Automatic Tactilization of Graphical Images

Computer Vision
CSE 455
Winter 2005

Richard Ladner

The Tactilization Problem

- Graphical images are heavily used in math, science and engineering textbooks and papers
  - Line graphs and bar charts
  - Diagrams
  - Illustrations
- Tactual perception is the best modality for the blind to understand such images
- Tactilization of graphical images
  - Currently done manually
  - Labor-intensive and time consuming
  - How much of this process can be automated?

Outline

- Tactual Perception
- Overview of tactilization process
- Text segmentation
- Braille text placement
- Other subprojects
- Demonstration

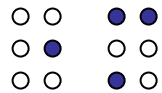
Tactile Perception

- Resolution of human fingertip: 25 dpi
- Tactual field of perception is no bigger than the size of the fingertips of two hands
- Color information is replaced by texture information
- Visual bandwidth is 10^6 bits per second, tactile is 10^2 bits per second

Braille

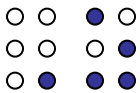
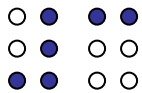
- System to read text by feeling raised dots on paper (or on electronic displays). Invented in 1820s by Louis Braille, a French blind man.

a  b  c  z 

and  the  with  mother 

th  ch  gh 

Critical fact: Fixed height and width

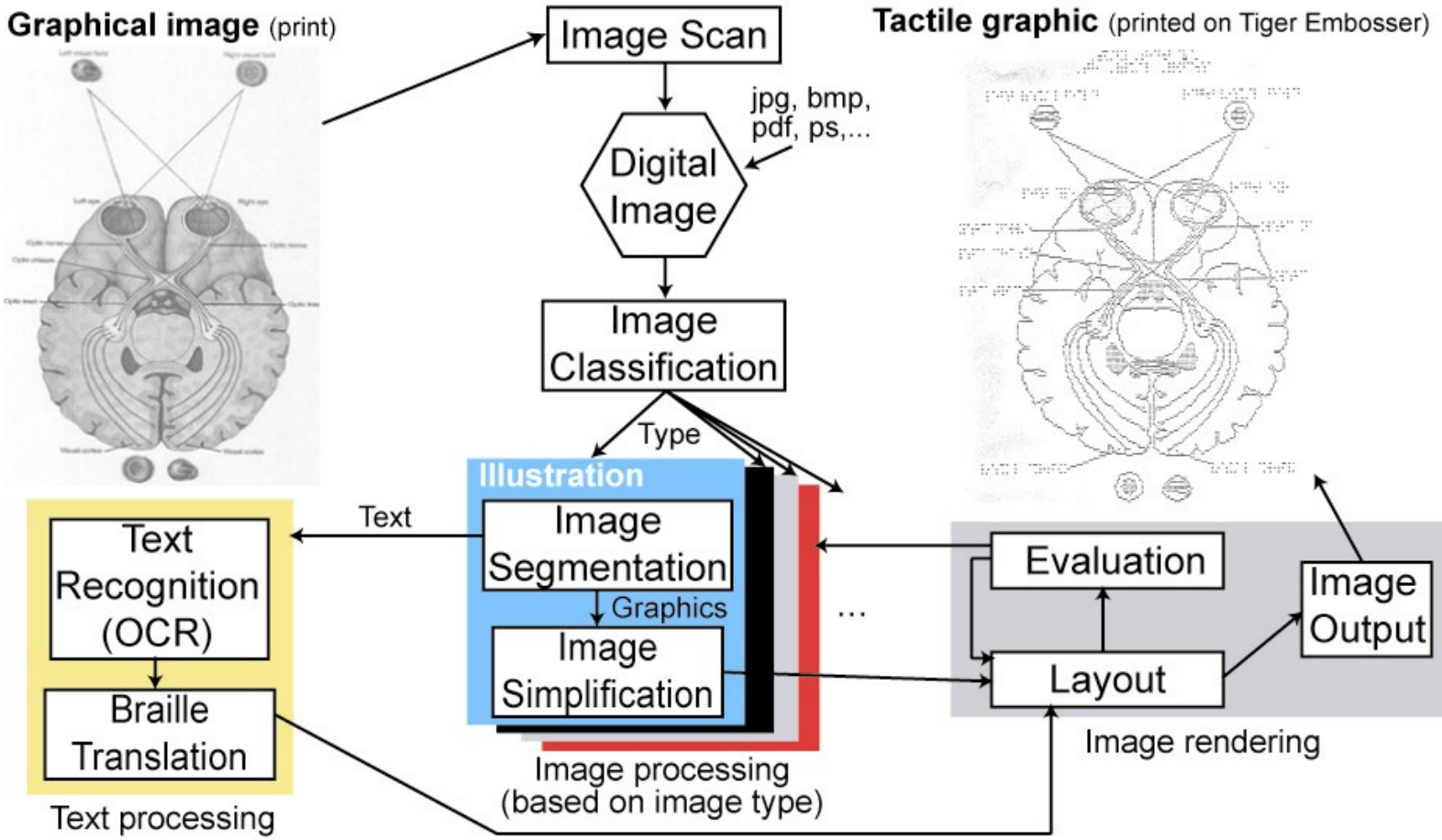
Z  3 

Mode characters: cap and num.

Tiger Embosser

- 20 dpi (raised dots per inch)
- 7 height levels (only 3 or 4 are distinguishable)
- Prints Braille text and graphics
- Prints dot patterns for texture
- Invented by a blind man, John Gardner

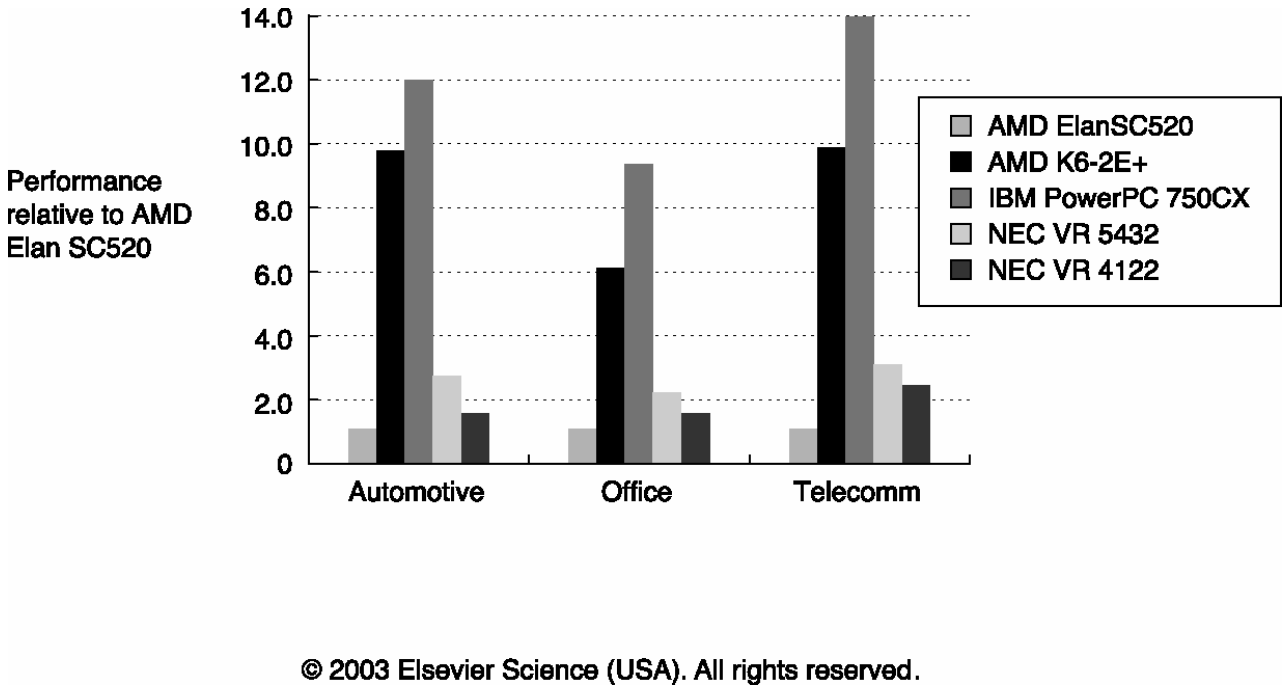
Automatic Tactilization Process



Key Problems

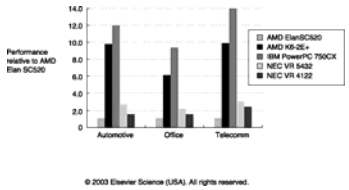
- Graphical images meant for the visual mode must be modified for the tactual mode.
  - Text $\Rightarrow$ Braille
  - Colors $\Rightarrow$ replace with textures or reduce number
  - Area $\Rightarrow$ Larger for Braille text to fit
  - Resolution $\Rightarrow$ Lower to 20 dpi
  - Shading or 3-D effects $\Rightarrow$ replace with outlines
  - Noise $\Rightarrow$ remove noise, enhance contrast
- Classification of graphical images for mass production
  - Images in the same class require similar processes.

