Analysis and Representation for Automatic Comparison and Retrieval of Digital Images

Nicholas R. Howe

May 7, 2001

Why Image Retrieval?

- Many applications:
  - World Wide Web.
  - Video libraries.
  - Digital photography.
- Appealing research:
  - Image segmentation.
  - Object recognition.
  - Image understanding.

Road Map

- Motivation
  - Pushing the abstraction barrier
- Vector method (Stairs)
  - Representation & comparison
  - Evaluation
- Partial-image (object) queries
- Conclusion

Image Retrieval Framework

User supplies a *query image*.

Diverse collection of images.

System ranks images

– Returns most similar images from collection.

Levels of Similarity

- Similarity can occur at different levels of abstraction:

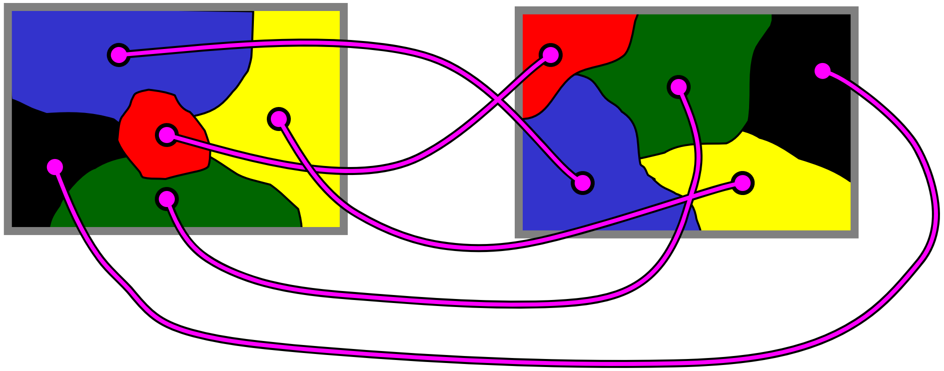
Red

Horses

France

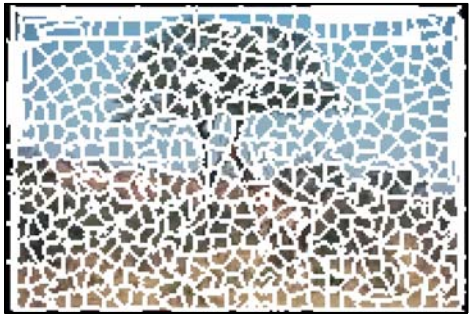
Motivation & Approach

- Image Similarity ⇒ Retrieval.
- Area Matching Approach:

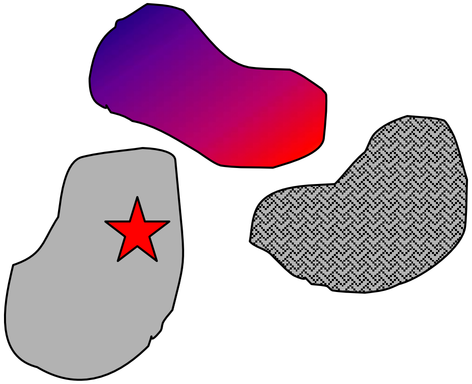


- Compare regions in terms of color, crude texture, and location.

A Region-Based Representation



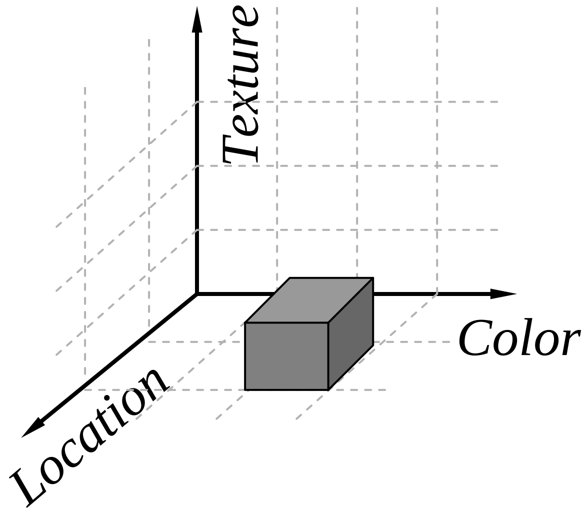
- Begin with segmentation.
  (Provides locality.)
- Describe each patch using multiple features.
  - Color
  - Texture
  - Location



- Combine such that each piece is preserved.

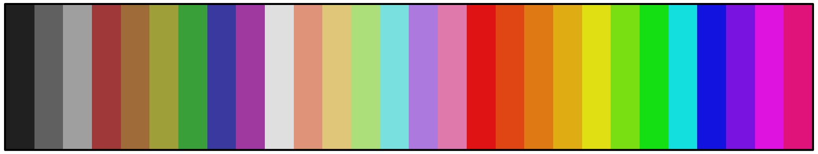
Stairs: Parts Describe the Whole

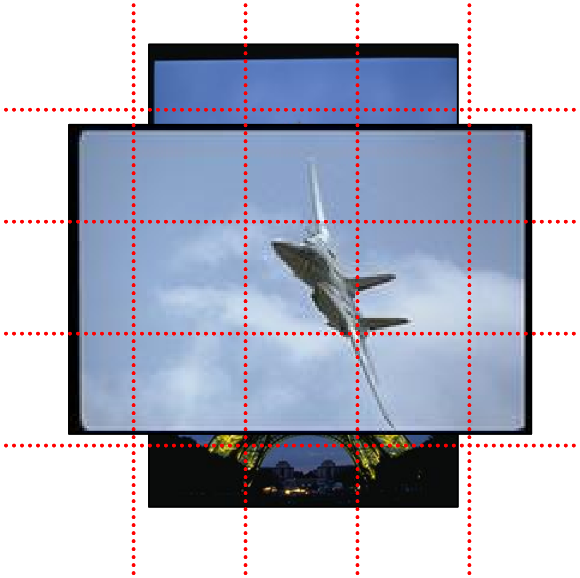
- Discretize the range of each feature.
  (Color, texture, and location)
- Count patches in image described by each combination of features.
  - Blue-Smooth-TopLeft: 5,
    Blue-Smooth-TopMiddle: 1,
    …
    Green-Smooth-TopLeft: 0, etc.



Discretization

- Color: 28 bins
- Texture: 3 bins
  (smooth, textured, rough)
- Location: 25 bins





- Total: 28×3×25
  = 2100 combinations

Vector Representation (Stairs)

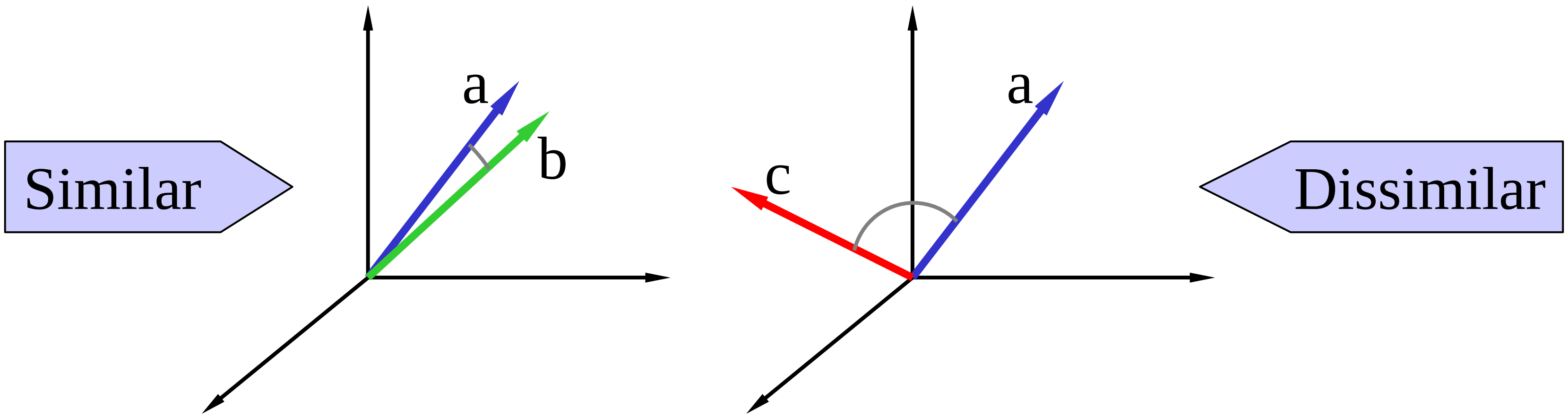
- Final representation of image is a vector with 2100 dimensions.

$$\mathbf{v} = \left\langle v_{c_1 t_1 l_1}, v_{c_1 t_1 l_2}, \ldots, v_{c_1 t_1 l_{25}}, v_{c_1 t_2 l_1}, \ldots, v_{c_{28} t_3 l_{25}} \right\rangle$$

- Each dimension records how much of a particular type of material is present.
  - e.g., how much smooth blue in the top left corner?

