libavg, including:
    - eye tracking
    - multi-projector blending on curved surfaces
    - Robot terrarium: Real robots, virtual obstacles
    - Of course: Multitouch tables

libavg

Code Examples

libavg

Layout elements (nodes):

- Images
- Text
- Videos
- Camera Images
- Vector Graphics
- Plugins

-Show twovideos.py, start it

-Show videochooser.py, start it

libavg

Layout engine:

- Structure through nodes
- Pos, scale, rotate nodes
- Transparency
- Mask images
- Cropping
- Offscreen buffers (FBO)

libavg

Implementation:

- C++
- OpenGL backend
- Uses:
  - ffmpeg
  - Pango
  - GraphicsMagick
  - libSDL
  - ...

-All compositing is done on the graphics card

libavg

Text formatting:

- Paragraph-level layout (like)
- Full utf8 support: Japanese, Chinese, Arabic, Latin (etc) scripts.

-Utf8: Japanese version of Biodiversity Session Desk: Japanese, Chinese, Latin in one paragraph

libavg

Videos:

- Alpha channel support
- Smaller videos (100x100 pixels incl. alpha): 200 at 30 Hz on a Core i7.
- Latency-free seeking
- Fast: Shaders, multithreading, sse2 intrinsics...

-Alpha channel: No performance penalty

-Seek: single-threaded mode that decodes everything synchronously. Can seek at 30 Hz in 720p videos

-Rotating earth in biodiversity: Video seeking

libavg

Videos: Sample code

-~/devel/tmp/alphavideos/alphavideos.py

libavg

Plugins:

- New node types
- Extend any other part of libavg
- Written in C++

-New Node types: 3D rendering (OpenSceneGraph)

-Custom Windchannel rendering for Computer Sport

-Other plugins: fiducial tracking, eye tracker

libavg

Multitouch driver model:

- TUIO
- Apple Trackpads
- Our own tracker
- (Half-finished linux mtdev driver)

libavg Ecosystem

- HTW Berlin
- Saar University
- XIBO
- c-base

-Where else is libavg being used?

-HTW Berlin: Snatch‘em

-Saar University: Just did a Seminar/student project

-XIBO: Open Source Digital Signage (web browser plugin)

-c-base: Hackerspace in Berlin,

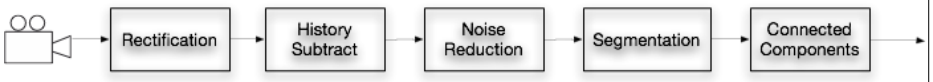
- Original multitouch garage project
- Several libavg installations
- Multitouch hackfest last december

Library Implementation

-Go through all the api features and see how they work internally

-Tip if you don‘t want to do everything by hand: Use OpenCV.

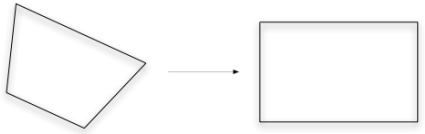
FTIR Tracking



-FTIR explained in Florian‘s tutorial.

-Standard tracking alg., basic CV.

FTIR Tracking: Rectification



Intrinsic Parameters:

- Barrel/pincushion distortion

Extrinsic Parameters:

- 3D perspective transform

Rectification: Turning the camera image into a rectangle ☺.

FTIR Tracking: Rectification

Calibration:

1. User touches lots of points with known positions.
2. Solve overdetermined system with camera params as unknowns.

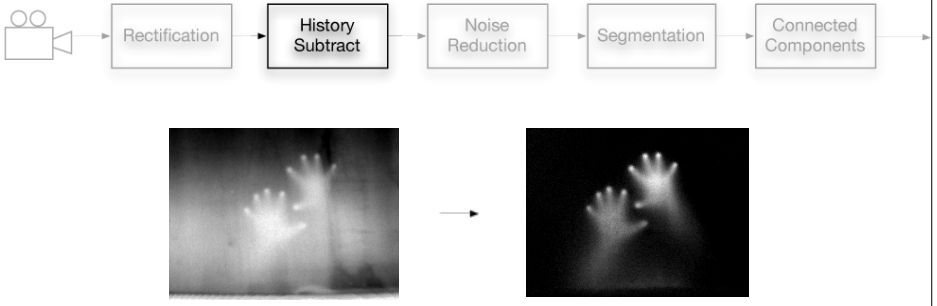
Safer convergence: Solve for intrinsic params first, then for extrinsic params.