Example

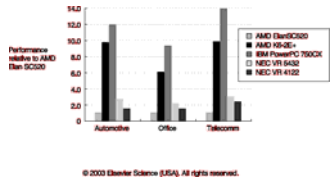


From *Computer Architecture, A Quantitative Approach, Third Edition*, by Hennessy and Patterson.

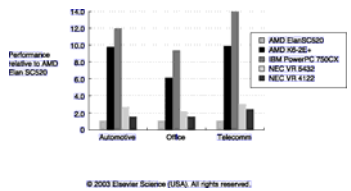
Overall Process



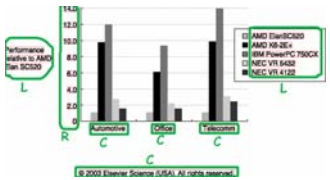
Identify letters



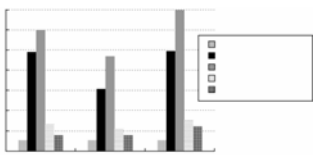
Merge letters into text blocks



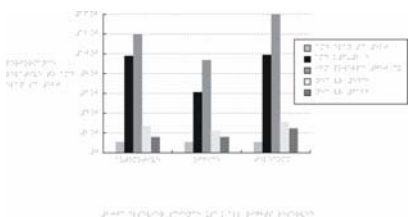
Identify text blocks: left or right justified or centered



Scale image, add texture

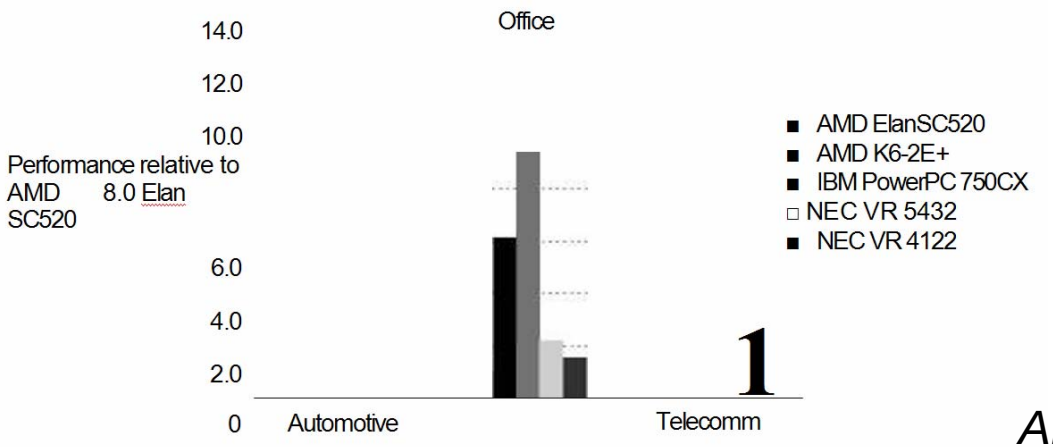


OCR text blocks, Translate to Braille



Finding Text

- Why not just use standard optical character recognition (OCR)?
  - OCR is not effective for graphical images.



ABBYY FineReader 7.0 Professional Edition

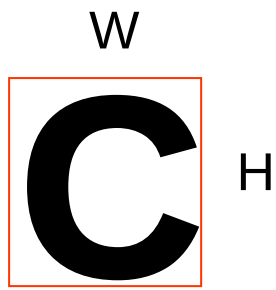
Finding Text Letters

- Uses the following principles
  - Text in an image is usually in one font
  - Fonts are designed to have a uniform density at a distance.
  - In the absence of noise an individual letter tends to be connected component of one color. Exceptions are i and j.
- Train on some simple features of letters. Connected components with similar features are also letters.

Features

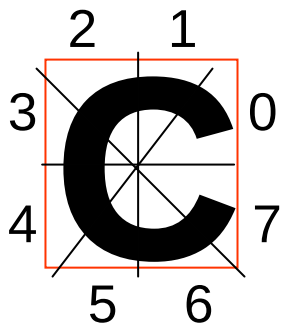
Century Gothic

W = width of bounding box
H = height of bounding box
A = area of bounding box
R_i = i-th radial slice density



$A = W \bullet H$

R_i = number of black pixels in i-th slice where a slice is an angle of 360/n. The total number of slices is n.

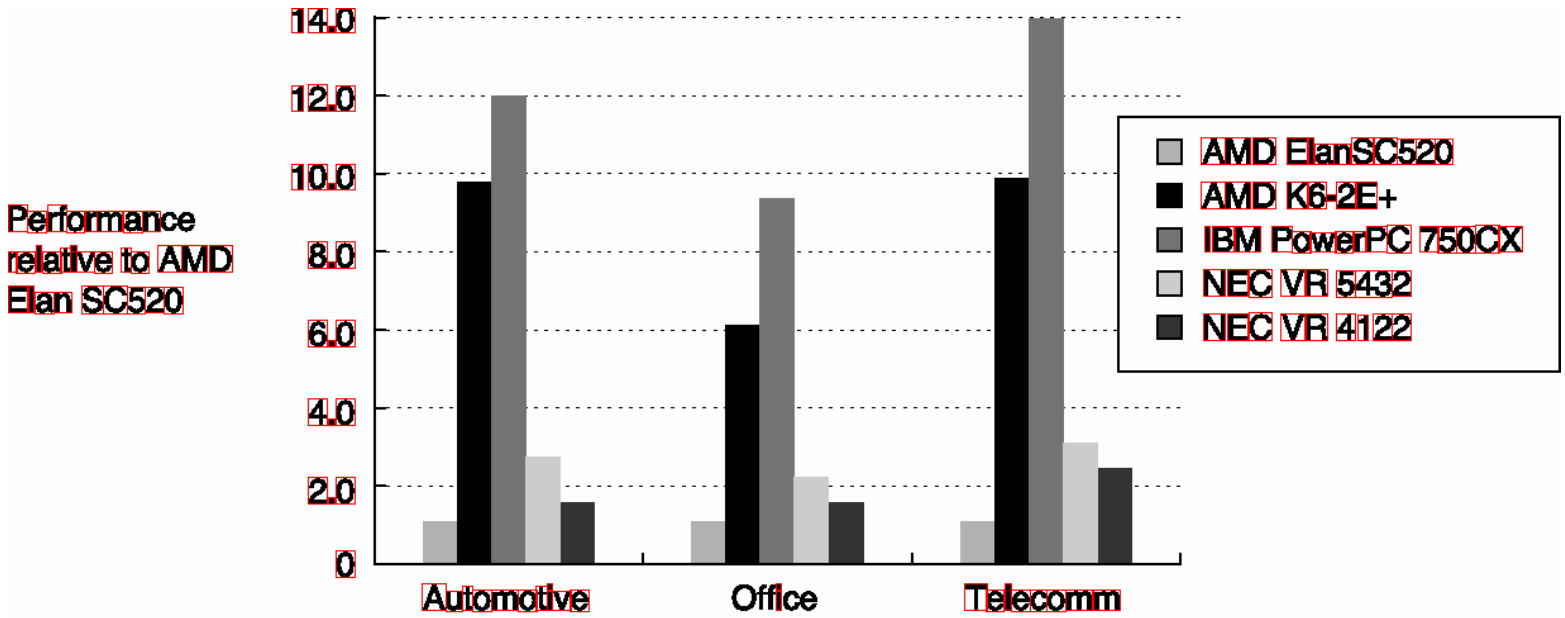


Center is center of mass of black pixels

Training/Finding

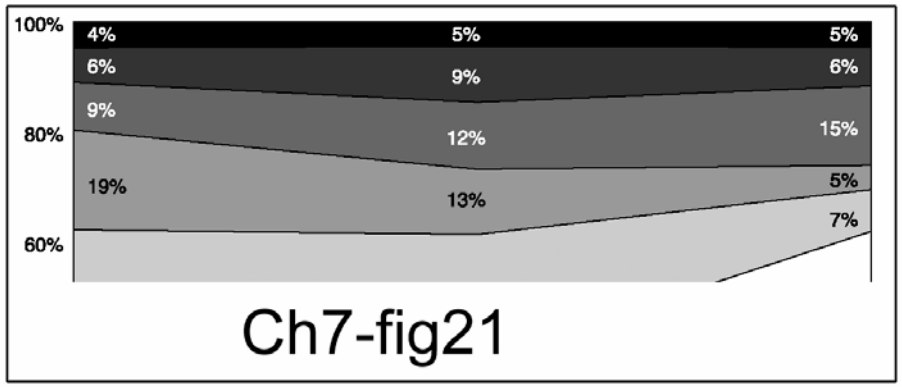
- Training:
  - Sample the connected components and compute their features.
- Finding:
  - For a new connected component compute its features.
  - If there is a close enough match of features with some member of the database then declare the component to be a letter.
- Parameters
  - How close is close enough
  - How many slices