Comparison

- Vectors are points in space.
- Images with similar composition will have similar (normalized) vectors.
- Angle between similar vectors will be small.



Comparison (2)

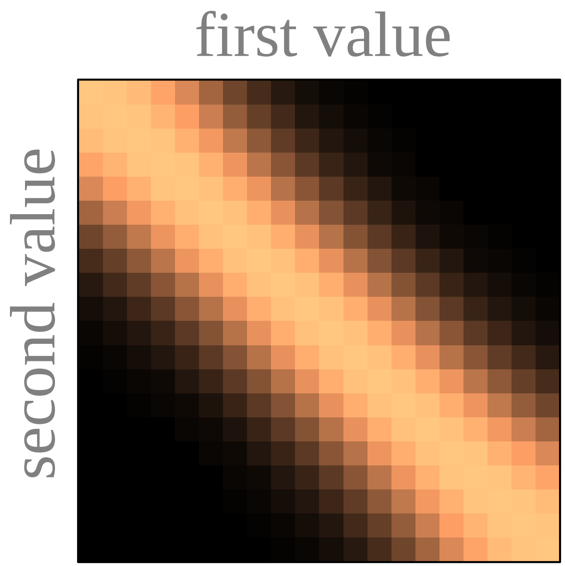
- Compare two images using a cosine metric:

$$D(\mathbf{v_1}, \mathbf{v_2}) = \cos^{-1}\left(\frac{\mathbf{v_1^T S v_2}}{\sqrt{(\mathbf{v_1^T S v}_1)(\mathbf{v_2^T S v_2})}} \right)$$

- Note generalization using **S** matrix:
  - **S** = **I** is standard cosine metric.
  - Other values of **S** allow adjustments to metric.

Comparison: Match Coefficients

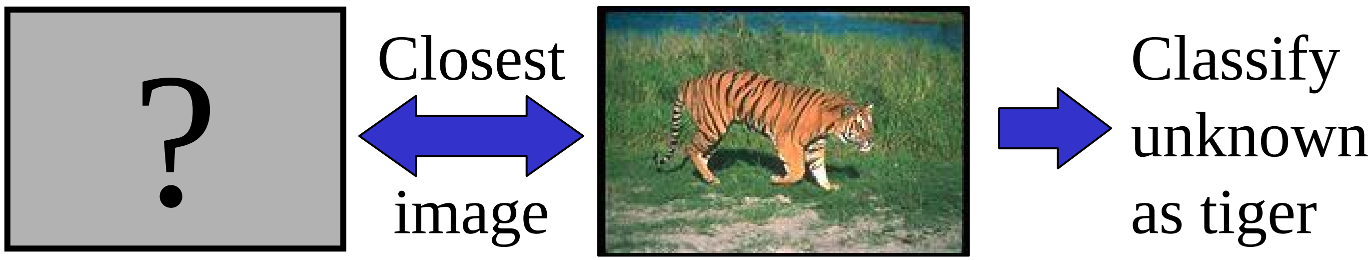
- Discretization of features loses some similarity information.
  - e.g., ***Blue*** is closer to ***Green*** than to ***Orange***.
- Such partial matches may be encoded in off-diagonal terms of **S**.



(lighter ⇔ more similar)

Evaluating the Vector Method

- Evaluations should test realistic conditions.
- Traditional method: Classification set.
  - 12 & 16 categories of ~100 images each.



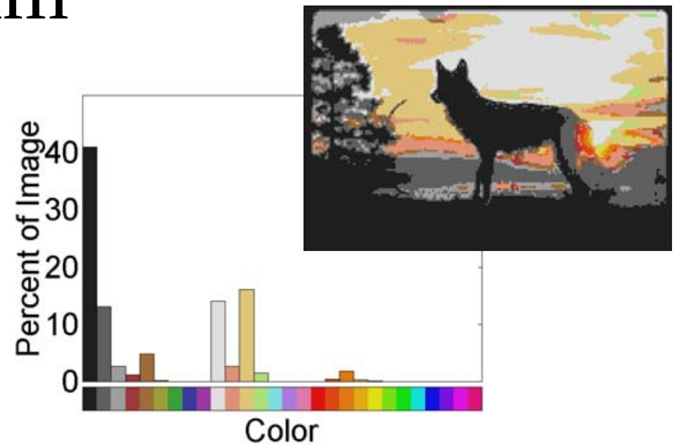
- New method: Altered-image queries.

Comparison Methods

- Statistical methods in common use:
  - Color Histograms

    (Swain & Ballard 1991)
  - Banded Autocorrelogram

    (Huang et. al. 1997)