[Tsai 87, "A Versatile Camera Calibration Technique...."]

-Overdetermined system: Least squares.

-Intrinsic first: Unknowns are

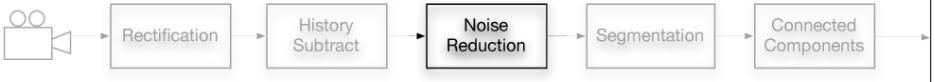
FTIR Tracking



-Continuous, but very slow subtraction.

-Initial history: Not just one frame

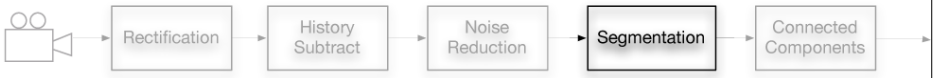
FTIR Tracking



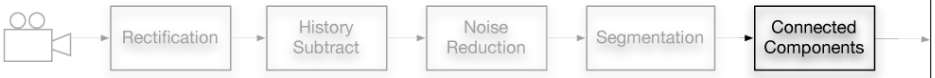
-Camera Noise is salt & pepper noise, so median filter should fit.

-But: Too aggressive in our case. Very small gauss blur helped.

FTIR Tracking



FTIR Tracking

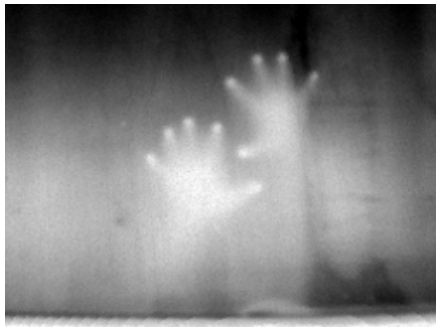


FTIR Tracking

Segmentation:

```
for each line:
   find all runs of pixels > threshold
   for each run:
      if underneath run in prev line:
         append to blob
         if underneath more runs:
            join blobs
      else:
         create new blob
```


DI Tracking

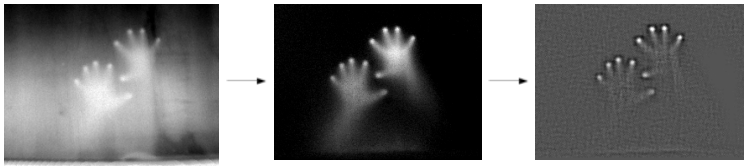


-Bad camera images!

-Bright points where fingers aren‘t touching, so threshold isn‘t enough.

-Solution: Highpass filter

DI Tracking



Camera Image History Subtracted Highpass

-Bad camera images!

-Bright points where fingers aren‘t touching, so threshold isn‘t enough.

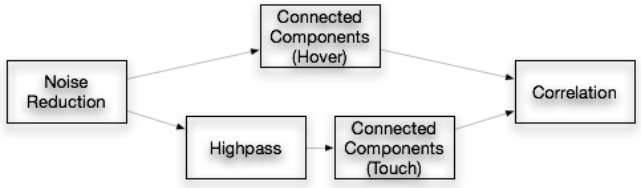
-But: the diffuser (aka projection surface) distributes light evenly, so only objects that are touching the surface are in focus.

-So use highpass filter (image minus gauss blur of image)

-Or edge detection filters

-Archimedes Session Desk: Tempered glass, 6mm between ‚touch‘ and surface, wide highpass -> GPU

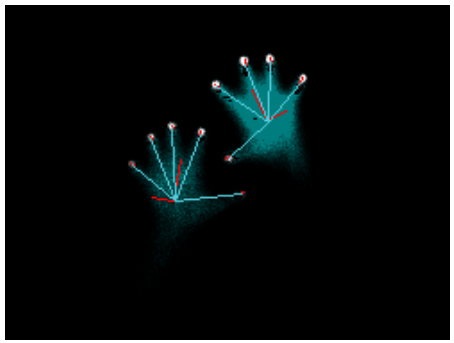
DI Tracking: Hover



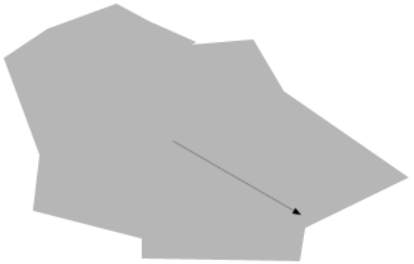
-To find hands above the surface, simply run the standard algo w/o highpass

-Touches can be correlated to hands: fingers are inside of hands

DI Tracking: Hover



Blob Statistics: PCA



-PCA: Principal Components Analysis.

