Vorlesung
Advanced Topics in HCI
(Mensch-Maschine-Interaktion 2)

Ludwig-Maximilians-Universität München
LFE Medieninformatik
Heinrich Hußmann & Albrecht Schmidt
WS2003/2004
http://www.medien.informatik.uni-muenchen.de/

Chapter 2: Information Visualization

Fisheye Views

Principles, Applications and Programming

Heiko Drewes

The Fisheye View Metaphor

The **fisheye view** is a metaphor coming from the fisheye lens used in photography. Such a wide angle lens distorts an image in the way that things in the central area appear enlarged, while things aside appear small.

Taken from the internet: www.rolfwegst.com

The idea behind the fisheye is enlarging the focus and keeping the context.

The Fisheye View Metaphor

In many contexts, humans often represent their own "neighborhood" in great detail, yet only major landmarks further away.
(George W. Furnas - CHI 1986)

The fisheye metaphor is more than a distortion of an image to display. It can be applied to many fields – networks, hierarchical structures.
All you need is a metric/context/distance function, that means something that tells whether another object is far or near.

The Fisheye View Theory

(George W. Furnas - CHI 1986)

Degree of interest (DOI) function:

$$DOI(a|.=b) = API(a) - D(a,b)$$

DOI(a|.=b): DOI of a, given the current focus is b.
API(a): static global a priori importance measure.
D(a,b): distance between a and b.

The Fisheye View Theory

```
(a) Distance from y:
dtree(x,y)
                  ____________________3____________________
                  |                   |                   |
        ______4_______      ______4_______      _______2_______
        |     |     |       |     |     |       |     |      |
      __5__ __5__ __5__   __5__ __5__ __5__   __1__ __3__  __3__
      | | | | | | | | |   | | | | | | | | |   | | | | | |  | | |
      6 6 6 6 6 6 6 6 6   6 6 6 6 6 6 6 6 6   0 2 2 4 4 4  4 4 4
                                                y
                                         "current focus"
(b) A Priori Importance in the tree:
Imp(x) = – dtree(x,root)
                                      root
                  ____________________0____________________
                  |                   |                   |
        ______-1_______     ______-1_______     ______-1_______
        |      |      |     |      |      |     |      |      |
      _-2__  _-2__  _-2__ _-2__  _-2__  _-2__ _-2__  _-2__  _-2__
      | | |  | | |  | | | | | |  | | |  | | | | | |  | | |  | | |
    -3-3-3 -3-3-3 -3-3-3 -3-3-3 -3-3-3 -3-3-3 -3-3-3 -3-3-3 -3-3-3
                                                y
                                         "current focus"
(c) The Fisheye DOI:
DOIfisheye(tree) (x|.=y) = API(x) – D(x,y)
= –( dtree(x,y) + dtree(x,root) )
                                      root
                  ____________________-3____________________
                  |                   |                   |
        ______-5_______     ______-5_______     ______-3_______
        |      |      |     |      |      |     |      |      |
      _-7__  _-7__  _-7__ _-7__  _-7__  _-7__ _-3__  _-5__  _-5__
      | | |  | | |  | | | | | |  | | |  | | | | | |  | | |  | | |
    -9-9-9 -9-9-9 -9-9-9 -9-9-9 -9-9-9 -9-9-9 -3-5-5 -7-7-7 -7-7-7
                                                y
                                         "current focus"
Figure 1. Distance, A Priori Importance and the Fisheye DOI for a rooted tree.
```

(George W. Furnas - CHI 1986)

Example for DOI function applied on a tree structure

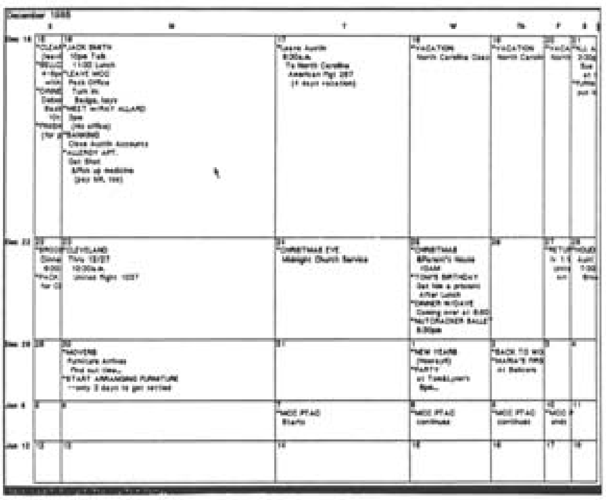
This ideas are from 20 years ago. At that time computational graphics power was very limited.