Example



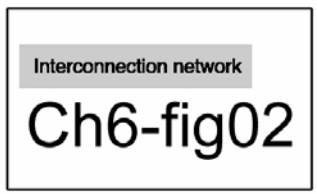
Trained on a different images from the same book.
About 200 letters in the training set.

Other Issues



Ch7-fig21

Ch5-fig55



Ch6-fig02

Computer Architecture: A Quantitative Approach
Ch8-fig43

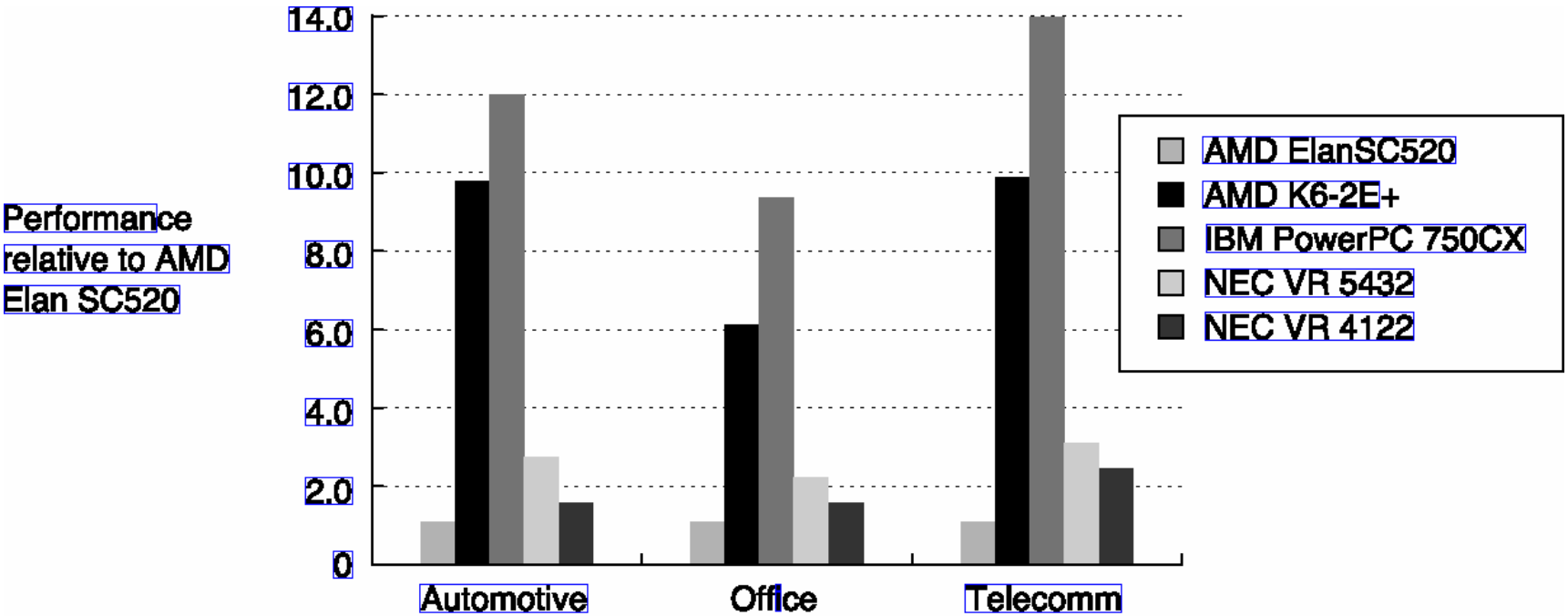
Ch5-fig43

Ch6-fig11

Finding Text Blocks

- Principles
  - Most text tends to be in horizonal lines
  - Some text is vertical
  - Some text is diagonal
- We are developing methods that find lines using the centroids of the letters found.
  - Minimum spanning tree
  - Merge test using linear regression

Current Progress



Problems

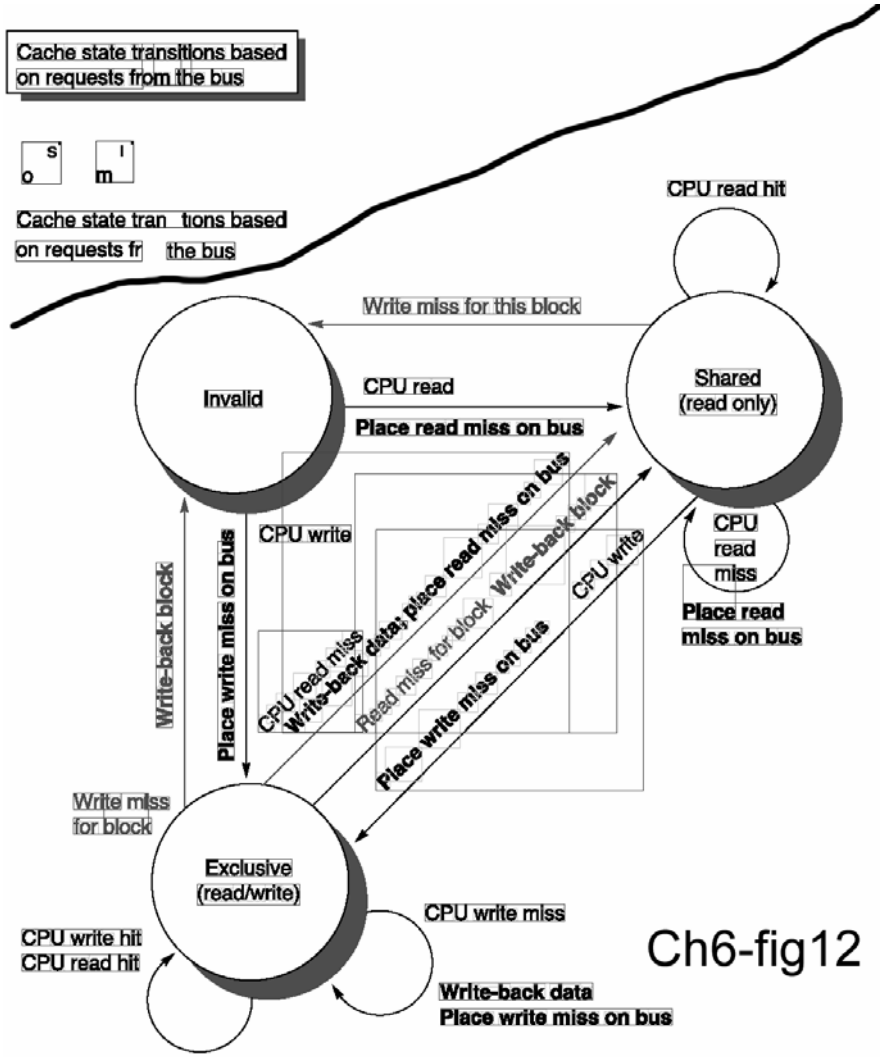
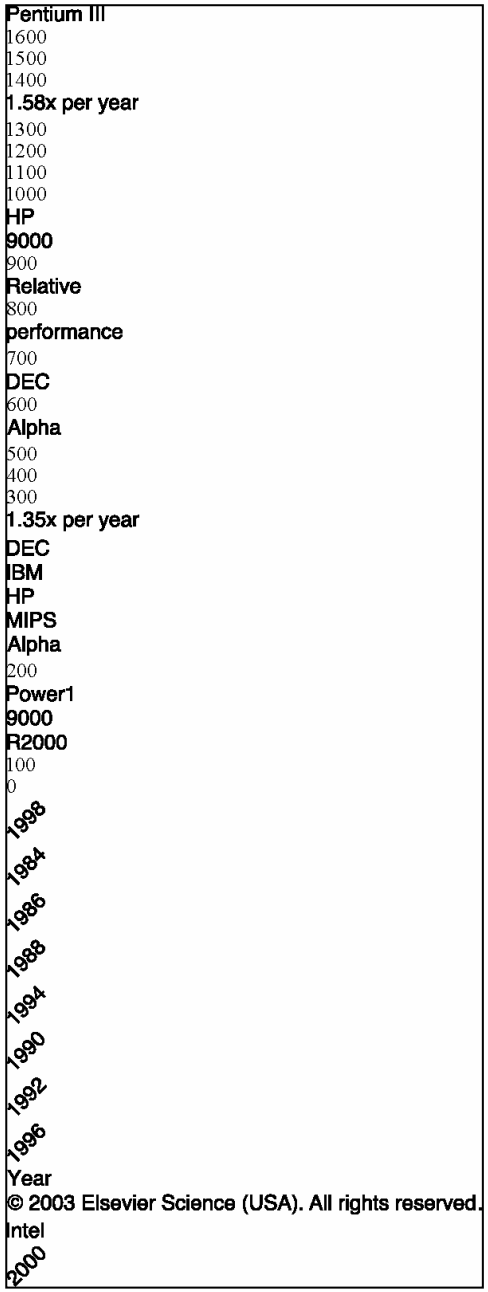