-Delivers an ellipse that approximates the blob.

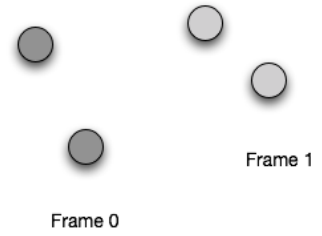
-Center of the blob is average of all pixel coordinates of the blob

-First ellipse axis is the axis with the greatest variance

-Can be derived analytically, involves covariance matrix, eigenvalues etc. and can be found in any good statistics book.

-Why? -> Object classification, Hand direction,...

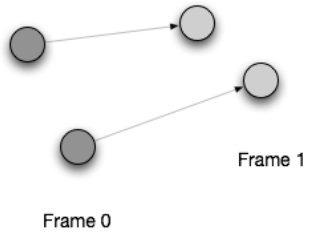
ID Tracking



-Correspondence of blobs from one frame to the next.

-Theoretical solution: Euclidian Bipartite Matching

ID Tracking



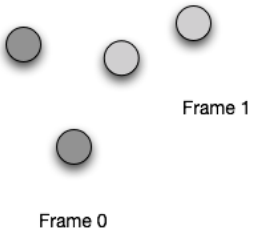
Euclidian Bipartite Matching

-Naive algorithm runs in O(n3) or so...

-Lots of papers on optimization, but

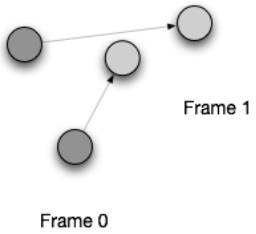
-They miss the real problem

ID Tracking



-Not enough knowledge to find the solution!

ID Tracking



-Wrong decision -> Picture viewer does an unintentional 180° turn of the image!

ID Tracking: Dead Reconing

ID Tracking: Dead Reconing

[Ridley, Mark, 1613. “A short treatise of magneticall bodies and motions”]

CHAP. XXXXII.

Of finding the variation of the Inclina- tory-needle.

HAuing declared the manifold cauſes of the variation of the *Inclinatory-needle* from his true and naturall angle of inclination with the horizon; it followeth that we deliuer plainely how to diſcerne theſe euents when they ſhall fall out, which are known perfectly by hauing the true eleuation of the pole, where theſe matters ſhall be required.

The rules and wayes how to know the eleuation of the pole in all places, is ſo perfectly ſet out by moſt Aſtronomers and writers of Nauigation, and ſo well knowne to all expert Pilots, that it is a thing needleſſe here to repeate them.

Beſides the ingenious Pilot knowing the eleuation of the pole in ſome places of his voyage that he hath paſſed, by keeping a true, not a dead reckoning of his courſe in pricking his Card aright, and obſeruing the way with the logge-line, with other currants and occurrants, will giue a very artificiall coniecture of the eleuation of the pole in that place where he is, though he ſee neither Sunne nor Starres.

CHAP. XXXXII.

Of finding the variation of the Inclina- tory-needle.

HAuing declared the manifold cauſes of the variation of the *Inclinatory-needle* from his true and naturall angle of inclination with the horizon; it followeth that we deliuer plainely how to diſcerne theſe euents when they ſhall fall out, which are known perfectly by hauing the true eleuation of the pole, where theſe matters ſhall be required.

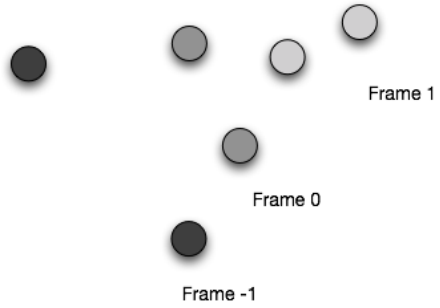
The rules and wayes how to know the eleuation of the pole in all places, is ſo perfectly ſet out by moſt Aſtronomers and writers of Nauigation, and ſo well knowne to all expert Pilots, that it is a thing needleſſe here to repeate them.

Beſides the ingenious Pilot knowing the eleuation of the pole in ſome places of his voyage that he hath paſſed, by keeping a true, not a dead reckoning of his courſe in pricking his Card aright, and obſeruing the way with the logge-line, with other currants and occurrants, will giue a very artificiall coniecture of the eleuation of the pole in that place where he is, though he ſee neither Sunne nor Starres.