Take this as inspiration for your ideas now.

The Fisheye View Theory

(George W. Furnas -CHI 1986)

A Fisheye Calendar.



The Fisheye View Theory

Y. K. Leung, M. D. Apperley (1994) A Review and Taxonomy of Distortion-Oriented Presentation Techniques

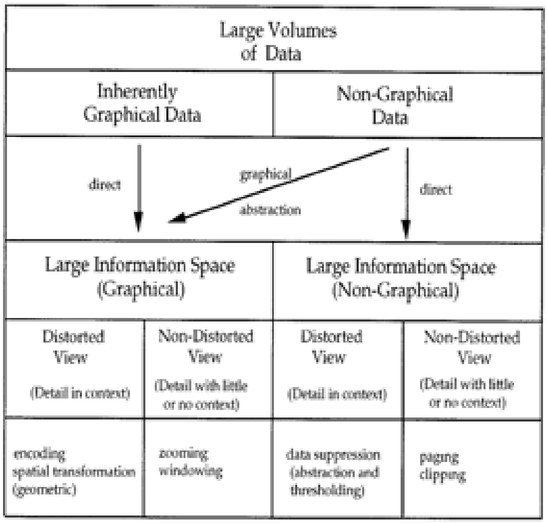
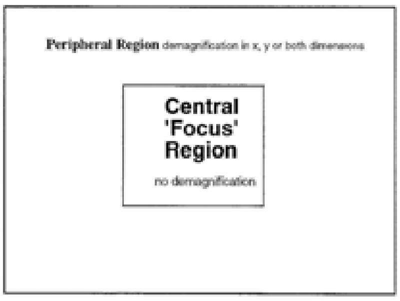


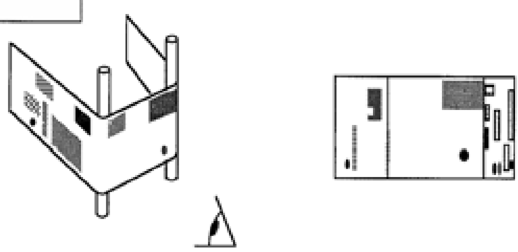
Fig. 1. A taxonomy of presentation techniques for large graphical data spaces.

The Fisheye View Theory



(Y. K. Leung, M. D. Apperley 1994)

Metaphor of a perspective wall

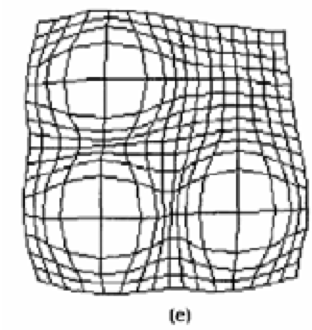


(a) (b)

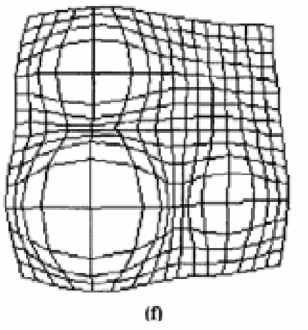
The Fisheye View Theory

Unified theory of distortion techniques (Y. K. Leung, M. D. Apperley 1994)

- "…stretchable rubber sheet mounted on a rigid frame"
- Stretching = Magnification
- Stretching one part must equal shrinkage in other areas



Multi focal projections



Fisheye Views Applications

- Semantic fisheyes
- 1-dimensional fisheyes
- 2-dimensional fisheyes
- Fisheyes for precise input

1-dimensional Fisheye

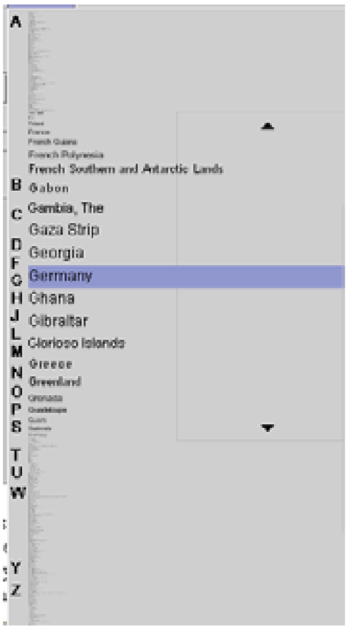
- Time axis
  - historical calendar
  - story line
- Menus

