



Blobworld

Image segmentation using EM
and its application to image
querying [Carson et. al.]

Presented by: Nikhil V. Shirahatti

[Introduction:]

- Why do we need an image retrieval system?
- So does text retrieval method work for images?
- Content-based Image Retrieval...
- Three fundas: Feature extraction, multidimensional-indexing and retrieval system design.

[The “thing” world]

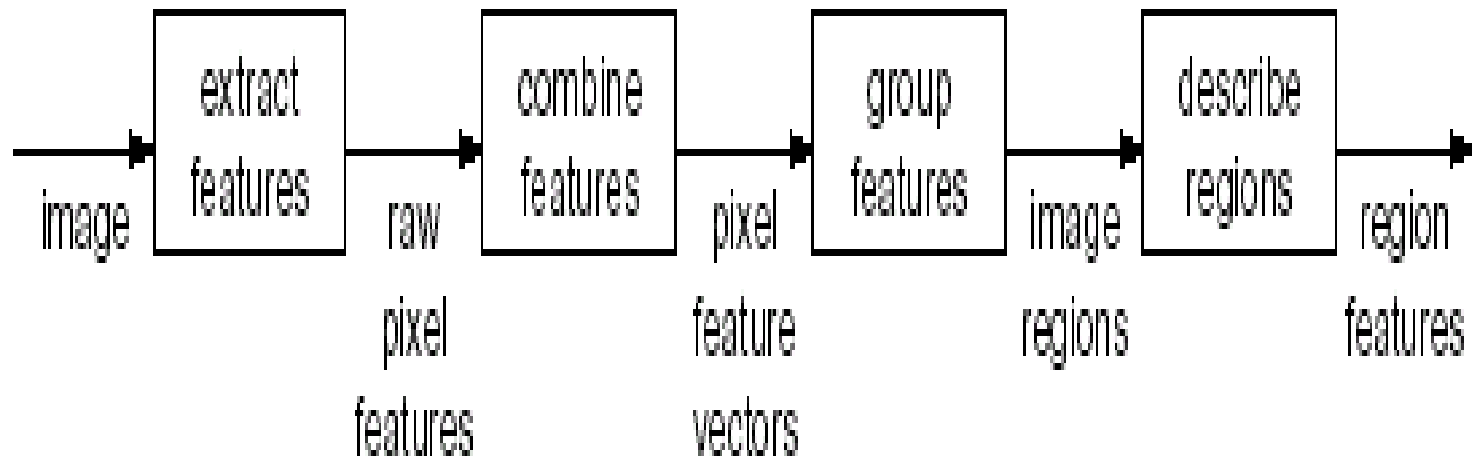
- Till recently: Low level features “stuff”
- Blobworld preaches:
 - “ Segmentation into regions and querying based on properties of these regions”
- Does it fare better than the “stuff” methods? ...later

“Stuff” methods

- Color histogram
- Color correlogram
- Wavelets

“None of these provides the level of automatic segmentation and user control to support OBJECT queries”

[Blobworld]



Stages of Blobworld processing

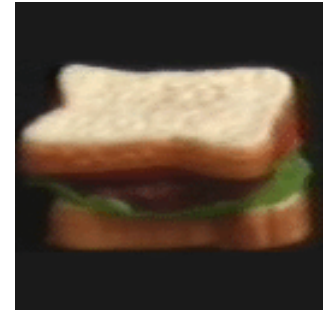
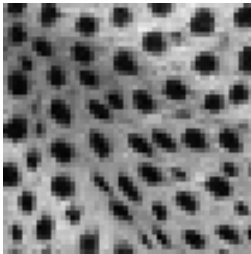
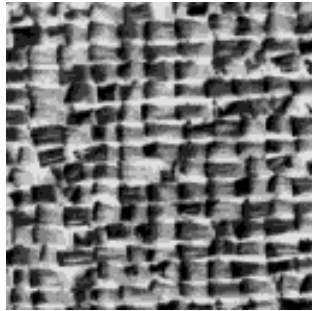
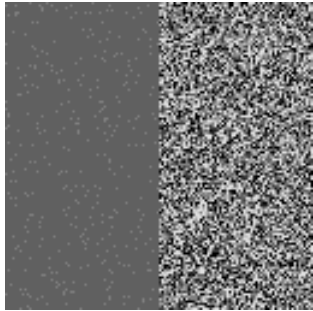
[Block 1: Feature extraction]

- Input: image | Output: pixel features
- What are the features?
- Algorithm:

“select an appropriate scale for each pixel, and extract color, texture and position features for that pixel “

- Color feature: L^*a^*b
- Texture features: discussed next...