V 2 Notwith-

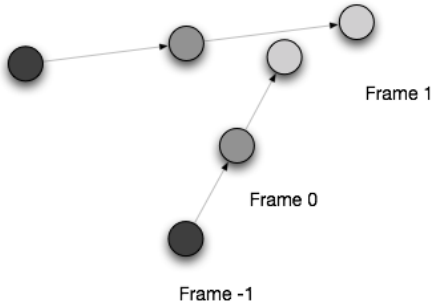
© ARCHIMEDES I NOVEMBER 201

ID Tracking: Dead Reconing



-Take into account previous positions
-Determine where the finger should be
-Take that as basis for the matching algorithm

ID Tracking: Dead Reconing



© ARCHIMEDES I NOVEMBER 2010

-Take into account previous positions
-Determine where the finger should be
-Take that as basis for the matching algorithm

ID Tracking: Simple Matching Algorithm

```
Calculate all blob distances, put into heap structure
While heap not empty:
   Match blobs with the shortest distance
```

Easy optimization: Ignore everything over a maximum distance.

-With dead reconing, a simplified approximate matching algorithm works

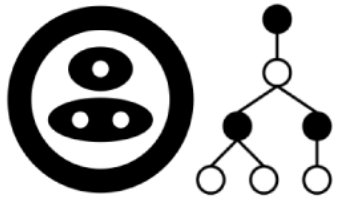
-Fingers have a maximum distance they can travel in a frame

Object Tracking

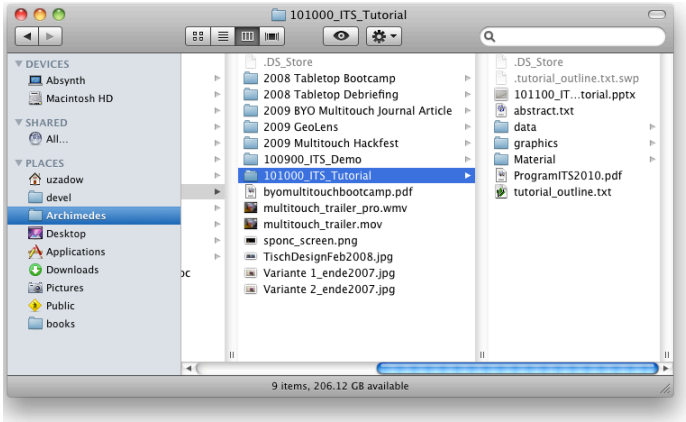
- Markerless: Use Values from PCA
- Marker-based: E.g. fiducials

-If the objects have different forms: use values from PCA to categorize them

Object Tracking: Fiducials



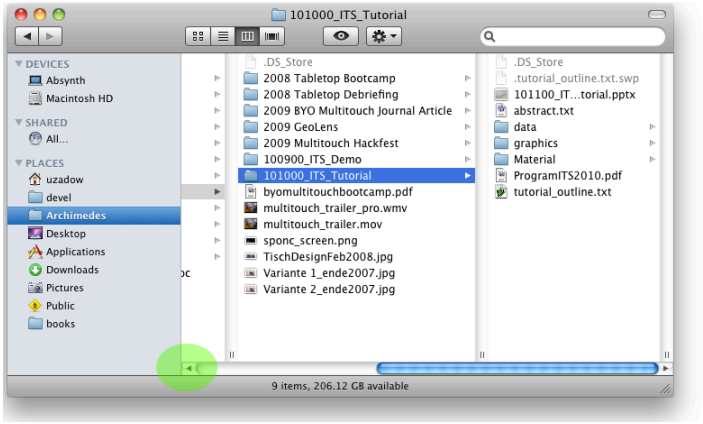
Event Bubbling



-Blob tracking gives tree of blobs.

-The tree topology is used as unique ID.

Event Bubbling



-Time-honored interface: UI elements form a hierarchy

- In this case: button->scrollbar->window

-Events travel up the hierarchy

- Button can have event handler, then scrollbar, etc.

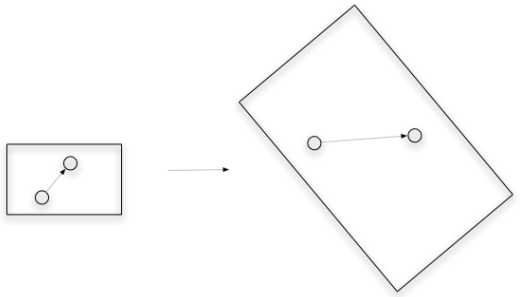
-Old, but stays useful for MT.