1-dimensional Fisheye

Example: Fisheye Menu

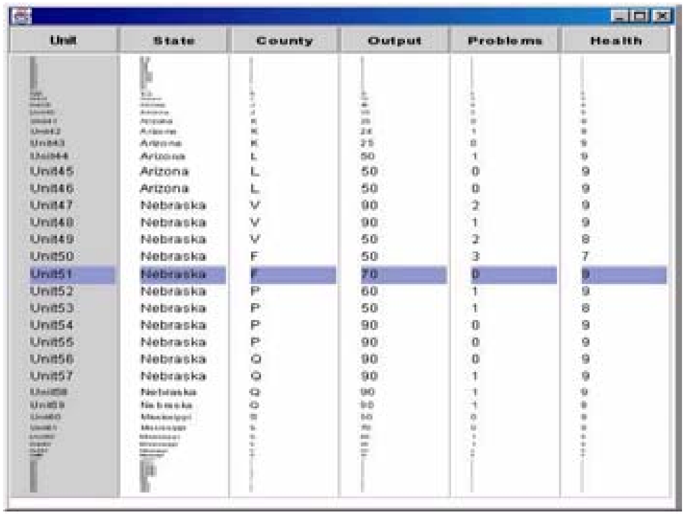
Benjamin B. Bederson.
Fisheye Menus. UIST'00

http://www.cs.umd.edu/hcil/fisheyemenu/fisheyemenu-demo.shtml



1-dimensional Fisheye

Fisheye Table



2-dimensional Fisheye

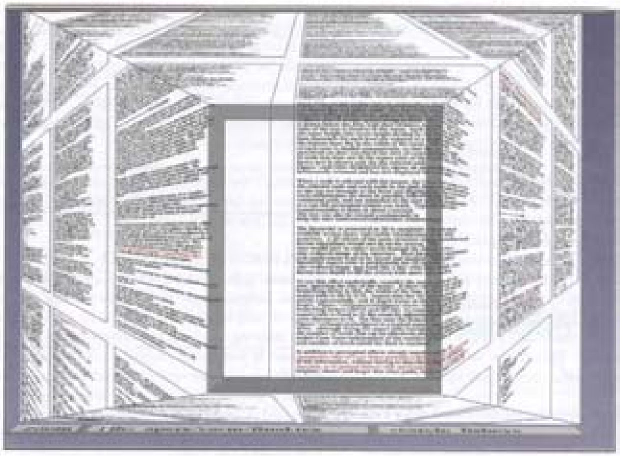
- Typically surfaces
  - geographical/topological data i.e. maps
  - desktop

Fisheye views controlled with the mouse avoid the scrolling interactions but also speeds up the mouse velocity.
(Think about a fisheye view for Google Earth)

2-dimensional Fisheye

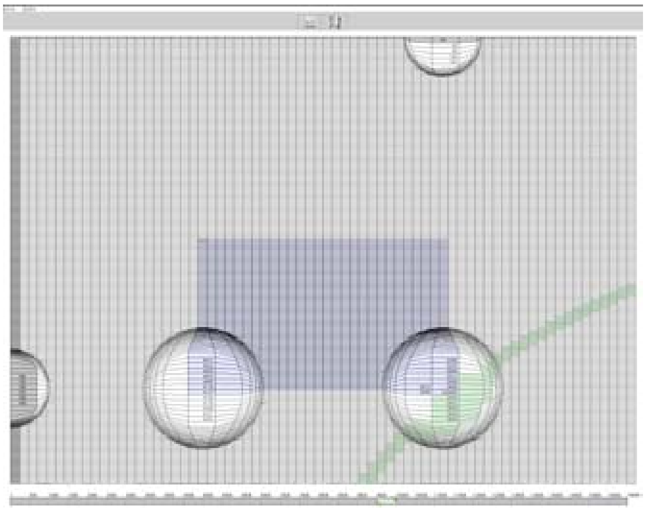
Document Lens

(G.G.Robertson, J:D.Mackinlay UIST 1993)