Cache state transitions based on requests from the bus
Cache state tran tions based on requests fr the bus
CPU read hit
Write miss for this block
Invalid
CPU read
Place read miss on bus
Shared (read only)
CPU write
Write-back block
Place write miss on bus
CPU read miss
Write-back data; place read miss on bus
Write-back block
CPU write
CPU read miss
Read miss for block
Place write miss on bus
Place read miss on bus
Write miss for block
Exclusive (read/write)
CPU write miss
CPU write hit
CPU read hit
Write-back data
Place write miss on bus
Ch6-fig12

Image of text boxes



OCR →

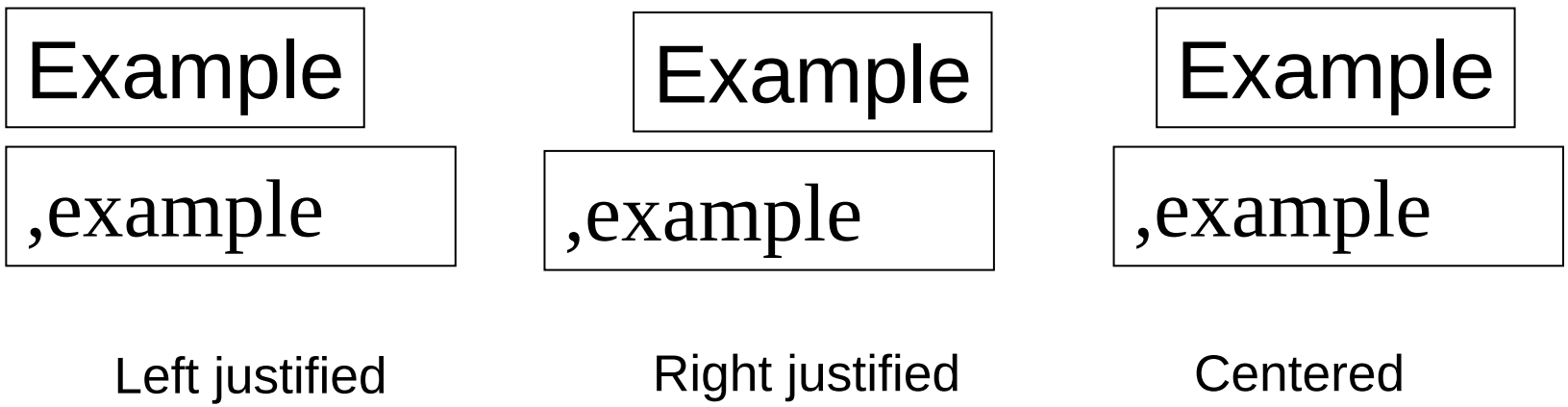
Pentium III
1600
1500
1400
1.58x per year
1300
1200
1100
1000
HP
9000
900
Relative
800
performance
700
DEC
600
Alpha
500
400
300
1.35x per year
DEC
IBM
HP
MIPS
Alpha
200
Power!
9000
R2000
100
0
#
^
#
^
f
f
f
f
Year