Event Capture

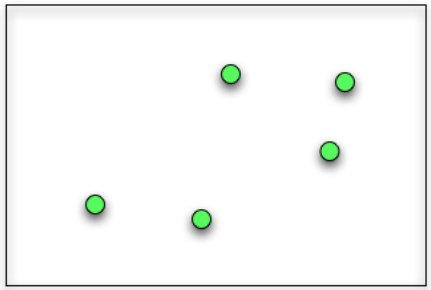
-Show os window close button

-Event capture: Applications can request that events from a contact be sent to a specific UI element, even if the contact leaves the element.

Direct Manipulation



Direct Manipulation: k-means



-1 Finger: Translation only

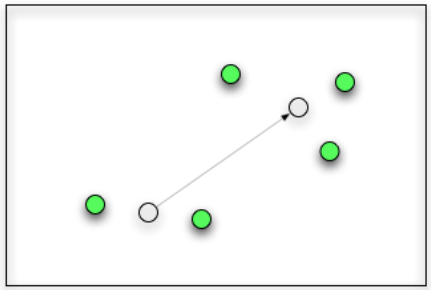
-2 Fingers: Simple 2D affine transform

- >2 Fingers: hm.

Direct Manipulation: k-means

```
p1, p2 = two of the touch points
repeat
   group points by distance to p1 or p2
   p1, p2 = centers of groups
until p1, p2 stops changing
```


Direct Manipulation: k-means



Direct Manipulation: k-means

When fingers appear/disappear:

```
calc k-means with new positions of old fingers
move object accordingly
calc new basis for transforms w/o moving
```


-Result: If no finger moves but a new one appears, the object doesn‘t move.

Direct Manipulation: Inertia

Extremely simple ‘Physics’:

- Calculate current speed on up event
- Decelerate

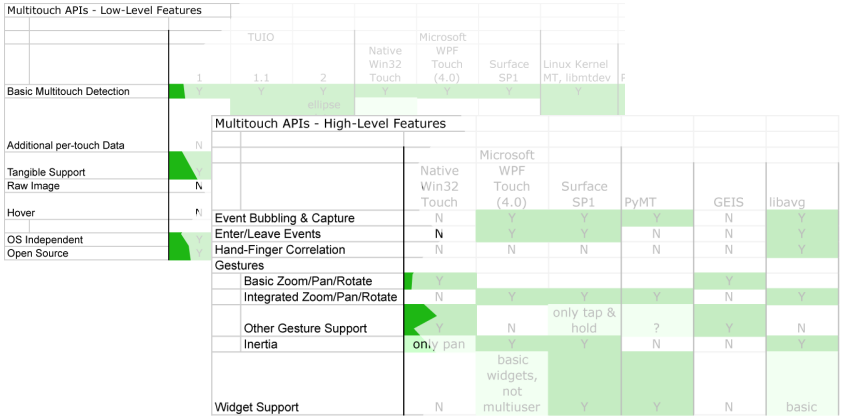
-To determine the initial speed, average several frames to reduce jitter

Library Implementation

Done :-)

-To determine the initial speed, average several frames to reduce jitter

Library Implementation



-This is one way to implement things – there are lots of alternatives

-

APPLICATION-LEVEL CONSIDERATIONS

-Archimedes builds science museums.

-Interesting experiences with multitouch ‘in the wild’

MUSEUM SETTING

A science museum is a place to learn things:

- Content must be scientifically sound
- Communication with scientists

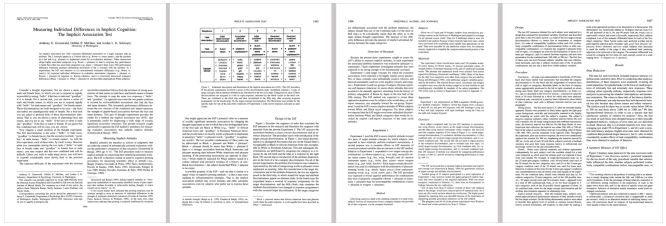
- Scientifically verifiable content

- Scientific communication: precise, verbose, peer audience (other scientists)

MUSEUM SETTING

Example: Scientific communication

[Tsai 87] R. Tsai. "A Versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-The-Shelf TV Cameras and Lenses." IEEE Journal of Robotics and Automation 3:4 (1987), 323–344.