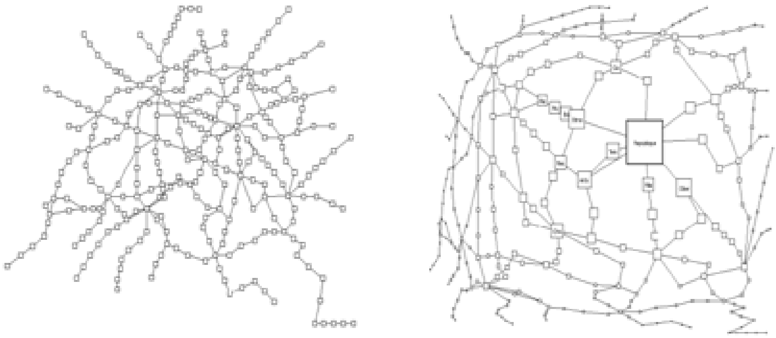
2-dimensional Fisheye

FiCell Project
http://iihm.imag.fr/vernier/



2-dimensional Fisheye

Fisheyes applied to networks

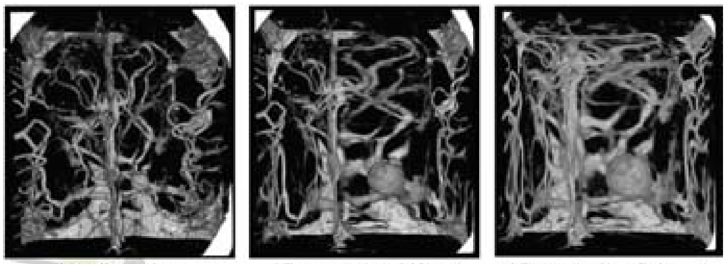


Manojit Sarkar and Marc H. Brown 1992

3-dimensional Fisheye

Marcelo Cohen, Ken Brodlie,

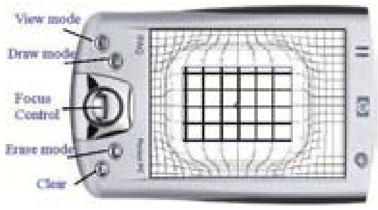
Focus and Context for Volume Visualization,



No distortion 3D cartesian bifocal 3D cartesian fisheye

Fisheye for input

- Edward Lank
  Fluid Sketching on a Pocket PC (Ubicomp 2004 Workshop)
  http://tlaloc.sfsu.edu/~lank/research/appearing/FocusMotion.pdf
- Edward Lank, Son Phan
  Focus+Context sketching on a pocket PC
  CHI '04 extended abstracts on Human factors in computing systems



Fisheye for input

Paper/Video from Mitsubishi

Forlines, C.; Balakrishnan, R.; Beardsley, P.; van Baar, J.; Raskar, R., "Zoom-and-Pick: Facilitating Visual Zooming and Precision Pointing with Interactive Handheld Projectors", ACM Symposium on User Interface Software and Technology (UIST), ISNB: 1-59593-271-2, pp. 73-82, October 2005 (ACM Press)

http://www.merl.com/people/forlines/papers/2005_forlines_zoom_and_pick.pdf
http://www.merl.com/people/forlines/videos/MERL_ZoomAndPick_highRes.mov

How to program

Fisheyes

for bitmaps

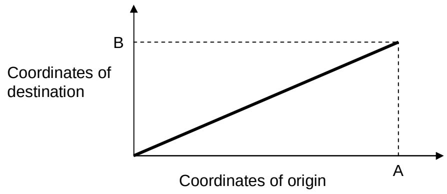
1-dimensional Fisheye

Normal scaling: Display an object of size A on a window of width B

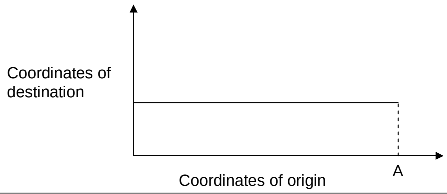
The magnifier function is the first derivate of the transfer function

The transfer function is the integral of the magnifier function

Transfer function T(X)



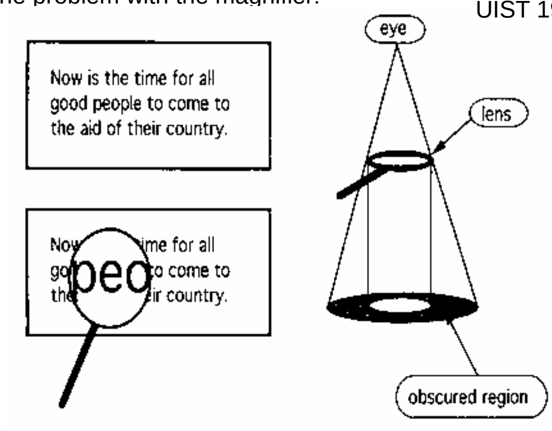
Magnifier function M(X)