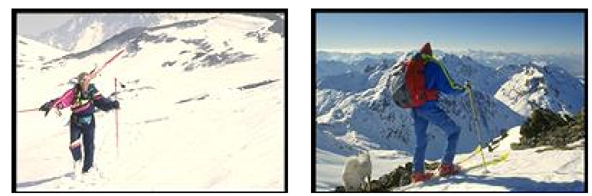


Sample Categories

Airshows

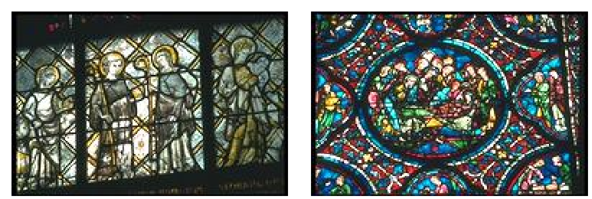
Caves

Elephants



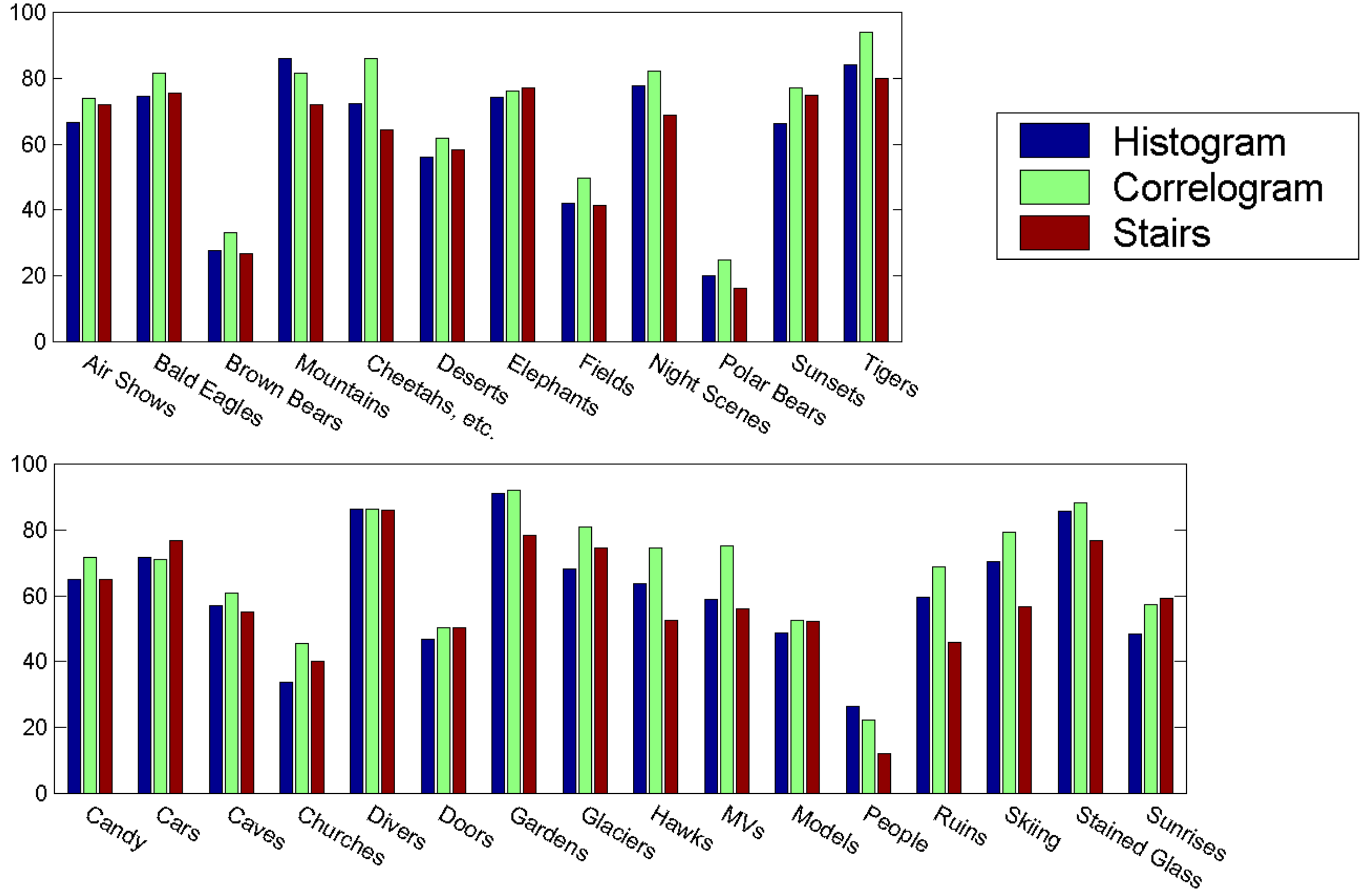
Skiers

Polar Bears



Stained Glass

Classification Results



Artificial Queries

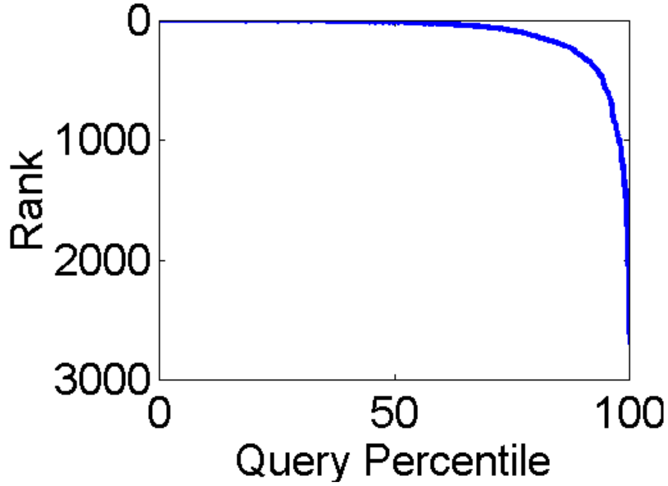
- Image → Altered Image

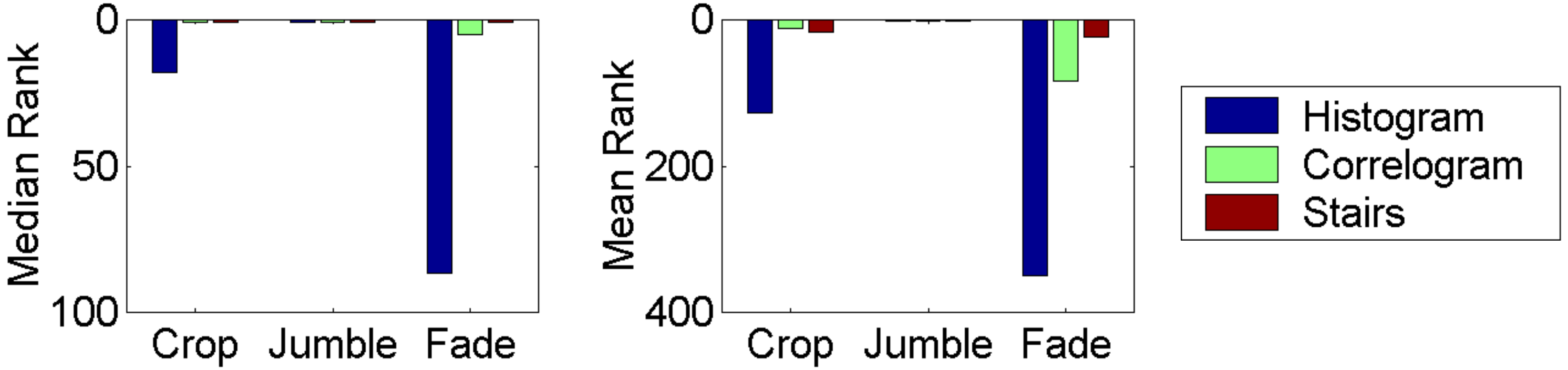
Original Crop Jumble Fade

- Goal: Locate original in library, using altered image as query.

Altered-Image Results

- Most images retrieved at low rank. (Good!)
- Minority of images retrieved at high rank.





Object Queries

- Something we've wanted to do all along:

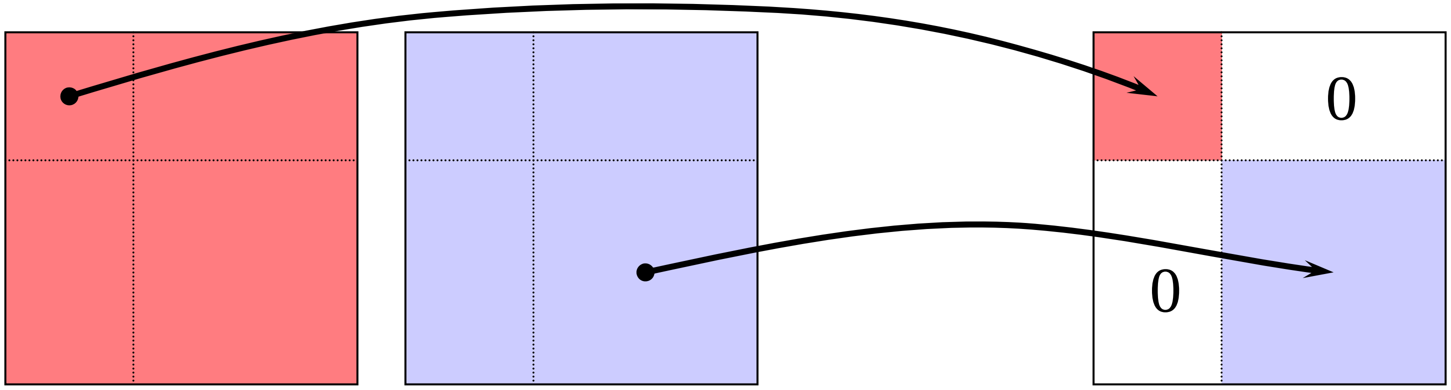
Search for objects, not whole images.

Rank 60
(of 19,000)

Rank 1
(of 19,000)

Object Queries, Method I (Feature-Space Division)

$$S(i,j) = \begin{cases} S_{object}(i,j) & \text{if i and j appear in the target object.} \\ S_{background}(i,j) & \text{if neither i nor j appear in the target.} \\ 0 & \text{otherwise.} \end{cases}$$

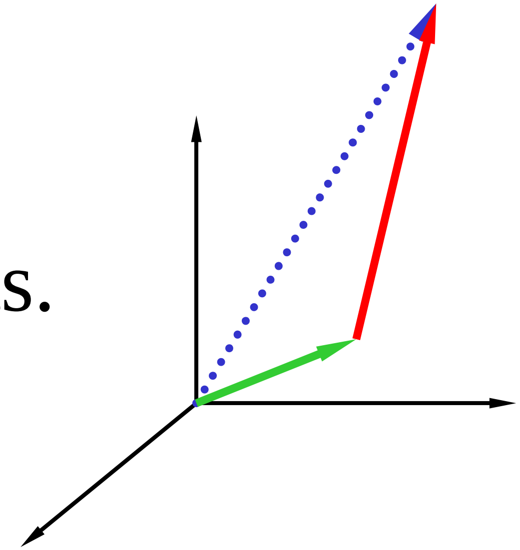


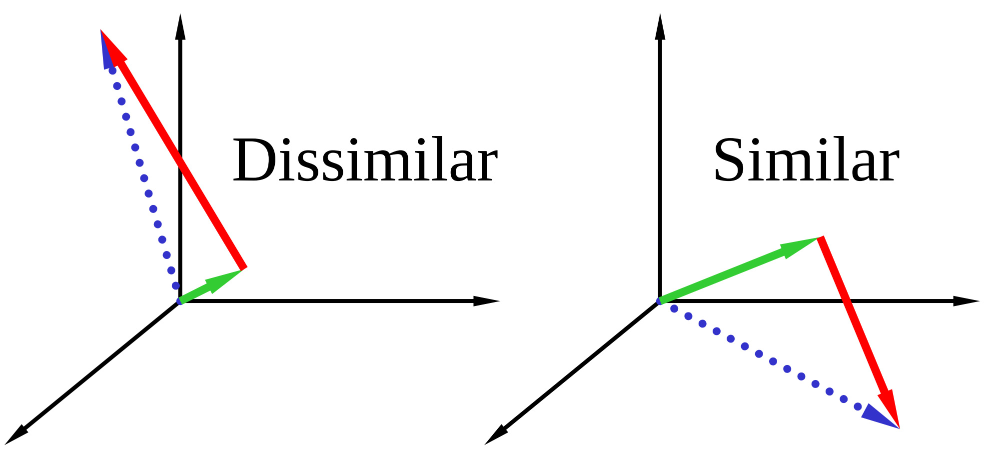
S_{object}
(strict match)