[What is Texture?]



Source: *Principles and Algorithms of Computer Vision*
Fall 2002

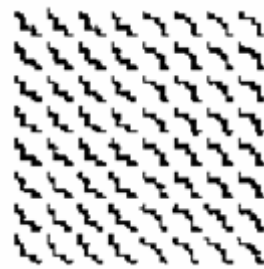
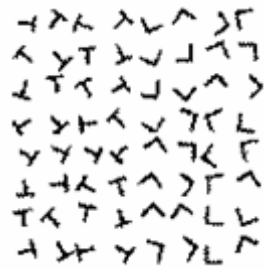
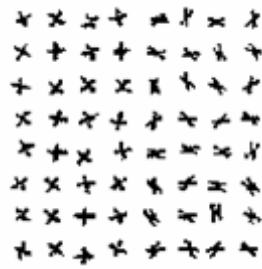
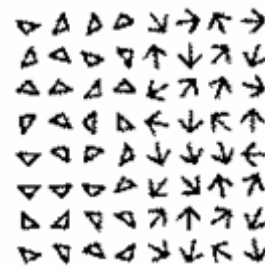
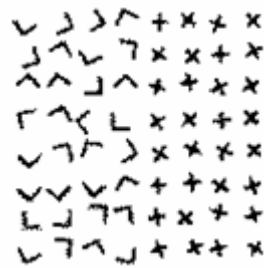
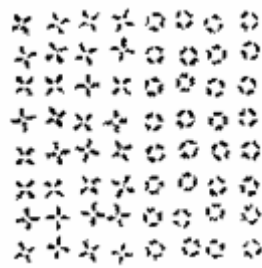
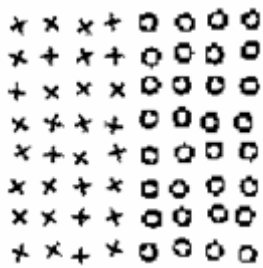
Department of Computer Science, Florida State University

Texture contd.

- Texture is a perceptual phenomenon
 - Whether an image is considered to be texture or not depends on the scale
 - For example, a single leaf is not considered as a texture
 - However, foliage of a tree is often considered to be a texture
 - Texture arises from a number of different sources
 - Examples include grass, foliage, brush, pebbles, and hair
 - Many surfaces with orderly patterns

[Texture contd.]

- Texture consists of organized patterns of quite regular sub-elements

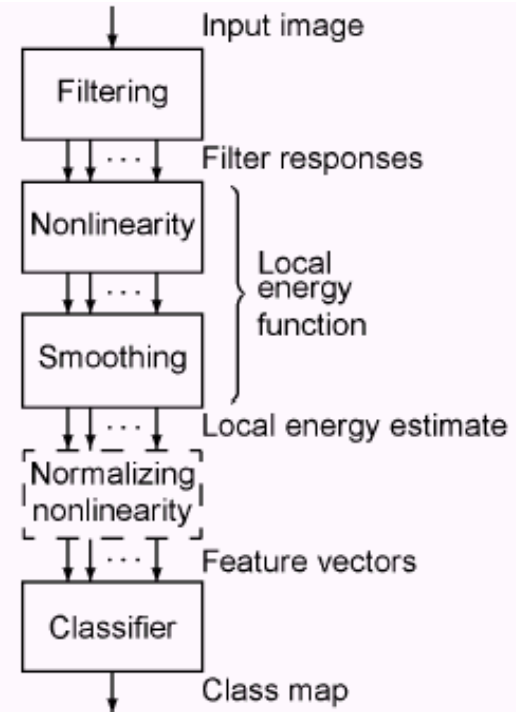


[Texture contd.]

A set of filtered images is not a representation of a texture.

- There are scales involved

- The scale of filters used.
- The scale to integrate filter responses to obtain a texture descriptor



Texture Contd.

Scale Selection

- Based on edge polarity?
- Texture feature (*ref: A framework for low level feature extraction- W. Forstner*).
- Measures of locally characterizing an image:
 - Intensity gradient: ∇g gradient of intensity along x and $y = (g_x, g_y)^T$
 - squared gradient: $\Gamma_g = \nabla g \nabla g^T$
 - $G_\sigma(x,y) = G_\sigma(x) * G_\sigma(y)$: **symmetric Gaussian**
 - Average squared gradient: $E(\Gamma_g * g(x,y)) = G_\sigma * \Gamma_g$

Texture..