Intel
/

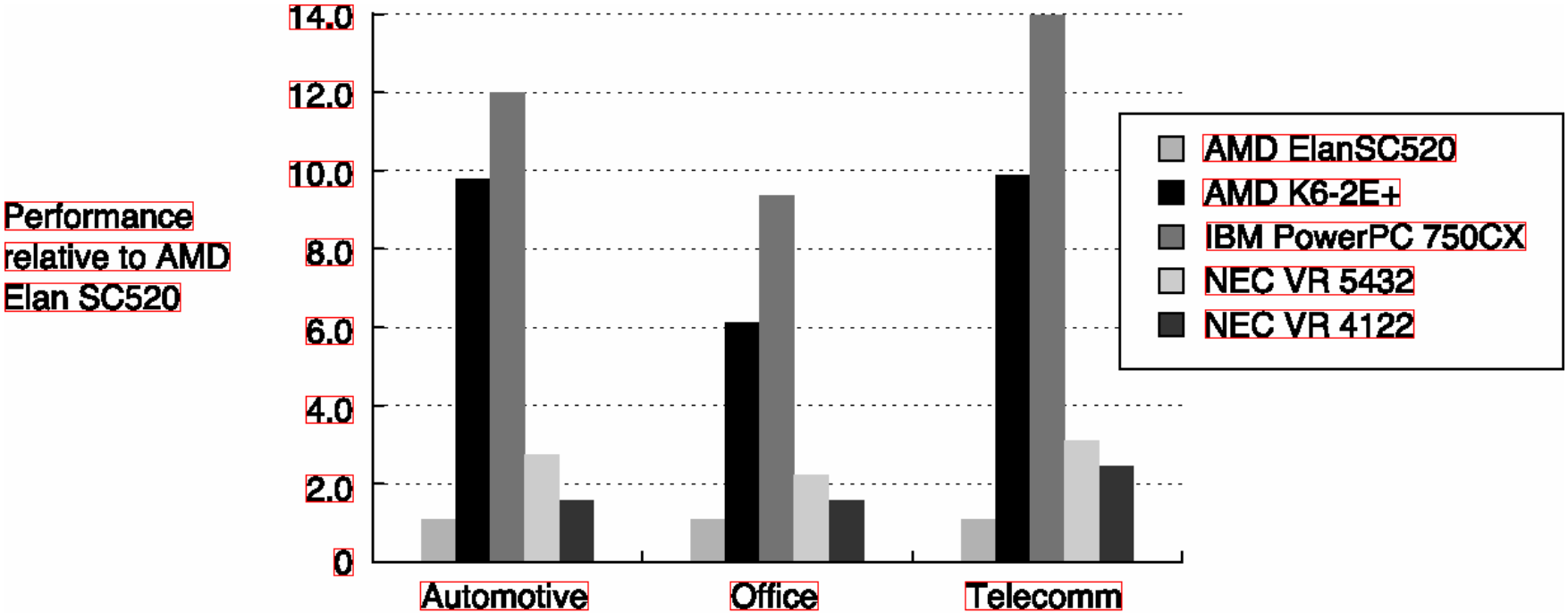
text

Classification of Text Boxes

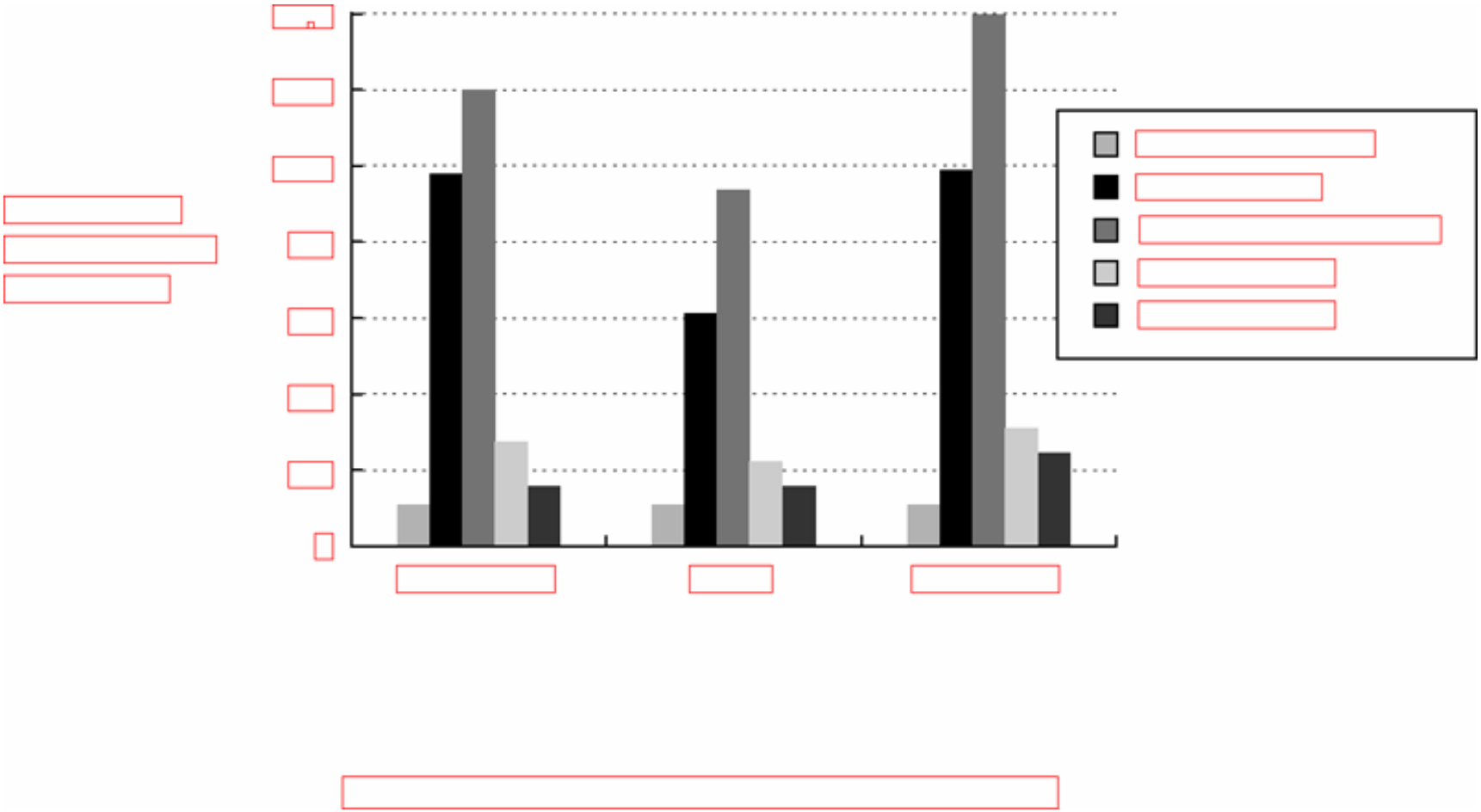
- Text boxes of Braille will be of different size than the original text boxes
  - Mode characters
  - Contractions
  - Braille is fixed width