1-dimensional Fisheye

The problem with the magnifier:

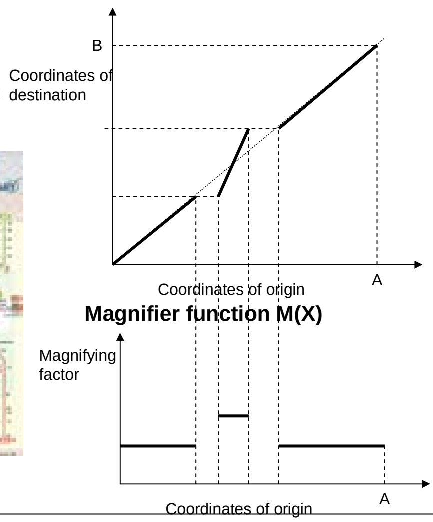
(G.G.Robertson, J:D.Mackinlay UIST 1993)



1-dimensional Fisheye

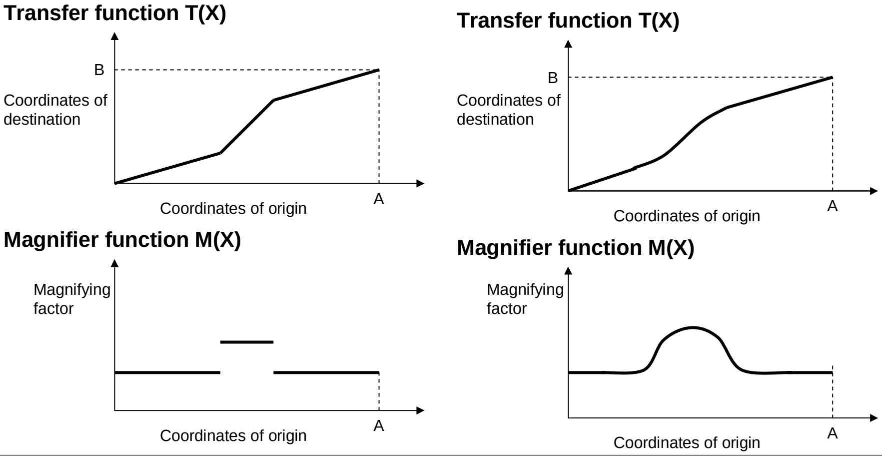
The problem with the magnifier:

Parts of the origin will not appear at the destination.
In the picture below the Central Station is visible, but not Marienplatz



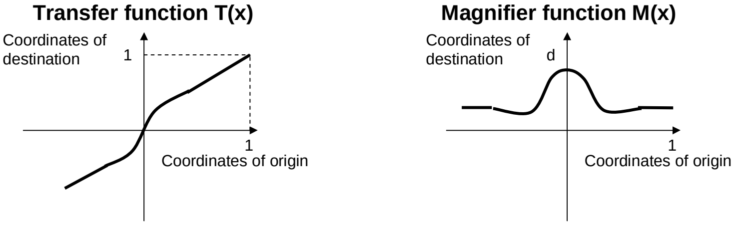
1-dimensional Fisheye

Bifocal: Continuous:



1-dimensional Fisheye

To have transfer function independent of window sizes and resolutions it is common to work with normalized coordinates, i.e. working with intervals from -1 to 1.



$T(X) = (1 + d) * X / (d * X + 1)$

$M(X) = (d + 1) / (d * X + 1)^2$

1-dimensional Fisheye

Transfer functions from
Y.K.Leung and M.D.Apperley

Table A.II. A Summary of Transformation and Magnification Functions

	Transformation Function T(x)	Magnification Function M(x)
Polyfocal Projection	$x+\frac{A.x}{(1+C.x^2)}$	$1+\frac{A.(1-C.x^2)}{(1+C.x^2)^2}$
Fisheye View	$\frac{(1+d).x}{(d.x+1)}$	$\frac{d+1}{(d.x+1)^2}$
Perspective Wall		
for $x \le a$,	$x.\frac{b}{a}$	$\frac{b}{a}$
for $x > a$,	$\frac{[b+(x-a).\cos\theta]}{1-[\frac{(1-b)}{(1-a)}-\cos\theta].(x-a)}$	$\frac{b.k+(1-b).\cos\theta}{[(k-\cos\theta).x+(a.\cos\theta-a.k-1)]^2}$ note: $k=\frac{(1-b)}{(1-a)}$
Bifocal Display		
for $x \le a$,	$x.\frac{b}{a}$	$\frac{b}{a}$
for $x > a$,	$b+(x-a).\frac{(1-b)}{(1-a)}$	$\frac{(1-b)}{(1-a)}$