$S_{background}$
(loose match)

Object Queries, Method II (Vector Components)

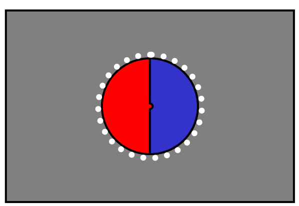
- Image vector includes components from both object & background.
- Idea: search for other vectors with similar object components.

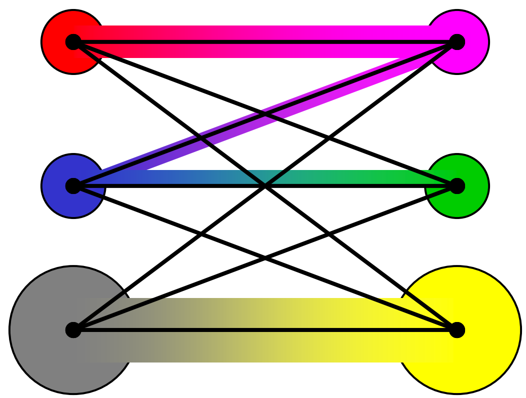


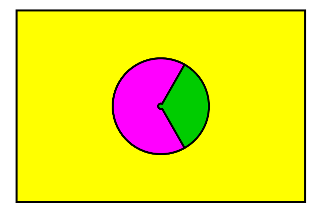


Object Queries, Method III (Explicit Area Matching)

- We can explicitly match the regions in the query area, using minimum-cost flow.


Query




Target

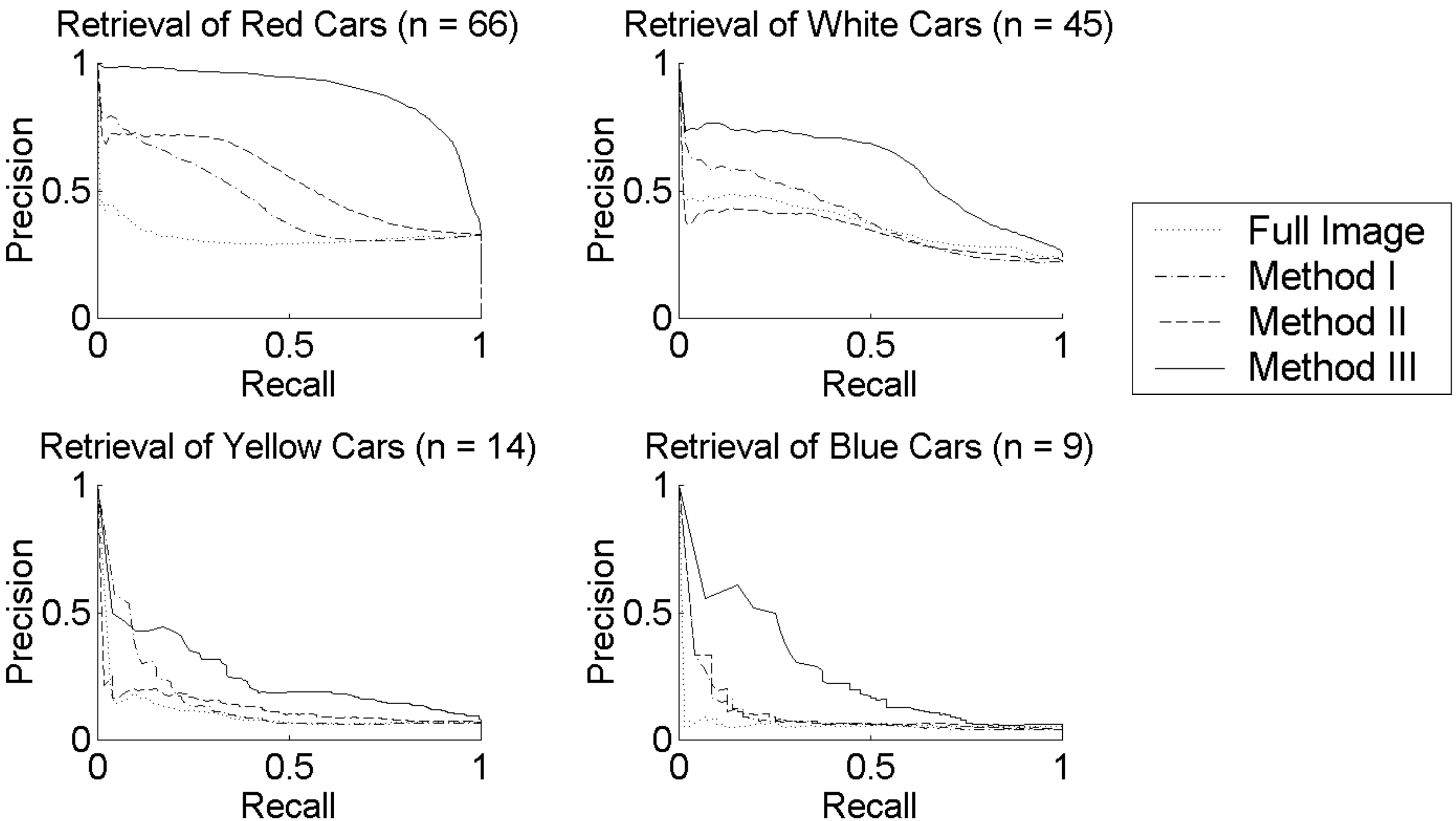
Supply & Demand $\propto$ Area

Cost $\propto$ Quality of Match

Testing Object Queries

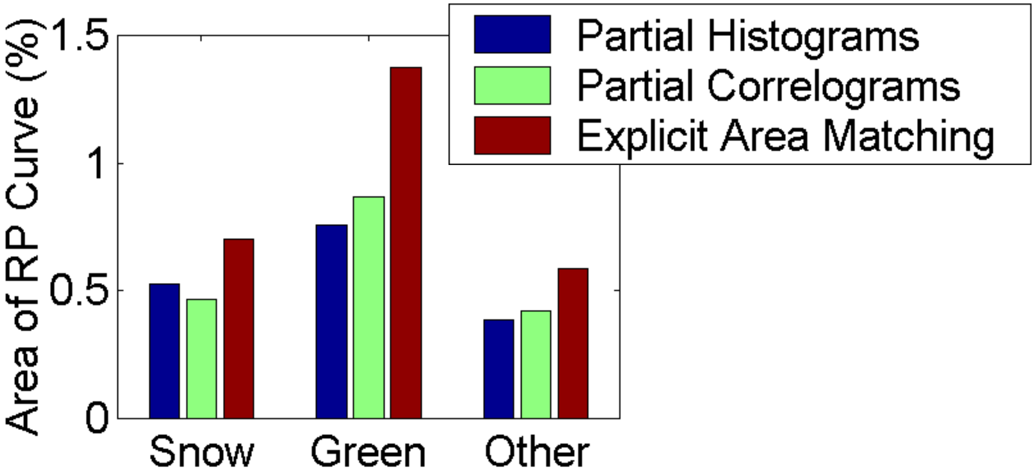
- 200 images of cars
  - Visual context is irrelevant.
  - Classes are colors of car.