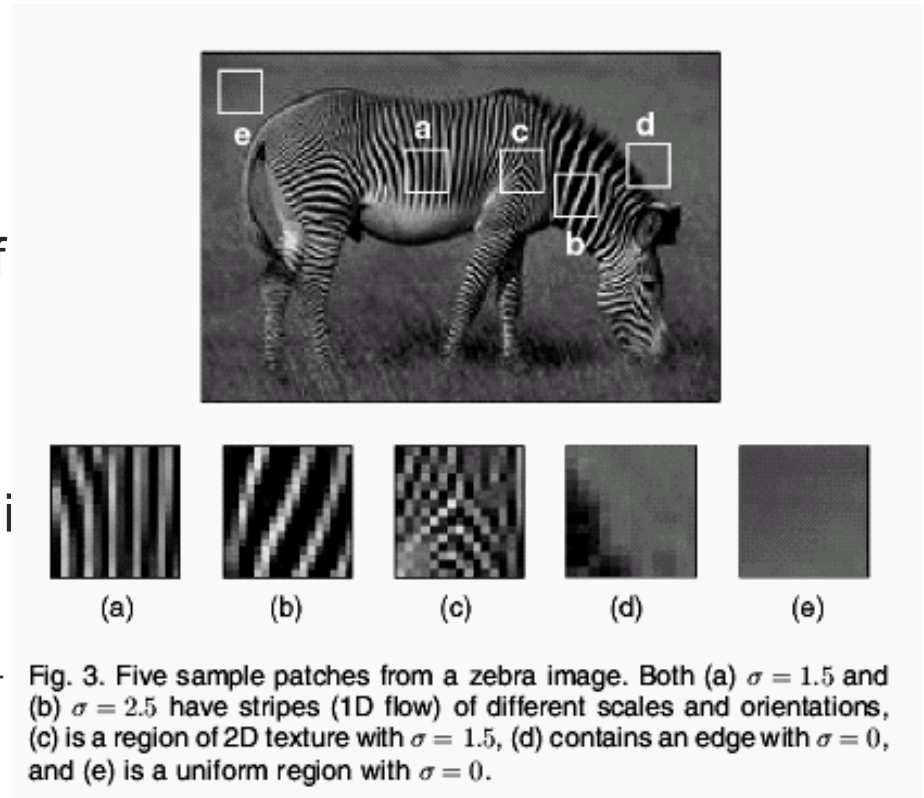
Scale Selection contd.

- Moment: $M_{\sigma}(x,y)$ = Second moment matrix.
- Important conclusions from Forstner
 - $h = \text{tr} [E(r_g)] : \lambda_1(g) + \lambda_2(g)$ “measuring the homogeneity of the segment features”
 - $v = \lambda_1 / \lambda_2$ “degree of orientation”
 - Largest eigen value is the estimate for the local gradient of the texture or edge.

Texture

Scale Selection contd.

- σ : **integration scale**
- To find $\sigma(x,y)$ to scale $M_\sigma(x,y)$.
 - Polarity: measure of the extent to which the gradient vectors in a local neighborhood point in the same direction.
 - $\rho_\sigma = |E_+ - E_-| / E_+ + E_-$ **varies with σ**



Texture

Scale Selection last... (at last!)

- Based on derivative of the polarity wrt scale.
- Algorithm:
 - Calculate polarity p_σ at every pixel for $\sigma_k = k/2$ ($k=1:7$).
 - Convolve each polarity image with Gaussian(variance $2\sigma_k$) to obtain smoothed polarity image.
 - For each pixel (x,y) select scale. (soft spatial frequency estimation?)

Block 2: Combining color, texture and position features (feature space)

- Color features: L,a,b
- Texture features: ac,pc,c
 - $ac = 1 - \lambda_2 / \lambda_1$
 - $pc = p_\sigma^*$
 - $c = 2\sqrt{(\lambda_2 + \lambda_1)^3}$
- Feature space = [L,a,b, ac,pc,c,x,y] @ each pixel.

Block 3: Grouping Pixels to regions

- Our good old friend EM 😊
- Determine the likelihood parameters of a mixture of K Gaussians in the feature space.
- What is the missing data?
 - Gaussian clusters to which the points in the feature space belong.
- What is significance of K ? .. *Later..*

Grouping Pixels to regions contd.