2-dimensional Fisheye

Applying transfer functions for x- and y-coordinates independently does not give a nice result.

2-dimensional Fisheye

The transfer function for X should depend on Y. For Y=0 in normalized coordinates the transfer function for x should be the 1-dimensional fish eye transfer function T(X). For y=1 it should be the undistorted transfer function T_u , normally $T_u(X) = X$.

This can be achieved by a weighting function W(Y) with values from 0 to 1. ("function morphing")

$T(X, Y) = (1-W(Y)) * T(X) + W(Y) * T_u(X)$; $W(0) = 0$; $W(1) = 1$;

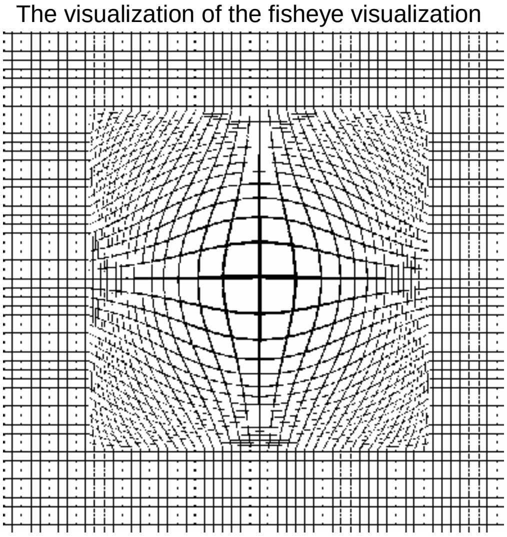
Examples:
$W(Y) = Y$
$W(Y) = Y^2$

2-dimensional Fisheye

Continuous transfer function

using Cartesian coordinates

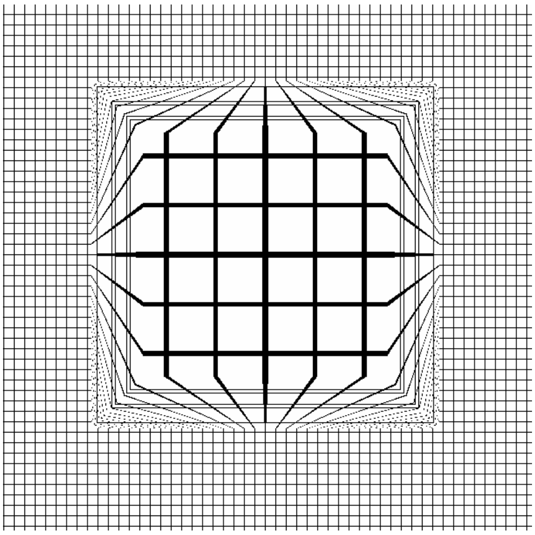
The visualization of the fisheye visualization