Object Query Results



Object Queries in Large Libraries

- Three sets of wolf images
- One set used as query for others.
- Method III vs. other techniques:



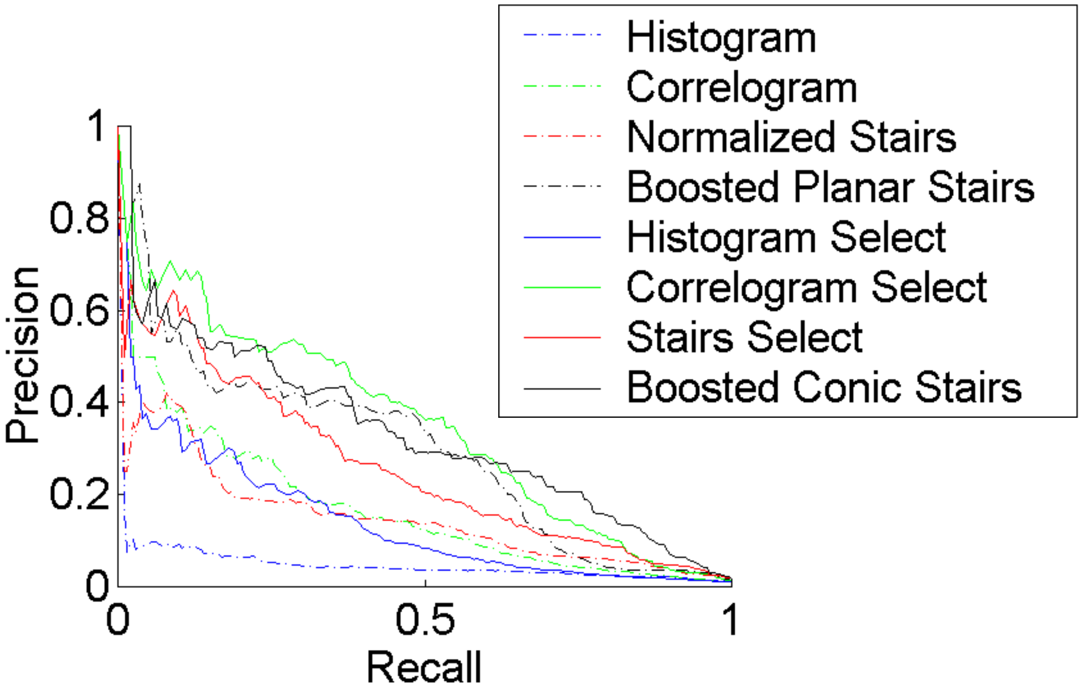
Snow Wolves

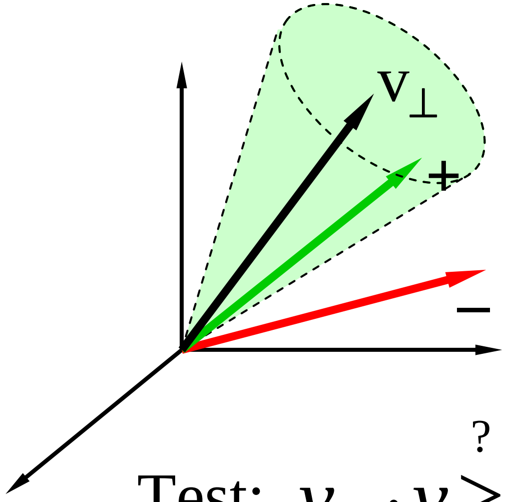
Green Wolves

Other Wolves

Boosting & Stairs

- Conic separators used as base learner:





Test: $v_\perp \cdot v \overset{?}{\underset{?}{\geq}} k$

Work Accomplished

- Identified need for object-conscious image representation.
- Developed Stairs representation & related algorithms.
- Assessed in comparison with existing techniques: mostly competitive.
- Flexible use of regions allows search for objects & arbitrary figures of interest.

Work Awaiting

- Region descriptions are impoverished.
  - Shape matters.
  - Texture is subtle.
  - Relative positions are important.

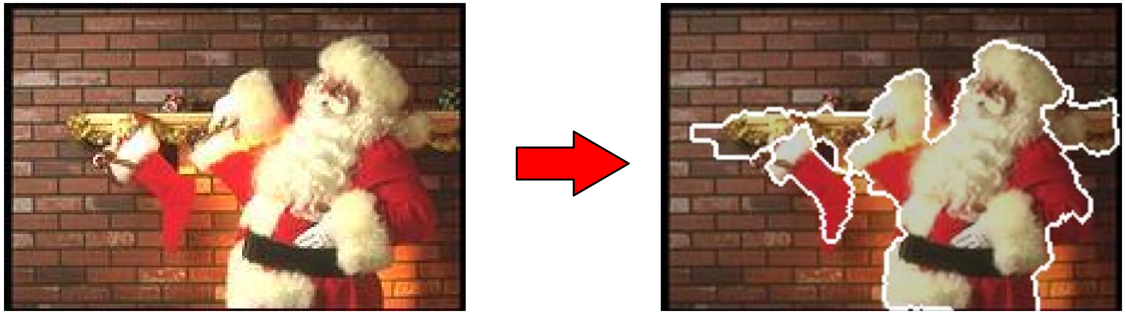
- Low-level reliability must improve.

Related Work

- Vector Representation
  - Howe & Huttenlocher, 2000; Howe, 2000; Howe 1998
- Earth Mover's Distance
  - Cohen, 1999
- Blobworld (UC Berkeley)
  - Carson et. al., 1999; Belongie et. al., 1997
- Netra (UCSB)
  - Deng & Manjunath 1999; Ma & Manjunath, 1997

Segmentation

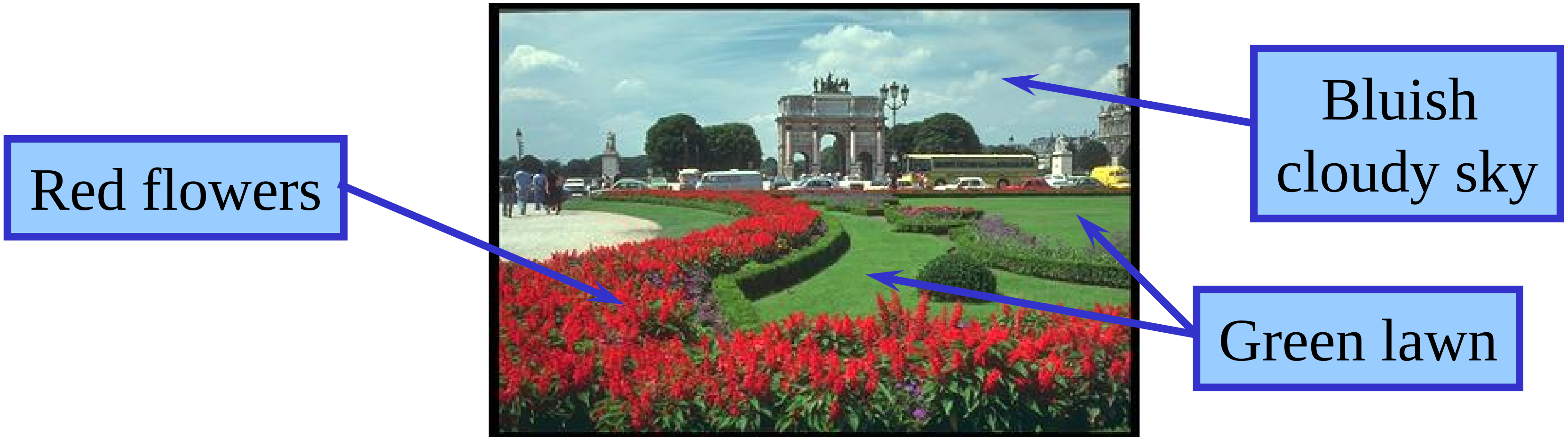
- Segmenting an image means dividing it into regions that “belong together.”



Q. What’s a sensible way to segment any given picture?

Characterizing Regions

- When humans segment an image, they can explain why each region hangs together.