- Scientifically verifiable content
- Scientific communication: precise, verbose, peer audience (other scientists)

MUSEUM SETTING

People don't stay forever:

- 2-3 hours per exhibition
- 2-5 *minutes* per exhibit!

- Scientists often aren't trained to explain to a layman.
- Is it *possible* to explain the paper on the last slide in 2-5 minutes?
- Tendency to cram too much data into exhibits

COPING WITH SHORT DWELL TIMES

- Very short learning phase, so:
  Self-evident interaction techniques
- Interaction often on a visceral, intuitive, emotional level
- Long text is almost always out of place

-Problem: Self-evident but still novel, attention-grabbing, etc...

INTERACTION DESIGN FOR MUSEUMS

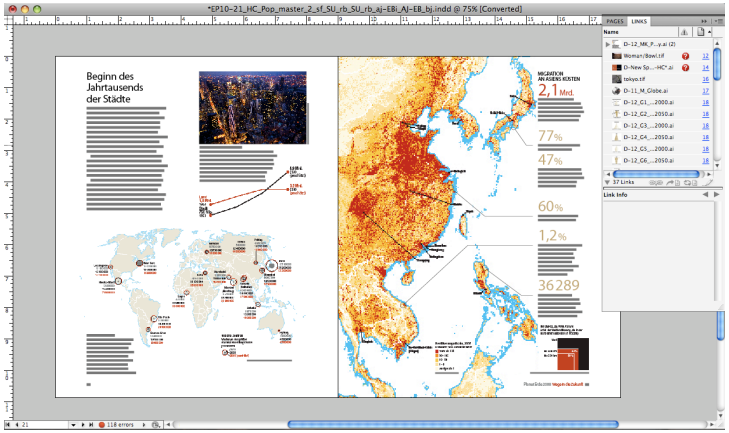
Example workflow: GlobalData exhibit in the Science Express Germany

Initial constraints:

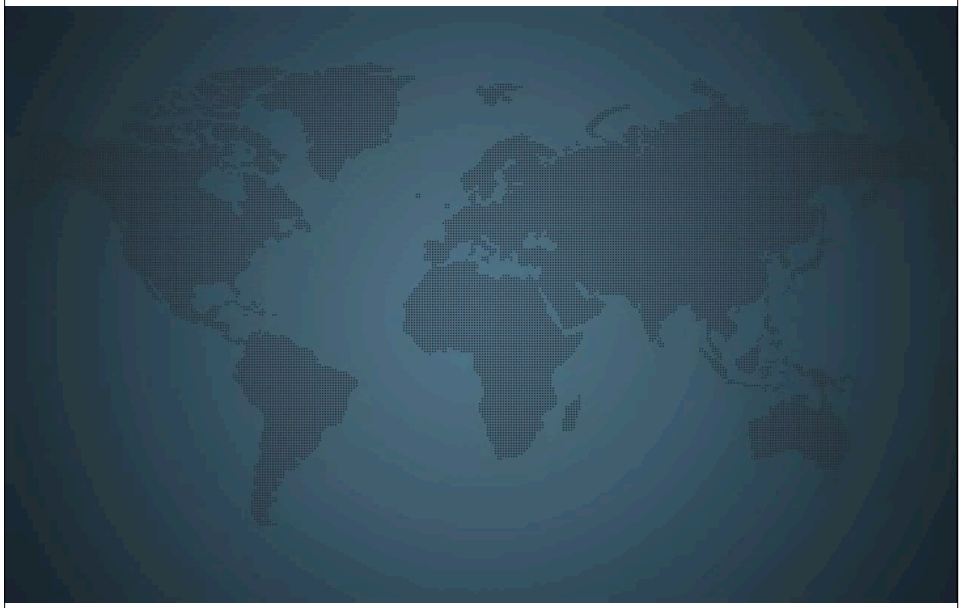
- Use Session Desk
- Use data from National Geographics
  (2008 special issue on Planet Earth)

- Demo tomorrow evening

THE GLOBALDATA EXHIBIT



So we had 50 pages of indesign documents from this NG issue.



Initial application sketch – a short movie showing the interaction

This was for a repeat customer, so there was a lot of trust and we basically got a ‘go’ at this point.

Found additional data: Per-country data on World population 1950-2050.