Perfect Text Boxes



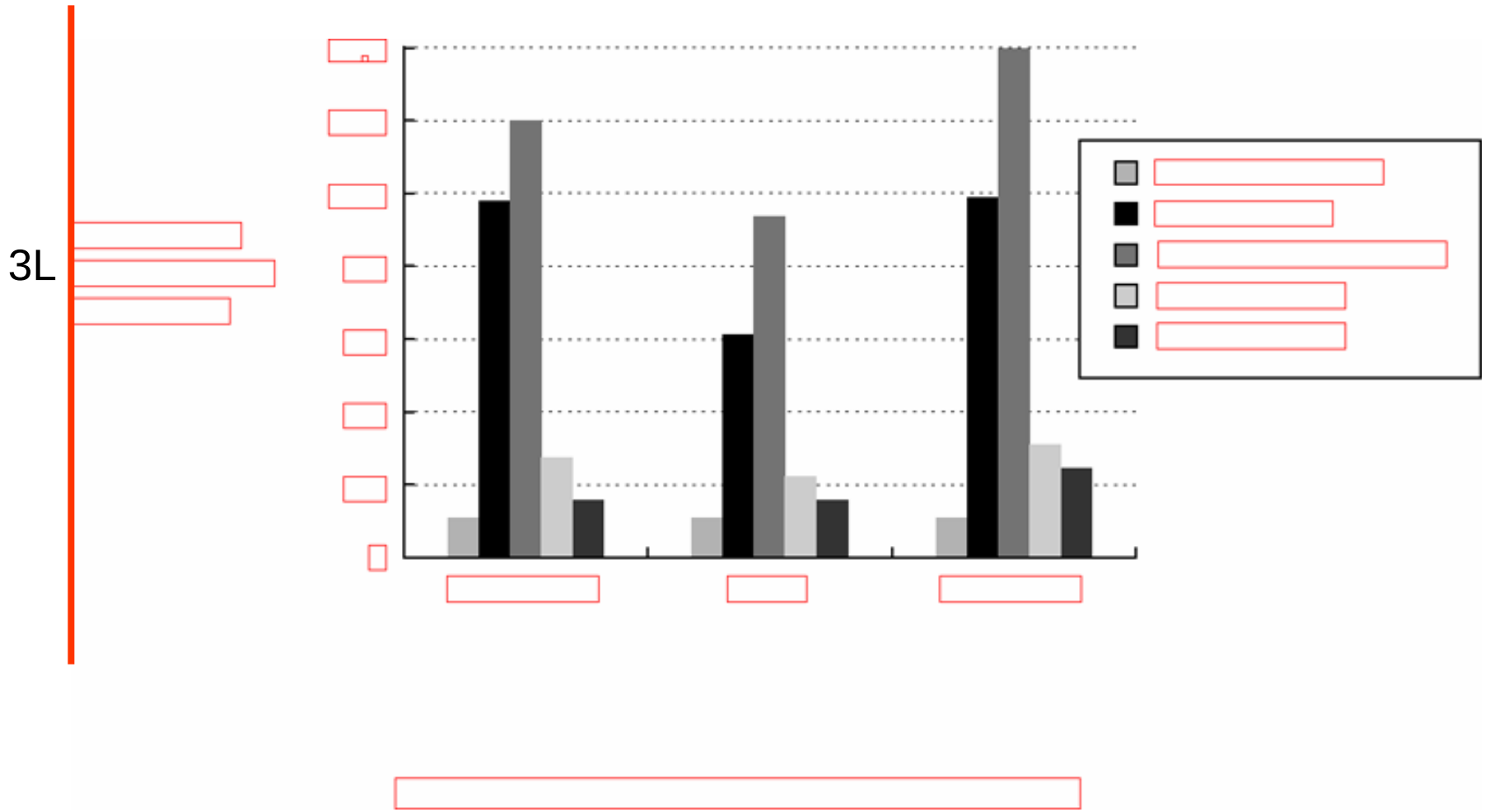
Text Boxes Only



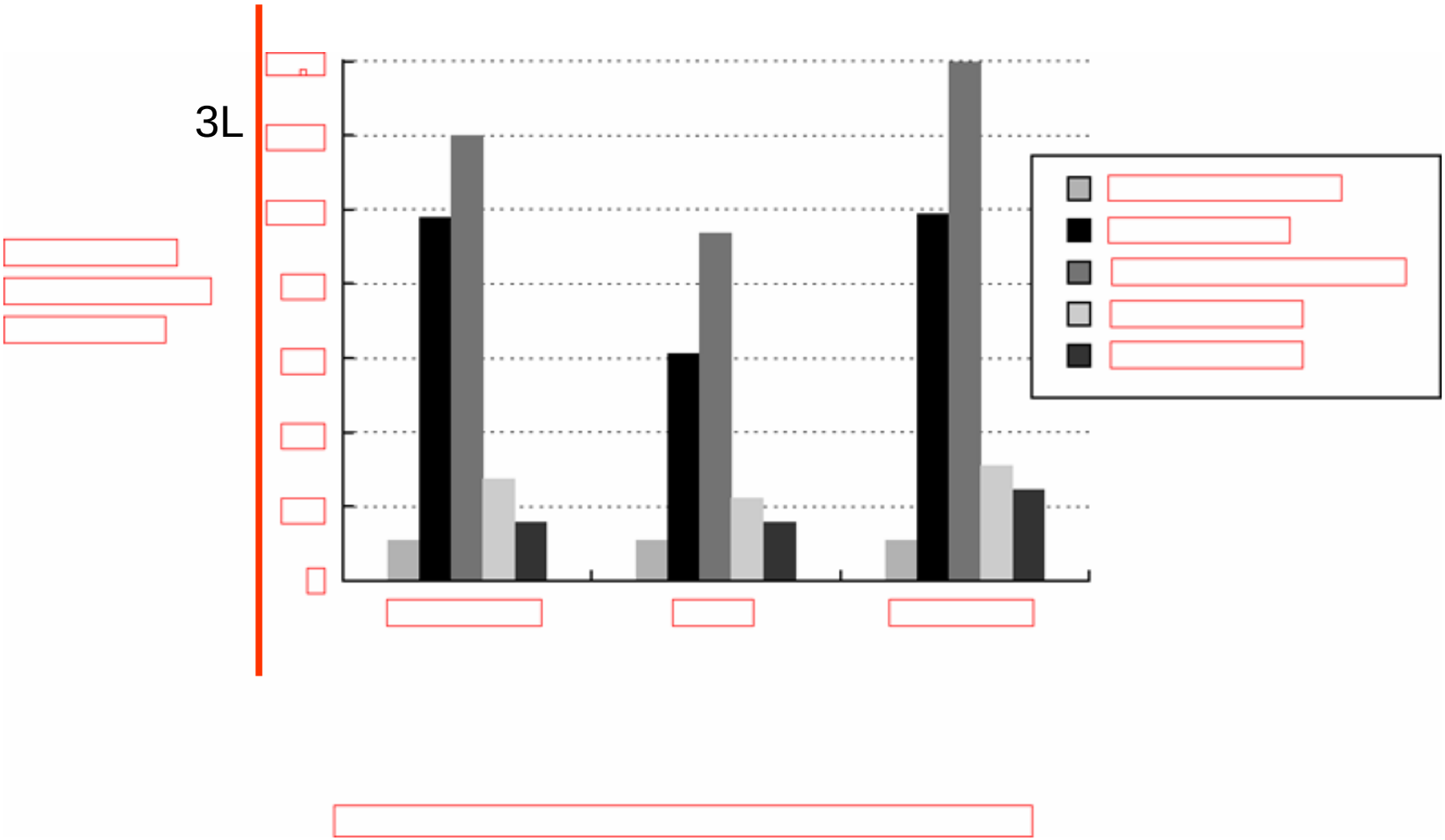
Justification Process

- Sort the upper left and lower right points of text boxes first by x then by y. Use a plane sweep algorithm.
- Left justify - runs (in y) of text boxes with the same (or similar) left x coordinates.
- Right justify - runs (in y) of text boxes with the same (or similar) right x coordinates.
- Center - otherwise