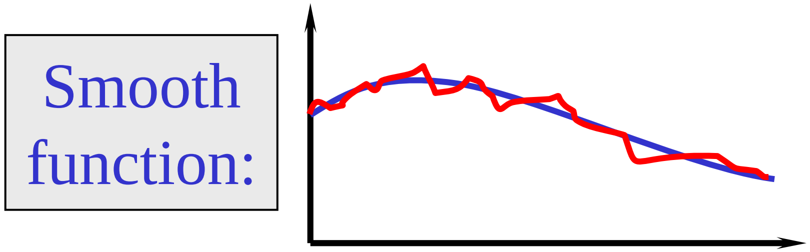


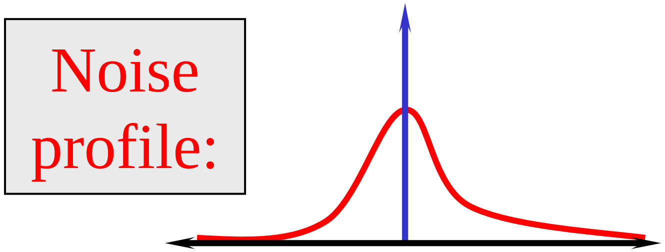
⇒ *Models motivate the grouping into regions.*

Mathematical Models of Regions

- Model regions as smooth functions + noise:

- 2D example:

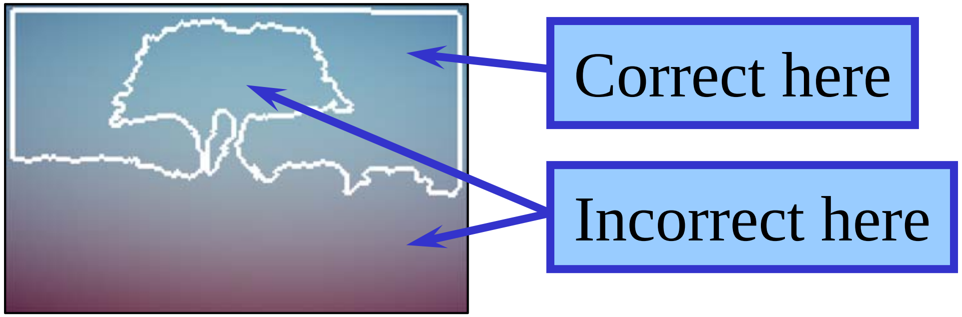




Models of Regions (2)

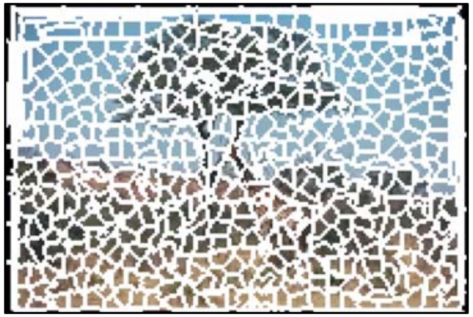
- Each model tries to predict the image.
- Successful models are rare.

Bad model



Good model

Outline of Segmentation Process



1. Start with small local regions.

   (Felzenszwalb & Huttenlocher 1998)
2. Create a pool of potential models.
3. Measure fit between all models & local regions.
4. Select a small number of models that fit many local regions well.

   (Details on the next slide)

Goal:

Segmentation Details

- Best segmentation found via energy minimization:

$$E(R) = \sum_{r \in R} Fit(r, M_r) + \sum_{r_1 \in R} \sum_{r_2 \in R} \Delta(r_1, r_2)$$

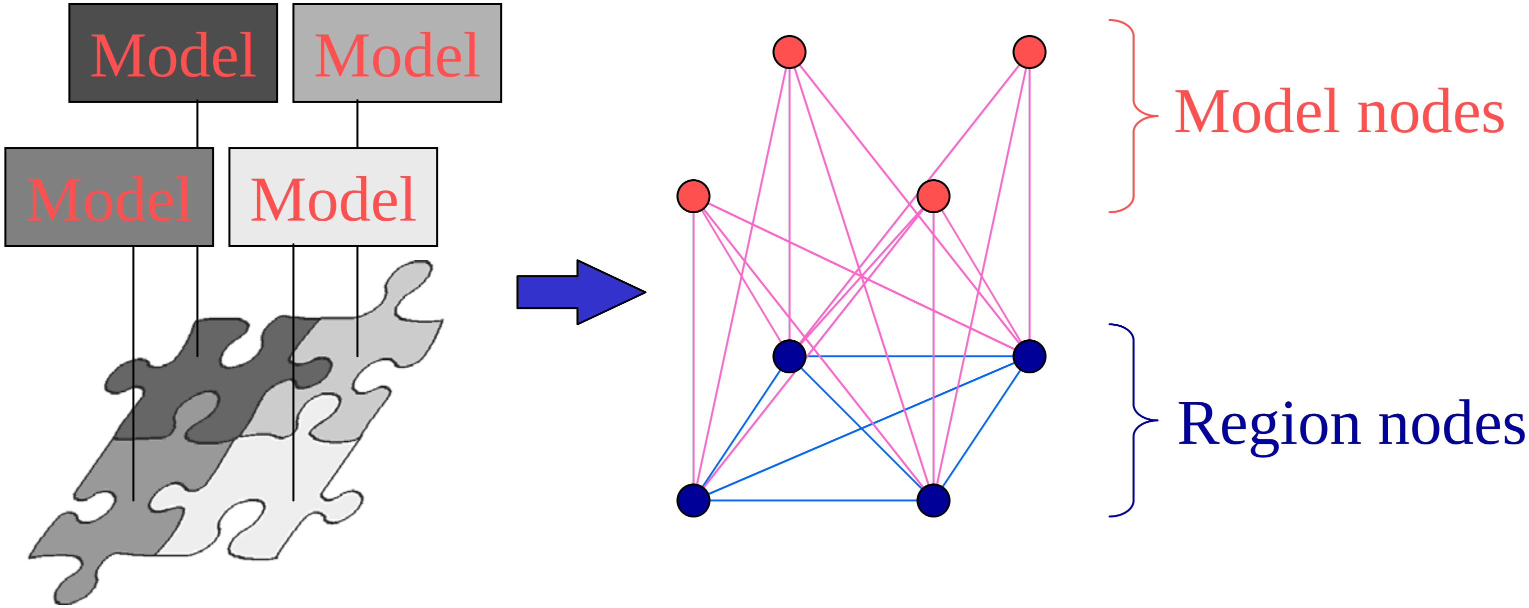
"The energy of a segmentation into regions R is equal to the fit of each region with its model plus a penalty to discourage excess regions."

- Minimum energy is difficult to compute in general.

Graph Formulation

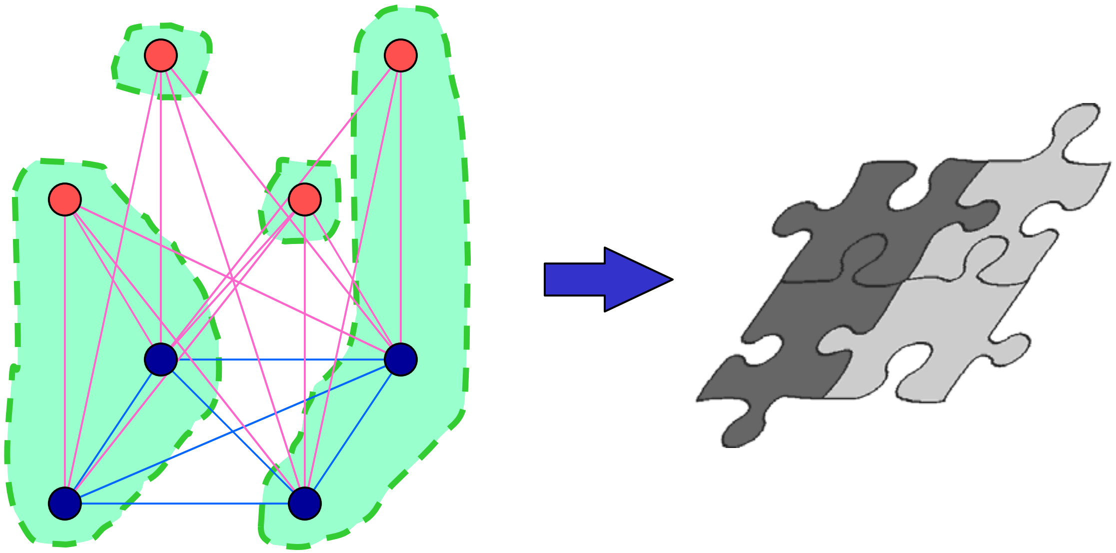
- Minimum energy = minimum graph cut

  (compare with Boykov, et. al., 1998)



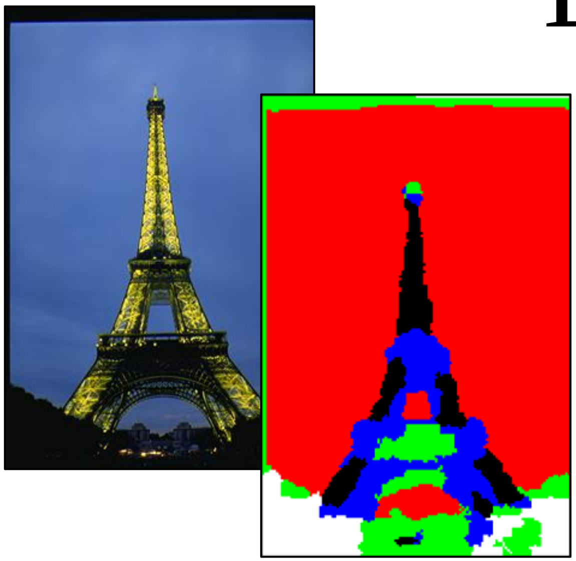
Graph Formulation (2)