Show live application.

THE GLOBALDATA EXHIBIT

Study in the train:

- Collaboration with Johannes Schöning, Florian Daibler and Antonio Krüger, Saar University
- Observed interactions with this exhibit for a day

THE GLOBALDATA EXHIBIT

Visitors:

- Wide age span of visitors
- Group sizes 1-5 + some school classes

Show live application.

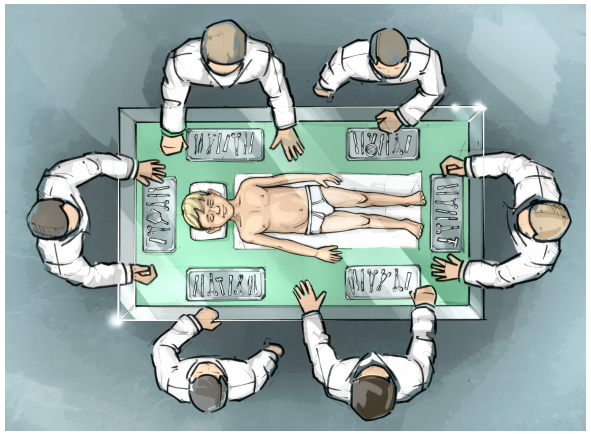
THE GLOBALDATA EXHIBIT

Interaction:

- Interaction time 5 secs to 10 mins, mean 3 mins
- Lots of collaboration
- 1/3 didn't realize the exhibit was interactive
- 1/3 used only one finger
- Of the ones using one finger: 50% didn't use drag
- No differences due to age
- But: Most users interacted with actual content

- As soon as one visitor started interacting, others joined
- One finger: handbags/coats!
- Age: Younger visitors->more experimentation, older ones -> more observation
- Short explorative period, presses on non-interactive elements etc., then longer interaction with the actual content

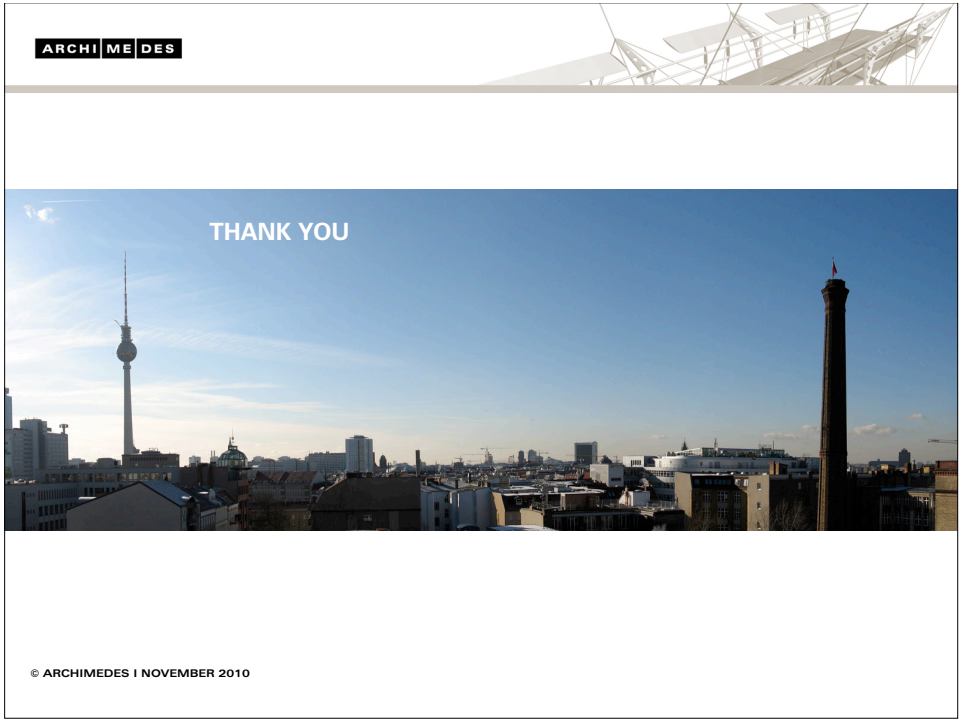
SimMed



Snatch'em

- As soon as one visitor started interacting, others joined
- One finger: handbags/coats!
- Age: Younger visitors->more experimentation, older ones -> more observation
- Short explorative period, presses on non-interactive elements etc., then longer interaction with the actual content

Personal space issues!



ARCHIMEDES