PIBE:
A Personalizable Image Browsing Engine

Ilaria Bartolini, Paolo Ciaccia,
and Marco Patella

IEIIT-BO/CNR – DEIS

University of Bologna, Italy

CVDB - Paris, June 13, 2004

Outline

- Why Image Browsing?
- Existing Browsing Systems
- PIBE (Personalizable Image Browsing Engine)
  - Functionalities
  - Customizable Hierarchical Browsing Structure
  - Visual Examples
- Experimental Evaluation
- Conclusions & Future Work

Motivation

- Exploration of large image databases (DB's) is a complex and tedious task
- Difficulties in characterizing the image content and in defining suitable comparison criteria
- Effective interaction with large DB's has to include not just querying but also browsing facilities:
  - to determine a good starting point for searching
  - to get an overall view of the DB contents

Existing Browsing Systems

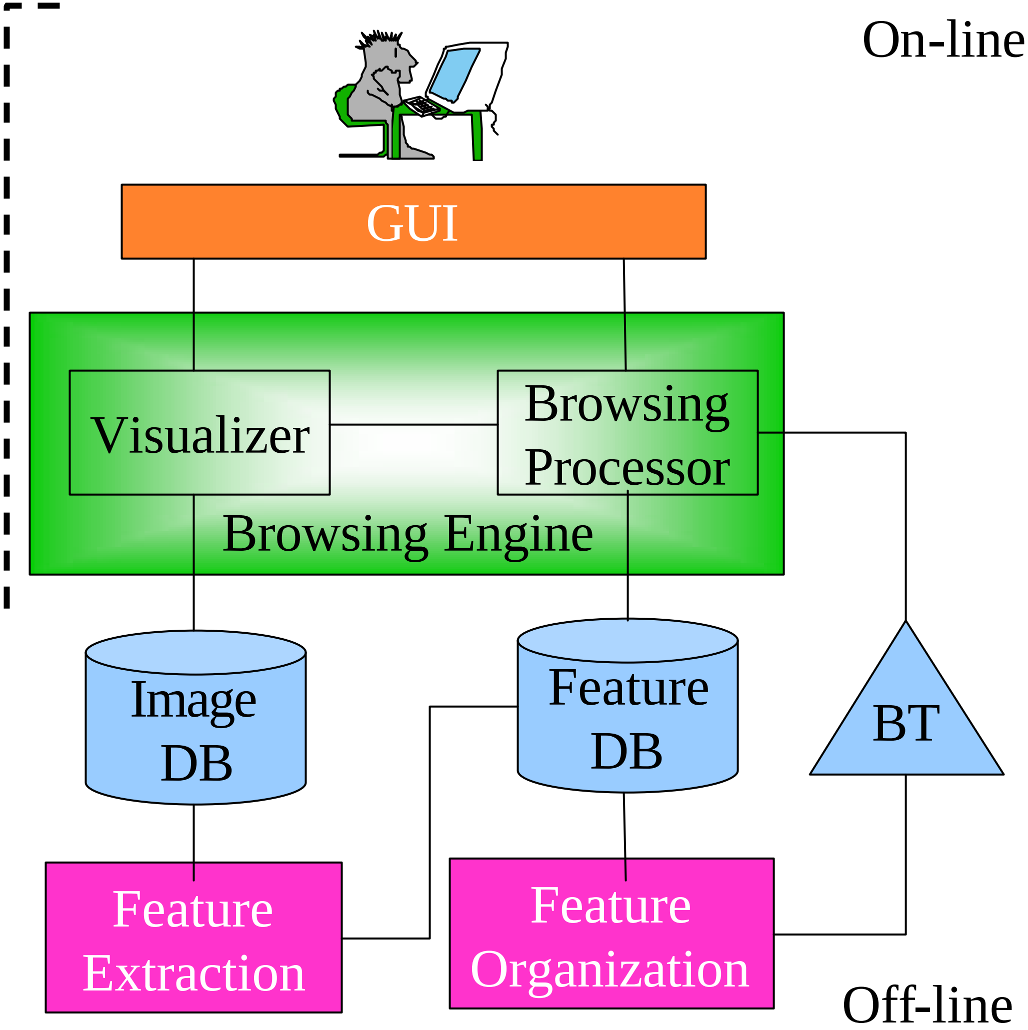
- Several browsing systems exist
  - Information considered to compute the similarity between images:
    - *automatically extracted low-level features (e.g. color, texture, etc.)*
    - manually extracted semantic concepts
    - both
  - Typically, they provide an hierarchical view of the image DB
- Problems that limit their applicability to large image DB's:
    - based on static browsing structure
    - providing customization facilities that need to reorganize a large part of the DB
    - no persistent updating of the browsing structure