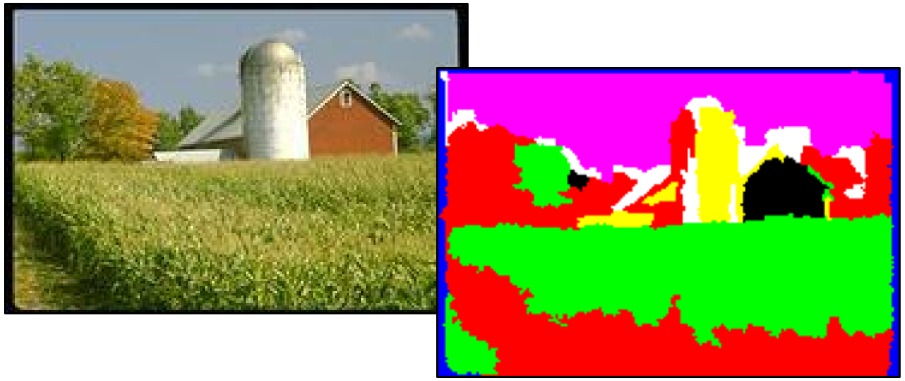
- Minimum graph cut = best segmentation

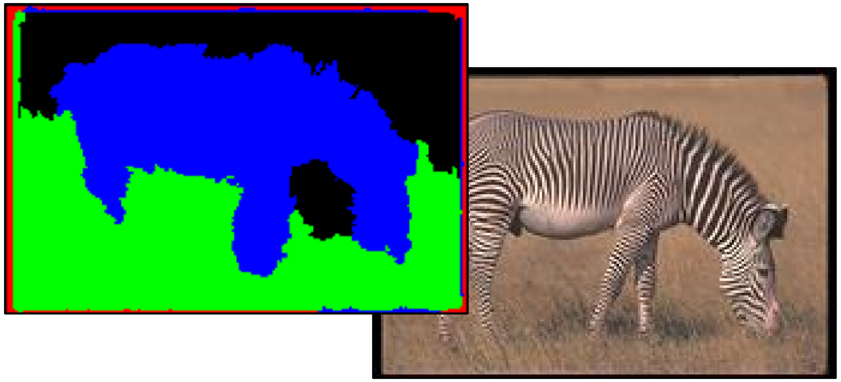


- Running time bound: quadratic in # of nodes
- Quality bound: Energy found is ≤ 2 × optimal.

Examples







Related Work

- Stereo Vision & Energy Minimization

  (Boykov, Vexler & Zabih, 1998)
- Normalized Cuts

  (Shi & Malik, 1997)
- JSEG

  (Deng, Manjunath, & Shin, 1999)

The Future

- Moving away from absolutism

  “OK, we can find red cars. Can we find *cars*?”

  – Relational encodings:

    - *White fur next to red velvet*
    - *A piece of all the same color*

- Interplay between segmentation, similarity, and compression/coding

  e.g., Color & texture from segment model

Challenges

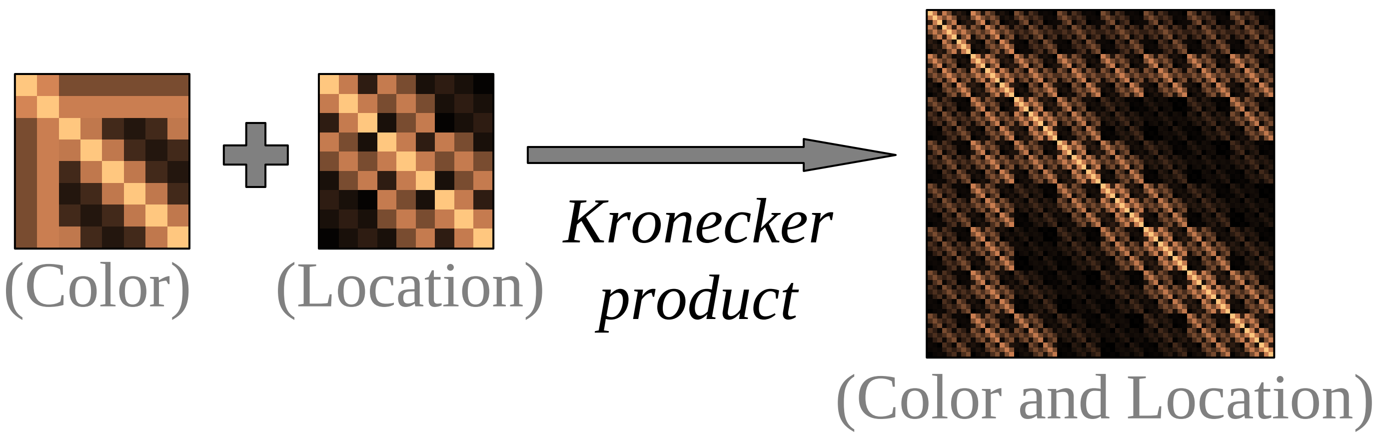
- Assumption: parts that belong together should look alike…

  …not always true!

- More sophisticated region models may help.

Generating the **S** Matrix

- **S** assembled from matrices $\mathbf{S}_j$ for each feature

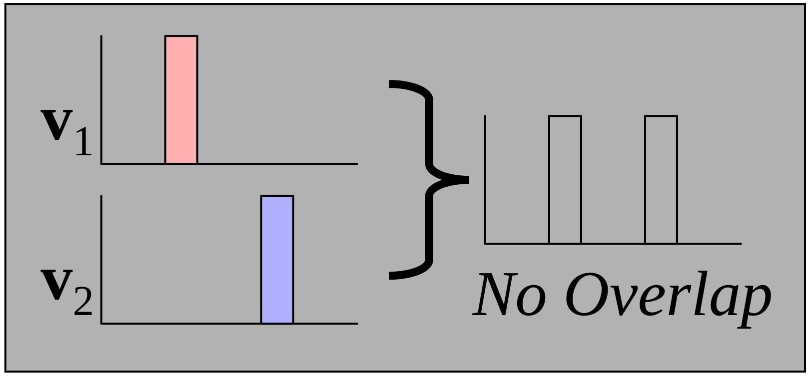


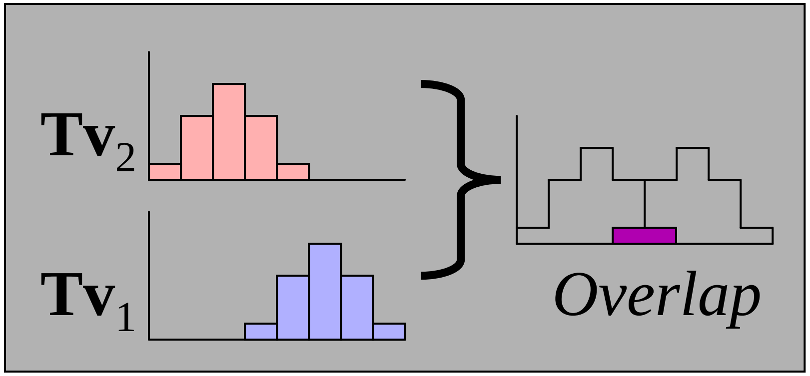
- Smaller matrices are determined by the similarity of the feature values.
  - e.g., ***Blue-Green*** vs. ***Blue -Orange.***

Alternate View of **S** Matrix

- Cholesky factorization of **S**: $\mathbf{S} = \mathbf{T}^{\mathrm{T}}\mathbf{T}$
- Cosine metric of modified vectors:

$$D(\mathbf{v_1}, \mathbf{v_2}) = \cos^{-1}\left(\frac{(\mathbf{Tv}_1)^{\mathrm{T}}(\mathbf{Tv}_2)}{\left((\mathbf{Tv}_1)^{\mathrm{T}}(\mathbf{Tv}_1)\right)\left((\mathbf{Tv}_2)^{\mathrm{T}}(\mathbf{Tv}_2)\right)}\right)$$





Optimizations

- Similarity computation is linear in sparse vector **v**.