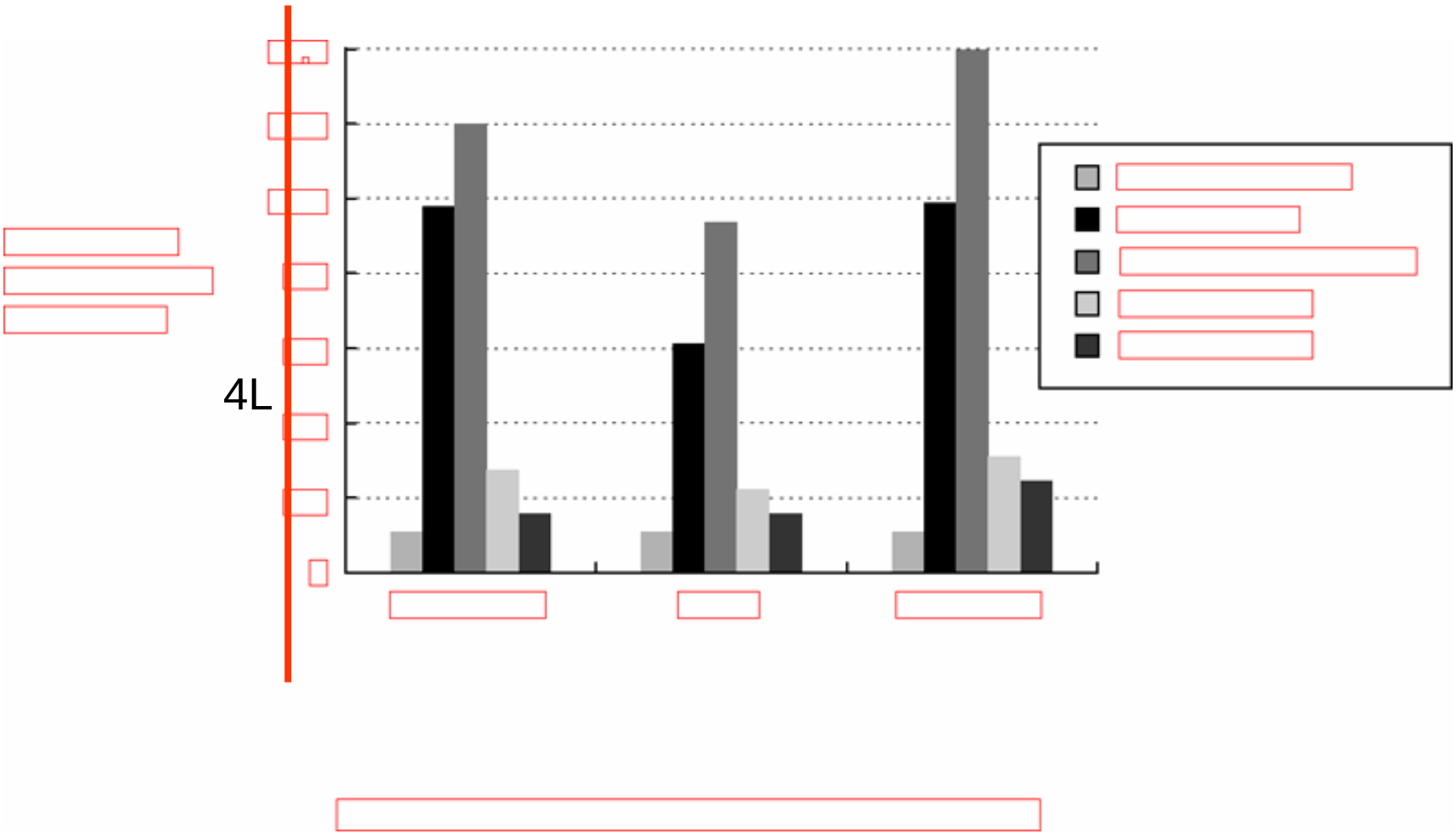
Example Plane Sweep



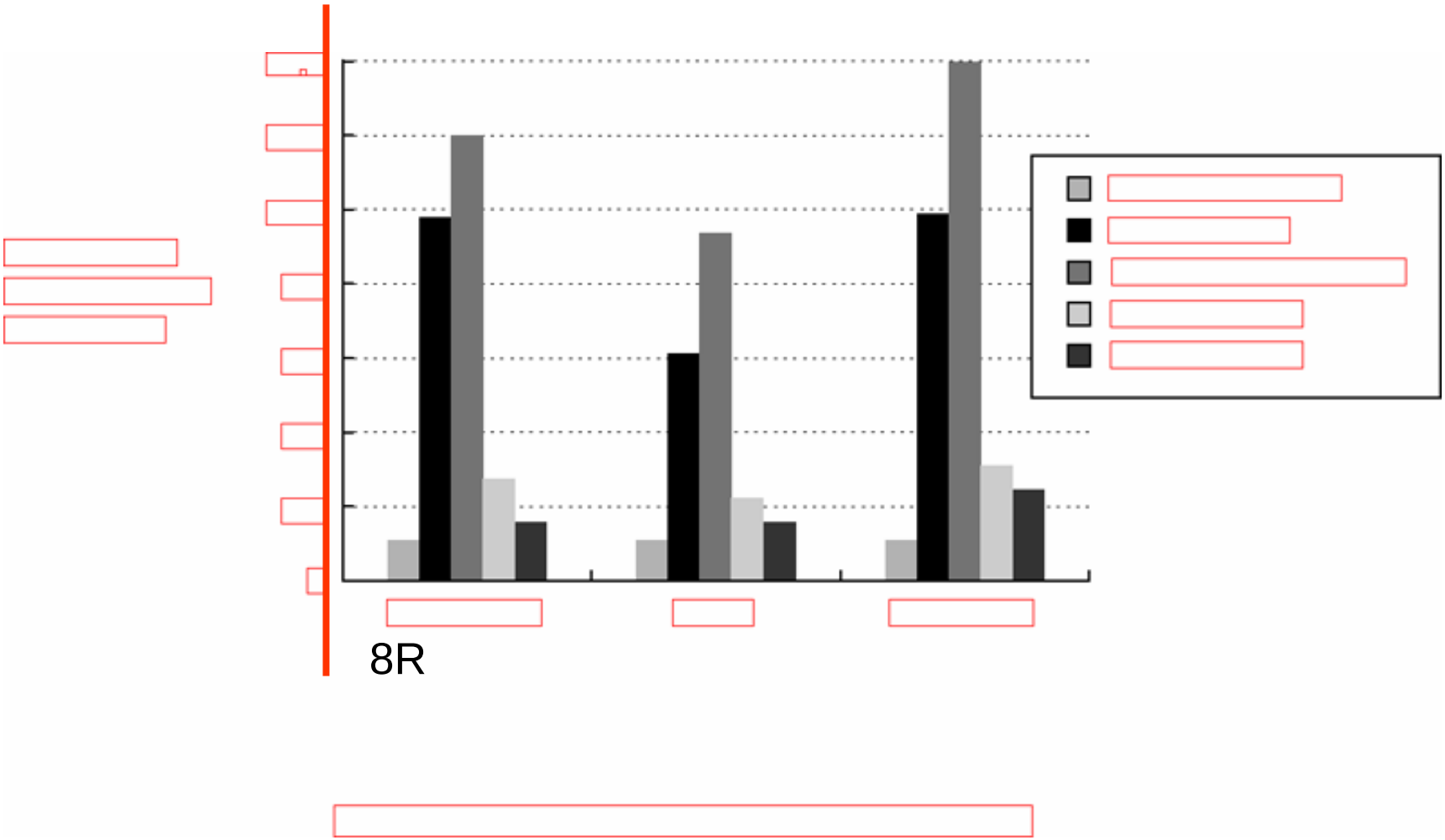
Example Plane Sweep



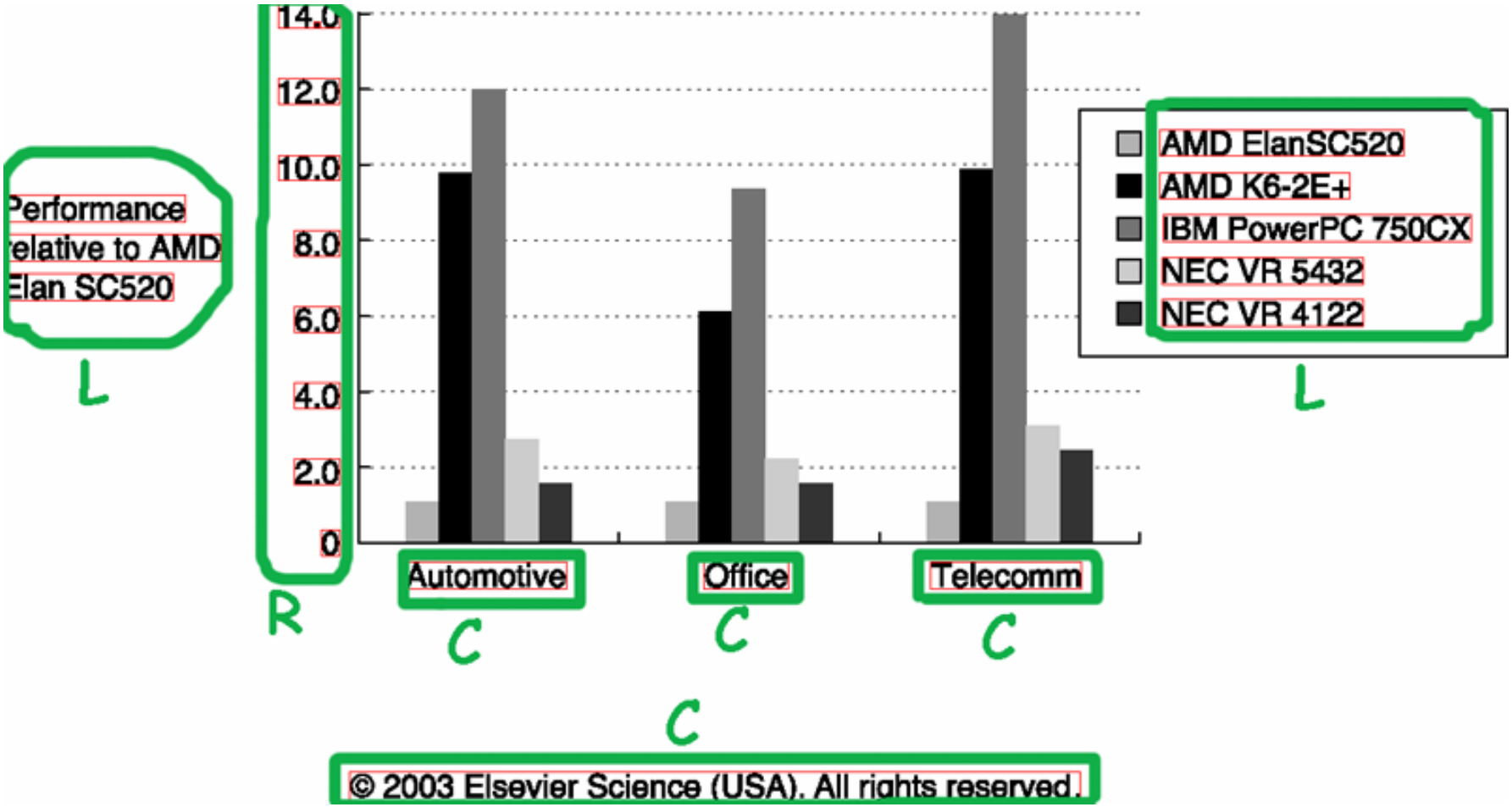
Example Plane Sweep



Example Plane Sweep



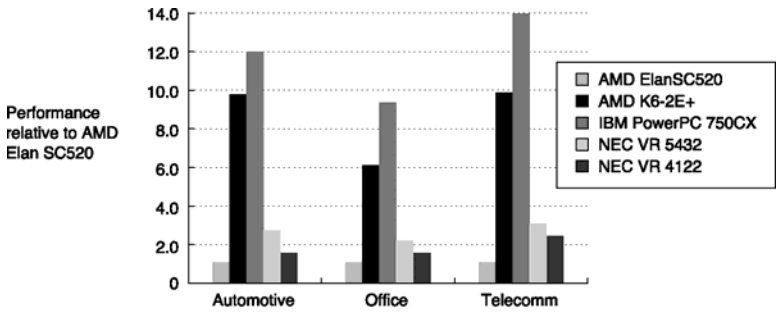
Classification

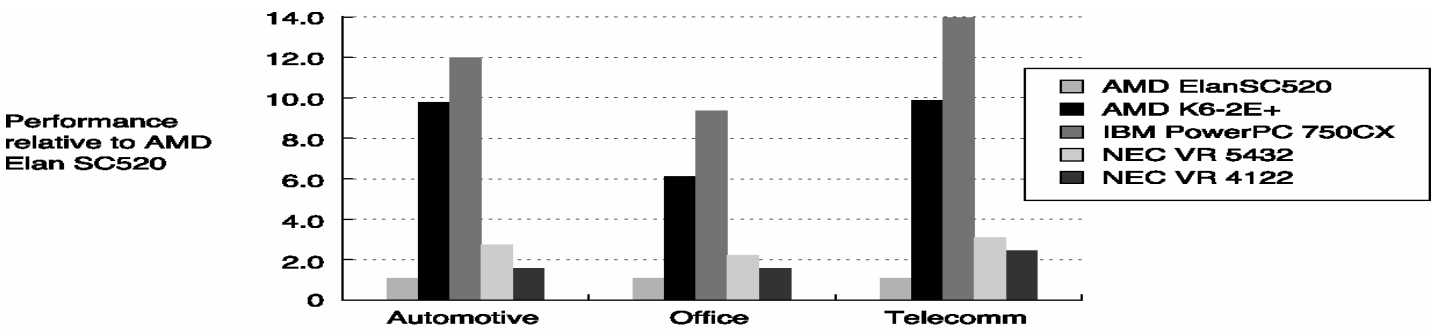


Scaling

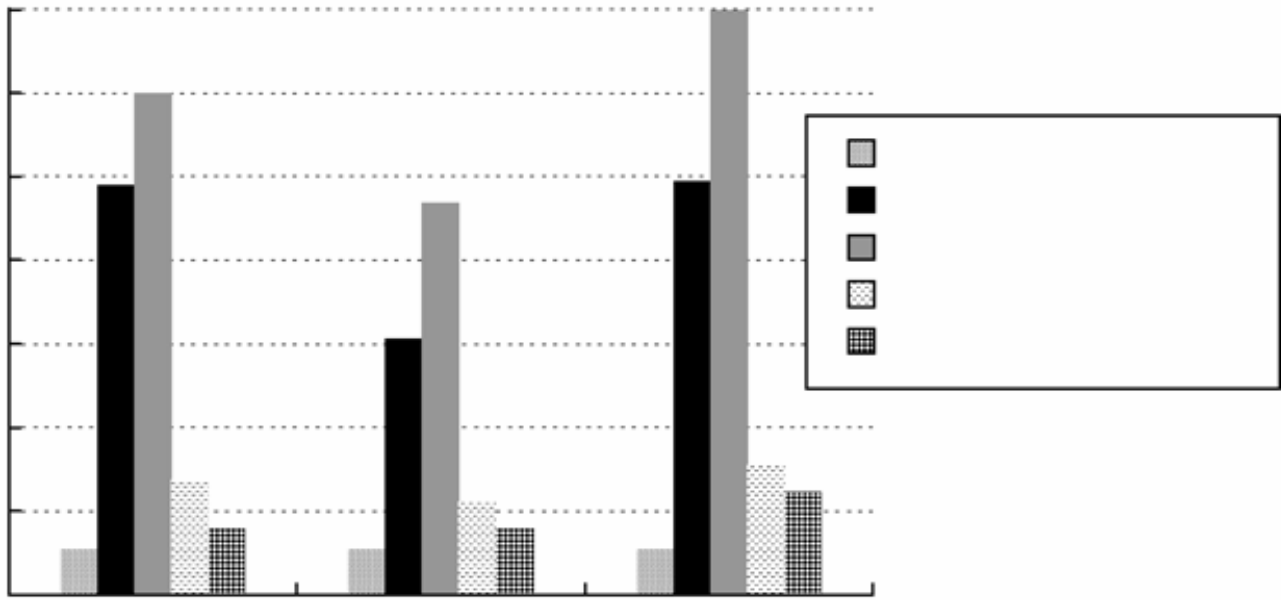
- General Procedure
  - Scale in y until the the text height is an acceptable Braille height
  - Scale in x until the Braille correctly justified fits
- The scale factor in x and y may differ, but the distorted image is usually readable.
  - The Braille text is fully readable.
- Scaling procedure is not always successful because of limited paper size.
  - Automatic abbreviations

Scaling Example

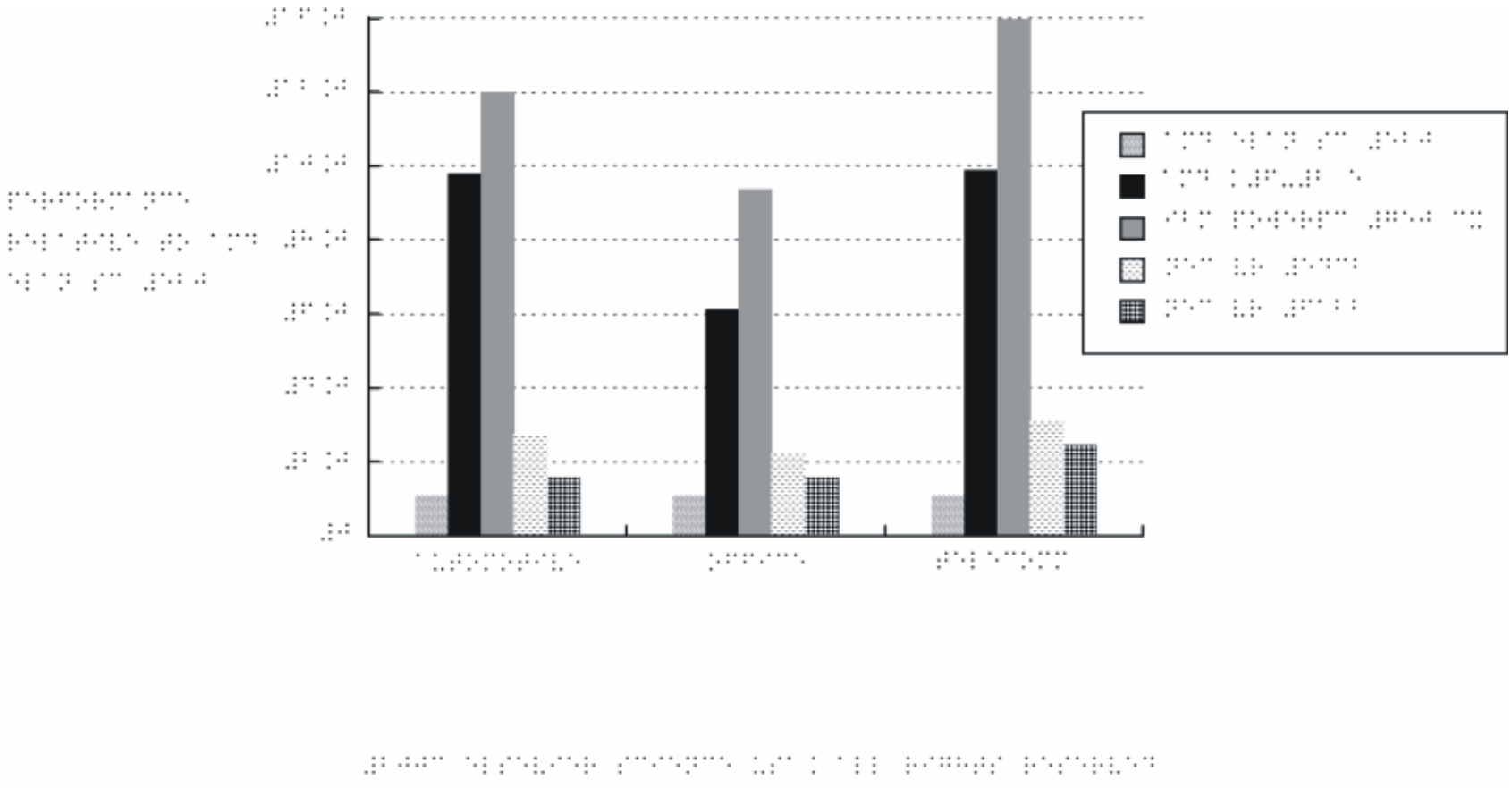




Color Replacement with Texture



Final Result



Other Subprojects

- Image Classification
  - Classify images in a text book by diagram, bar chart, or line graph
- User Interface Issues
  - Study of current practices
  - Information School participation
- Noisy Images
  - Noise removal
  - Can’t use connected components to find text
- Campus Map Project
  - Indexed color

Tactile Graphics Team

- Faculty:
  - Melody Ivory-Ndiaye (iSchool)
  - Richard Ladner (CSE),
  - Raj Rao (CSE)
  - Sheryl Burgstahler (DO-IT and Education)
- Grad Students:
  - Sangyun Hahn (CSE)
  - Beverly Slabosky (iSchool)
- Undergrads:
  - Satria Krisnandi
  - Maha Ramasamy
  - Matt Renzelmann
  - Jack Hebert
  - Dana Wen
  - Andrew Martin
  - Amy Lacenski
  - Stuart Olsen
- Staff:
  - Dan Comden (Access Technology Lab)

Tactile UW Campus Map

- Uses a different approach because images are different.
- Adobe Photoshop does much of the work.
- Steps we'll show you
  - Remove text
  - Change background color
  - Remove 3-d effect
  - Texture parking lots
  - Outline buildings