PIBE (Personalizable Image Browsing Engine)

- A novel adaptive image browsing engine
  - customizable hierarchical structure called *Browsing Tree* (BT)
  - graphical personalization actions to modify the BT
  - "local" reorganization of the DB
    - specific similarity criteria for each portion (sub-tree) of the BT
  - user customizations persist across different sessions

PIBE Architecture

- PIBE builds *off line* the BT from the features
- At *run time* the user interacts with the system by means of a GUI
- Browsing Engine:
  - Browsing Processor manages requests to modify the BT
  - Visualizer displays the BT's content



Principles of PIBE

- Three main ingredients behind the BT:
  - image descriptors (e.g. color histograms)
    - representable as points in a N-dimensional space
  - (dis)similarity functions (e.g. weighted Euclidean)
    - instance of a parametrized class of functions to support personalization actions
  - clustering algorithms (e.g. *k-means*)
    - BT is a hierarchical structure derived from a hierarchical clustering algorithm or, alternatively, by recursively applying a partitioning algorithm

- PIBE is parametric with respect to the choices of above points. The combination of choices should guarantee:
  - scalability with respect to the cardinality of the image DB
  - suitable cardinality of each cluster

Browsing Tree

- PIBE uses:
  - 32-D HSV color histograms ($\mathbf{p}$)
  - weighted Euclidean distances

$$d(\mathbf{p},\mathbf{q};\mathbf{w}) = \left(\sum_{i=1}^{32} w_i (p_i - q_i)^2 \right)^{1/2}$$

  - *k-means* algorithm applied to the whole image DB and, recursively, to each of the derived *k* clusters producing a default BT (w_i = 1)
  - each node of the BT corresponds to a cluster C_j of images and maintains the
    - centroid $c(C_j)$ of C_j
    - representative image $p(C_j)$ of C_j defined as

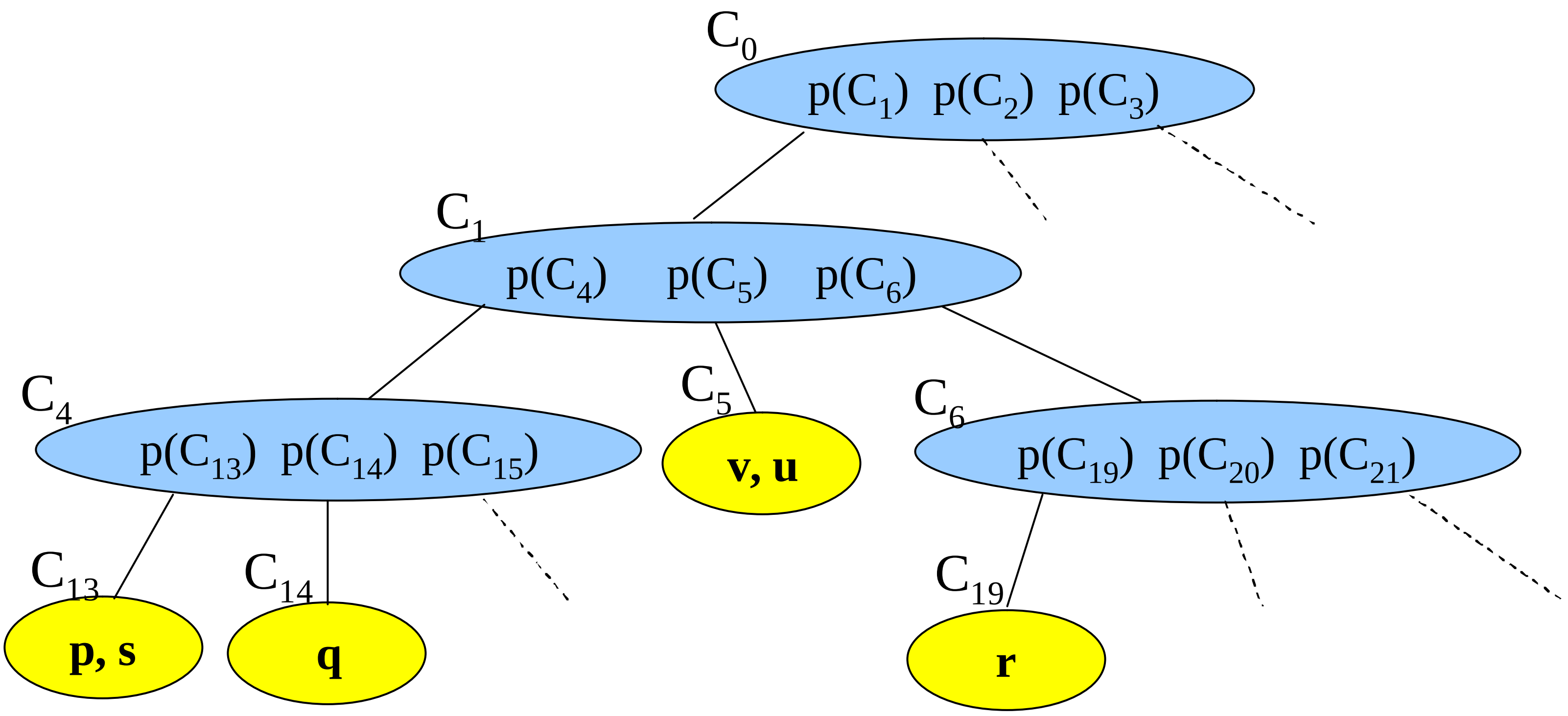
$$p(C_j) = \arg\min \{ d(\mathbf{p}, c(C_j); \mathbf{w}_j); \mathbf{p} \in C_j \}$$