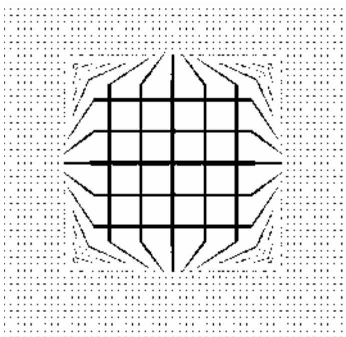
2-dimensional Fisheye

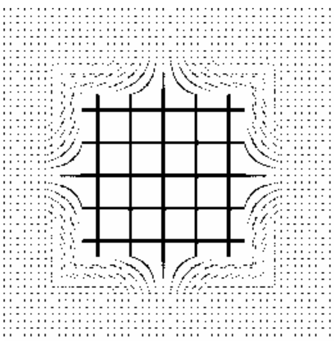
Bifocal transfer function

using Cartesian coordinates



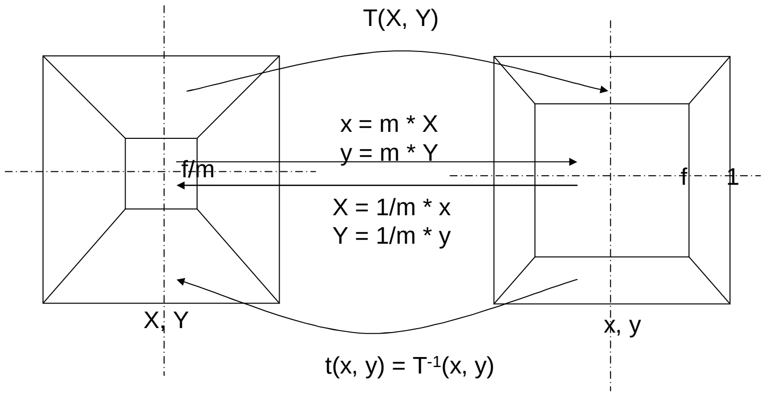
2-dimensional Fisheye





What is the difference?

2-dimensional Fisheye



This is one part of the exercise

2-dimensional Fisheye

Using polar coordinates

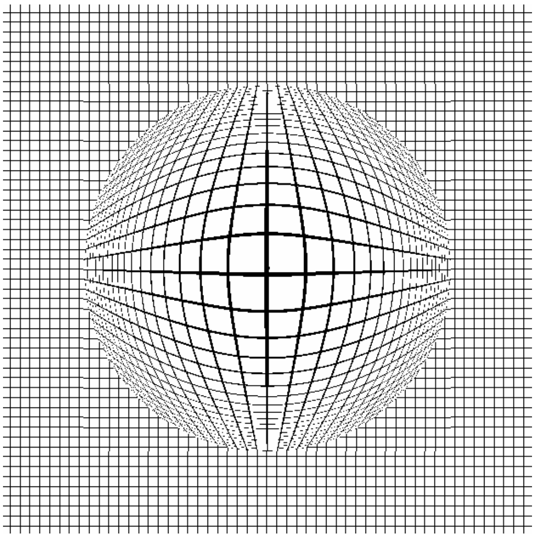
Because a fish eye should not twist the picture, the transfer function does not depend on the angular coordinate. So the transfer function for the 1-dim. case can be used for the radial coordinate.

$T(r, \varphi) = (T_{1dim}(r), \varphi)$

2-dimensional Fisheye

Continuous transfer function

using polar coordinates

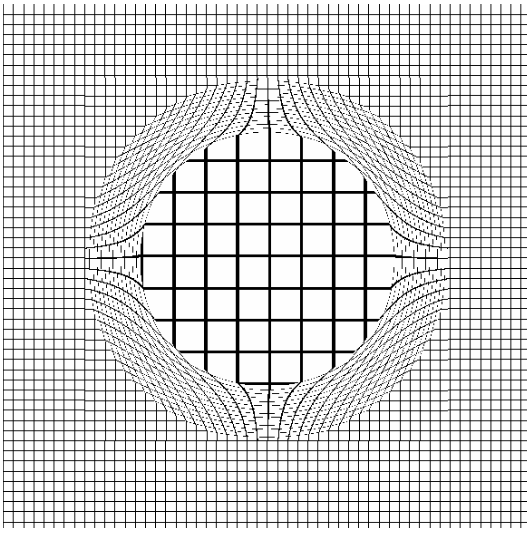


This is the other part of the exercise

2-dimensional Fisheye

Bifocal transfer function

using polar coordinates



Hints for Programming

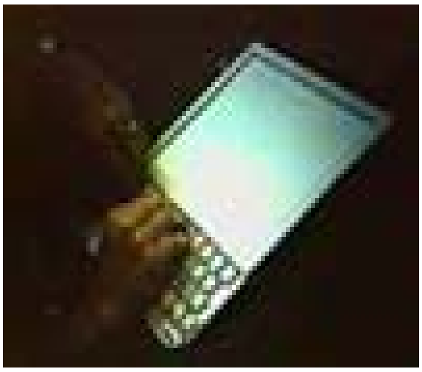
- For bitmaps iterate over the pixel of the destination bitmap using the inverse transfer function $(X,Y) = T^{-1}(x, y)$
  - No pixels are left out
  - The number of pixel are less
- The multiplication of integers and floats may have unexpected results!
- Use well chosen names for variables