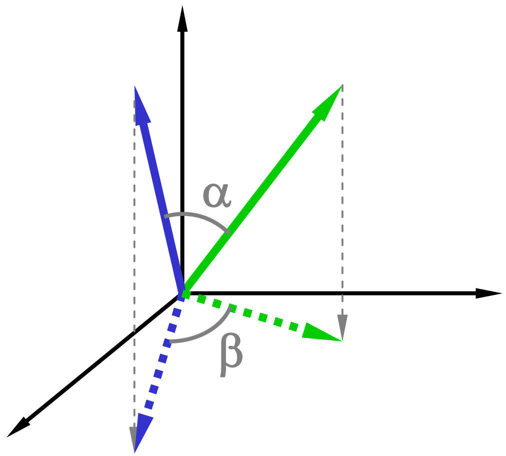
Computed once per query

$$D(\mathbf{v_1},\mathbf{v_2}) = \cos^{-1}\left(\frac{\mathbf{v_1^T S v_2}}{\sqrt{(\mathbf{v_1^T S v}_1)(\mathbf{v_2^T S v_2})}}\right)$$

Precompute & cache

Search Pruning

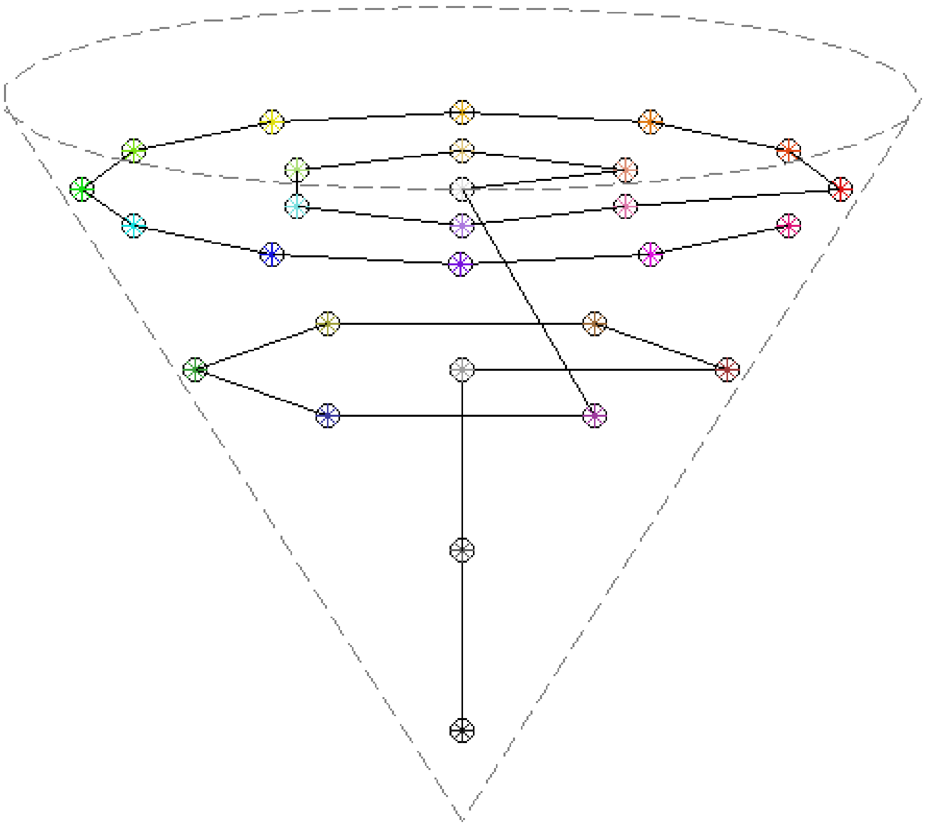
- Nearest neighbor search can be pruned by projection onto lower-dimensional spaces.



- β is lower bound on α.
- Images with β greater than some cutoff need not be considered.

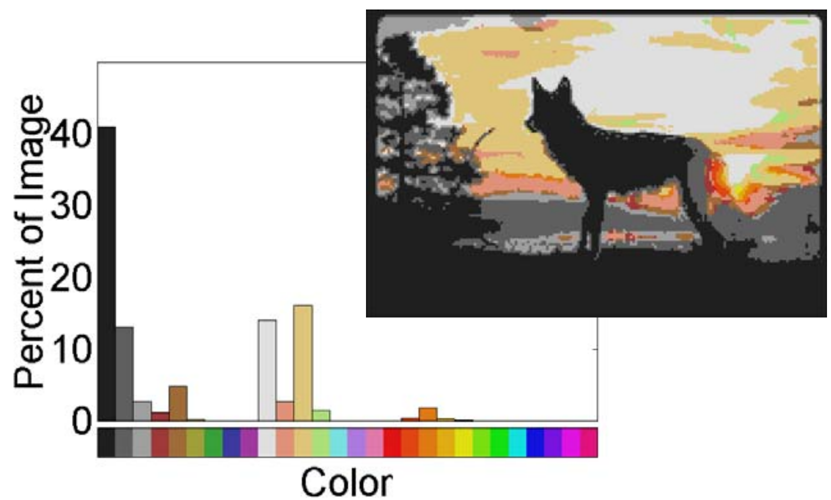
Dividing the Color Space

- Color seeds are dispersed evenly in HSV color cone.
- Divided into Voronoi regions.
- Ensures perceptual uniformity.



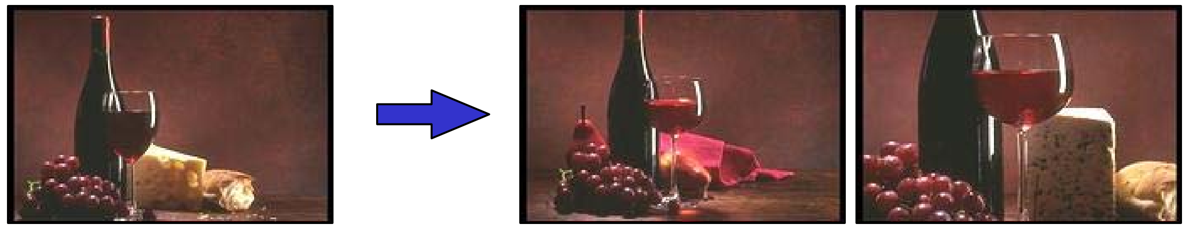
Color Histograms

- Simple & reliable.
- Limited extensibility.

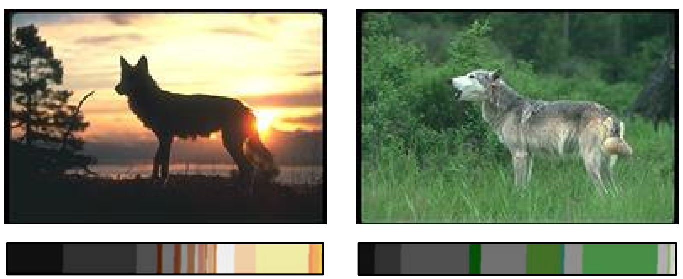


Color Histograms In Action

Successes…

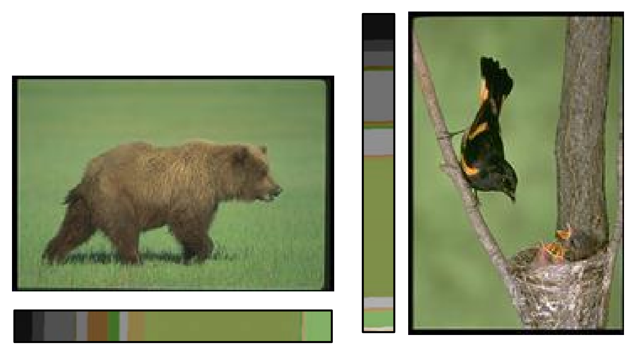


…and Failures:



• Some related images have very different histograms.

• Some unrelated images have nearly the same histogram.



Color Correlograms

- Correlograms consist of a table of probabilities.

$$C(x, y) = P\left(color(b) = x \,\middle|\, (color(a) = x) \wedge (\|a - b\| = y)\right)$$

	Red	Orange	Yellow	etc…
1 pixel	0.32	0.0	0.06	0.14
3 pixels	0.16	0.0	0.04	0.0
5 pixels	0.08	0.0	0.03	0.0

"Given a pixel of color *x*, the probability that a pixel chosen distance *y* away is also color *x*"

- Correlograms can be compared like vectors.