    - local weight vector $\boldsymbol{w}_j$ computed as $w_{j,i} \propto 1/s_{j,i}^2$

Browsing Tree Example (k=3)

n = number of images to be clustered

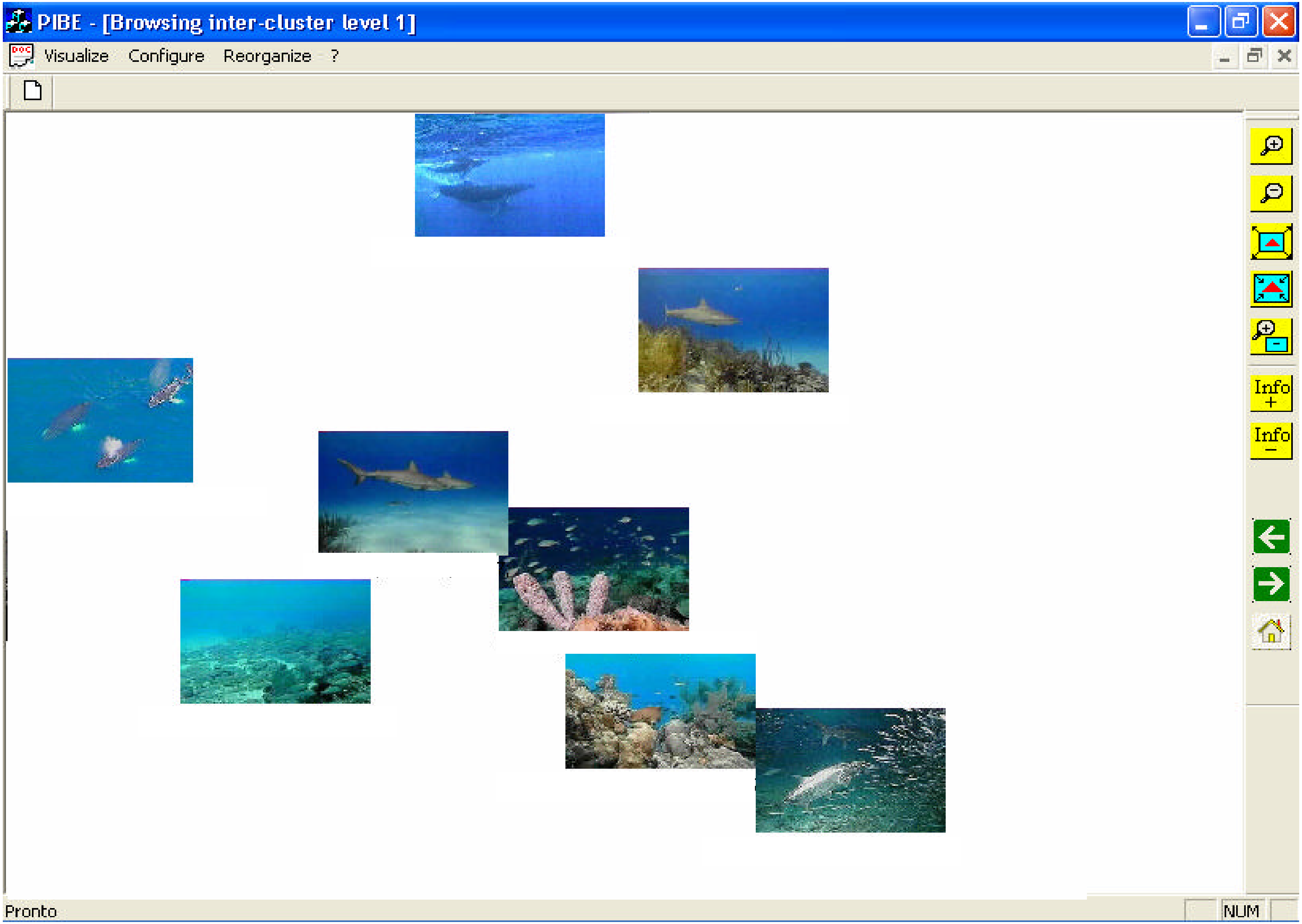
O(*kn log$_k$n*)