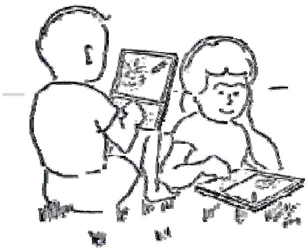
Chapter 3: Mobile HCI

Table of Content

- Input & Output Devices
- Input & Output Techniques
- Guidelines
- System Architectures for Mobile UIs
- Example: Applications for Mobile Phones

Dynabook Vision

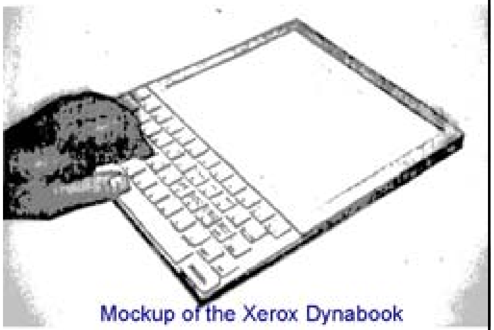




- Handheld,
- wireless connectivity,
- multimedia capabilities
- support for programming

Mobile Computing / mobile UIs 1972 Xerox Dynabook

- Alan Kay's group at Xerox PARC
- First description of "mobile computing" with a focus on the UI?
- a portable interactive personal computer, as accessible as a book
- a computer for children (learning aid)
- Big problem: software that facilitates dynamic interactions between the computer and its user



Mockup of the Xerox Dynabook

http://www.honco.net/os/kay.html
The Dynabook Revisited - A Conversation with Alan Kay

Mobile User Interfaces

- "Beyond the laptop…"
- Devices are used while the user is mobile
  - Handhelds & PDAs
  - Phones
  - Wearable Computer
  - Tablet Computers
  - Car Infotainment system

Apple Newton
Commercial Handheld Computer

- Recognition Architecture
  - Recognizes handwriting--printed, cursive, or a mixture of the two--with the assistance of a 93,000-word, built-in word list
  - Lets you add up to 1,000 words
  - Includes four pop-up keyboards: typewriter, numeric, phone, and time/date
  - Recognizes graphics and symmetrical objects
- 320 by 240 pixels Display
- Sold from 1993

http://www.oldschool.net/newton/papers/index130.html

Itsy
Pocket Computer

- Research platform
- Gesture and speech interaction
- *tilt-to-scroll* and *Rock 'n' Scroll* to include the use of gestures to issue commands.

- http://research.compaq.com/wrl/projects/itsy/itsy.html
- http://research.compaq.com/wrl/projects/itsy/movies.html

Input to Mobile Devices
What to input?

- Commands
- Text
- Drawings/sketches
- Images
- Audio
- Movies

Input to Mobile Devices How to input?

- Keyboards
  - Full-size
  - Miniature
  - Chord-keyboard
  - On-screen
- Stylus
  - Point and click
  - Handwriting recognition
- hard buttons / wheels
  - Scroll wheels
  - Joypad-style navigation
- Capture
  - Camera
  - microphone
- Future devices
  - Tilt scrolling
  - Virtual workspaces

Input Technologies for Mobile Devices

- Soft Keyboards
- Screen Keyboards

Input Technologies for Mobile Devices

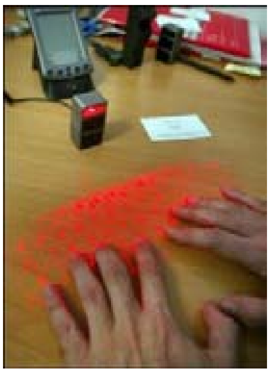
- Keyboards





Input Technologies for Mobile Devices

- Virtual Keyboards
- Projection Keyboards


http://www.alpern.org/weblog/stories/2003/01/09/projectionKeyboards.html

Input Technologies for Mobile Devices

- Chord Keyboard
- One-handed Keyboards
- Example Twiddler
  - Combines keyboard and Mouse
  - keypad designed for "chord" keying This means you press one or more keys at a time. Each key combination generates a unique character or command.
  - 12 finger keys and 6 thumb keys, the twiddler can emulate the 101 keys on the standard keyboard

Yoyo Input Device designed for artic environments

Figure 5. The Yo-Yo user interface.

- Smart Clothing for the Arctic Environment by J. Rantanen et al. in proceedings of the int. Symposium on Wearable Computing 2000 (ISWC2000)