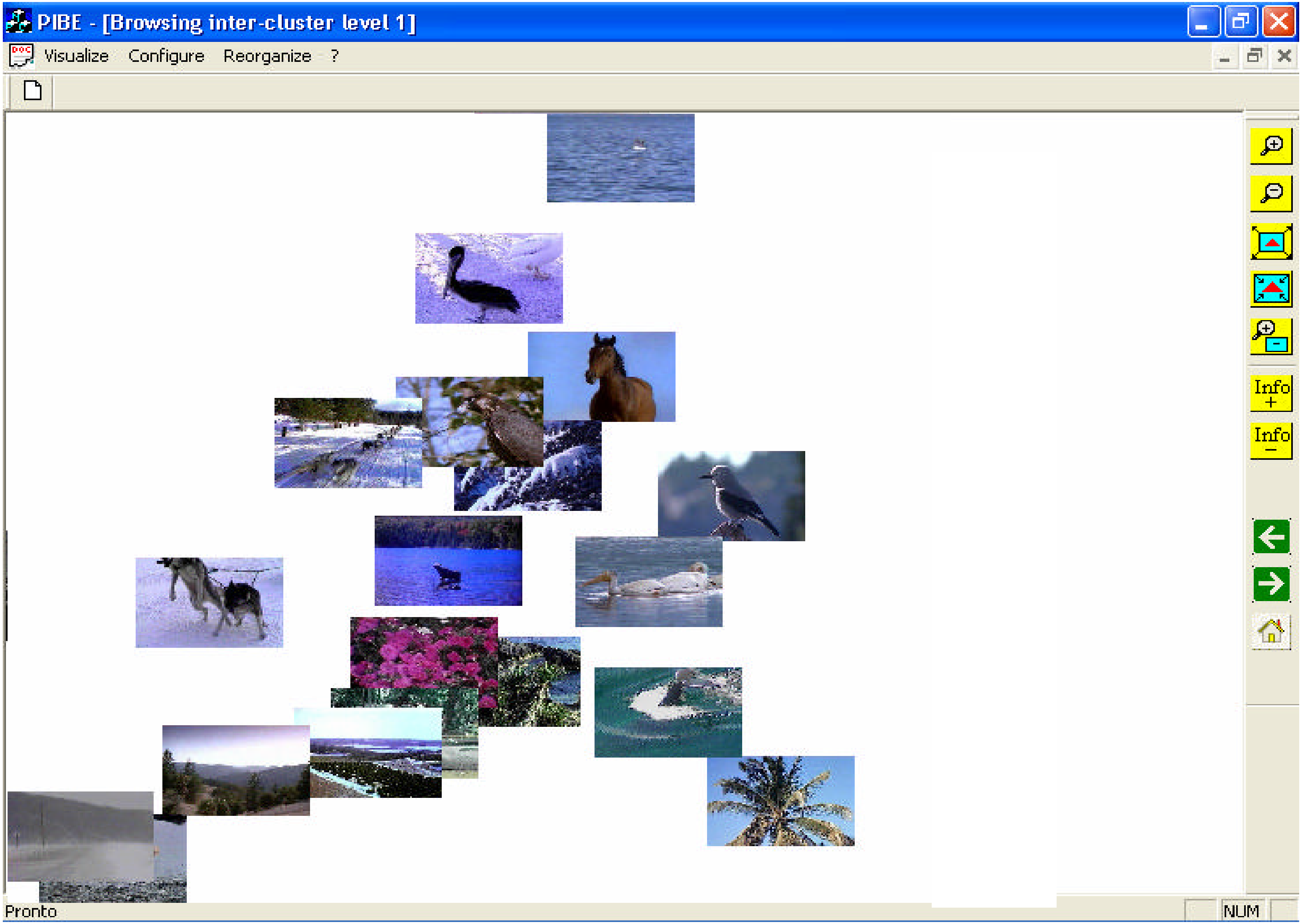
Browsing

- Visualizer and Browsing Processor implement the navigation mechanism
  - images displayed in a 2-D screen (spatial visualization technique) using relative distances
  - to display images **p** and **q** the weighted Euclidean distance corresponding to the most specific cluster containing both images is used
- Two modalities:
  - vertical: the user selects a representative image on the display and *zooms in* the cluster content
  - horizontal: the user explores regions of the space where no representative image is present

Browsing Examples



Browsing Examples



Personalization Facilities

- PIBE provides graphical facilities allowing the user to move a "source" cluster representative image $p(C_s)$ on a "target" one $p(C_t)$
- Asymmetric operations
  - C_t is the "leader"
- Two different actions
  - "moving image"
    - $p(C_s)$ is considered as single image
  - "moving cluster"
    - $p(C_s)$ represents the whole cluster C_s

Updating the BT

- Browsing Processor is in charge of the BT updates (we consider the more demanding case of moving cluster action)
  - $\mathbf{w}_t$ is recomputed ($C_t = C_t \cup C_s$)
  - C_s is deleted from the BT
  - *k-means* algorithm is recursively applied on the updated $C_t = C_t \cup C_s$ using $\mathbf{w}_t$ (O(k $|C_t|$ $log_k|C_t|$))
  - representative images of clusters that contained C_s need to be updated
    - if C_t and C_s have the same parent C_p, $p(C_p)$ does not change
    - if C_t and C_s have different parents C_{pt} and C_{ps} (but the same grand-parent C_{gp}), only $p(C_{pt})$ and $p(C_{ps})$ have to be updated

Experimental Evaluation

- 2,000 color images of IMSI-PHOTOS data set
- Goal: "given a set G of target images (in our experiments, 51 "fishes"), evaluate how PIBE is able to reflect user's needs in localizing them"
- Scenarios involved:
  - Default: this strategy uses the default BT
  - PIBE: a customized version of the BT is considered, following two ways to insert the target images of G into a single target cluster C_t
    - Top-down (C_t is an internal node that contains a large number of target images)
    - Bottom-up (C_t is the leaf node containing the highest number of images in G)

Effectiveness

- Computed as precision (P) of the target cluster C_t with respect to the set G of target images
- P is the the number of target images in C_t over the cardinality of C_t

$$P = | C_t \cap G | / | C_t |$$

n. actions	n. images	n. relevant images	P (%)
0	34	23	67.65
1	62	48	77.42
2	80	51	63.75

Top-down strategy (k=11)

n. actions	n. images	n. relevant images	P (%)
0	2	2	100
10	22	22	100
20	34	34	100
30	52	51	98.08

Bottom-up strategy (k=11)

Efficiency

- Computed as time (T) of a browsing session, i.e. number of mouse clicks needed to reach all target images by means of the PIBE GUI
- We computed the saved-time (ST) as

$$ST = (1 - T_{\mathbf{PIBE}}/T_{\mathbf{Default}})*100$$

where $T_{\mathbf{PIBE}}$ is the time needed using a customized BT, whereas $T_{\mathbf{Default}}$ is that associated to the default BT

strategy	$T_{\mathbf{Default}}$	$T_{\mathbf{PIBE}}$	ST (%)
Top-Down	55	29	47.27
Bottom-up	55	20	63.64

(k=11)

Conclusions & Future Work

- PIBE as a new solution to image browsing:
  - personalization graphical actions
  - hierarchical structure (Browsing Tree)
  - persistent and local customizations

- We are investigating several issues:
  - variable fan-out for Browsing Tree nodes
  - “split” actions
  - integration of semantic concepts extracted by textual descriptors (using lexical ontologies, e.g. WordNet)