- Math of EM
- x = feature vector
- Θ = parameters (α 's and θ 's).
- f_i is a multivariate Gaussian.

$$f(x|\Theta) = \sum_{i=1}^K \alpha_i f_i(x|\theta_i),$$

$$f_i(x|\theta_i) = \frac{1}{(2\pi)^{d/2} \det \Sigma_i^{1/2}} e^{-\frac{1}{2}(x-\mu_i)^T \Sigma_i^{-1} (x-\mu_i)}$$

[EM]

■ Steps:

1. Initialize K mean vectors μ 's. and K covariance matrices Σ 's.

$$\alpha_i^{new} = \frac{1}{N} \sum_{j=1}^N p(i|x_j, \Theta^{old})$$

2. Add noise to each mean on EM restart.

$$\mu_i^{new} = \frac{\sum_{j=1}^N x_j p(i|x_j, \Theta^{old})}{\sum_{j=1}^N p(i|x_j, \Theta^{old})}$$

3. Update equations:

$$\Sigma_i^{new} = \frac{\sum_{j=1}^N p(i|x_j, \Theta^{old}) (x_j - \mu_i^{new})(x_j - \mu_i^{new})^T}{\sum_{j=1}^N p(i|x_j, \Theta^{old})}$$

[EM contd.]

4. Repeat 1-3 until the log-likelihood increases by less than 1% from one iteration to the next.
5. Repeat iteration 4 times (adding Gaussian noise each time) to avoid shallow local minima.

$$\log \mathcal{L}(\Theta|\mathcal{X}) = \log \prod_{k=1}^N f(x_k|\Theta)$$

[What about K?]

- Ideally: image dependent.
- MDL: minimum description length
 - The purpose of statistical modeling is to discover regularities in observed data. The success in finding such regularities can be measured by the length with which the data can be described. This is the rationale behind the **Minimum Description Length (MDL)** Principle introduced by **Jorma Rissanen** (Rissanen, 1978).

[MDL]

- Tasks:
 - Model selection
 - Parameter estimation
 - Prediction
- Idea:
 - Any set of regularities we find reduces our uncertainty of data and we can use to encode data in a shorter and less redundant way.

[MDL contd.]

- Do you want the details?
- <http://www.cs.helsinki.fi/u/ttonteri/information/lectures/Lecture4.html>
- $m_K = (K-1) + Kd + Kd(d+1)/2$
- K may not be perfect! But its selection allow us to segment the image effectively.

Post-processing and Segmentation Errors

- Why do we need post-processing?
 - Boundary-problem.
- Errors:
 - Background may be split
 - Boundary problem (discussed above)
 - Missing data- no initial mean falls near the objects feature vector. (**Danger!**)

Image Retrieval by Querying: Blobworld

- Old systems drawbacks?
- “atomic query” : particular blob query (e.g.. Like blob-1)
- <http://elib.cs.berkeley.edu/photos/blobworld>

Scoring and Retrieval

■ Notations:

- μ_i – score on each atomic query
- \mathbf{v}_i - feature vector

■ Scoring system:

1. For each blob b_j in the database image (with feature vector \mathbf{v}_j):
 - a. Find the Mahalanobis distance between \mathbf{v}_i and \mathbf{v}_j : $d_{ij} = (\mathbf{v}_i - \mathbf{v}_j)^T \Sigma (\mathbf{v}_i - \mathbf{v}_j)$.
 - b. Measure the similarity between the two blobs using $\mu_{ij} = e^{-\frac{d_{ij}}{2}}$. This score is 1 if the blobs are identical in all relevant features; it decreases as the match becomes less perfect.
2. Take $\mu_i = \max_j \mu_{ij}$.

Scoring and Retrieval contd.

- The matrix Σ is block diagonal.
 - Block corresponding to texture: I weighted by texture weights set by user.
 - Block corresponding to color: A quadratic distance weighted by color weight.

$$d_{\text{color}}^2(\mathbf{x}, \mathbf{y}) = (\mathbf{x} - \mathbf{y})^T \mathbf{A} (\mathbf{x} - \mathbf{y})$$

$\mathbf{A} = [a_{ij}]$: a symmetric matrix of weights $[0, 1]$ representing the similarity between bin i & j based on the distance between the bin centers; neighboring bins have weight 0.5. **Why Is this measure useful?**

Scoring and Retrieval contd.

- Compound query:
 - (like blob-1 and like blob-2 or blob-3)\
 - $\text{Score} = \min \{ \mu_1, \max\{\mu_2, \mu_3\} \}$
- Rank the images according to the overall score and return best matches.
- Including Background in Retrieval ?

[Results

- 10,000 Corel images

Query image: 1000127

Query labels:

Block and feature importance				
Block ID	Block importance	Color	Texture	Location change
Block 1	important	high	unimportant	not

Querying from 10,000 images (full search):

10 100004 (score = 0.94)

Mask image:

10 100004 (score = 0.94)

Mask image:

11 100002 (score = 0.94)

Mask image:

11 100002 (score = 0.94)

Mask image:

12 100003 (score = 0.94)

Mask image:

12 100003 (score = 0.94)

Mask image:

13 100005 (score = 0.94)

Mask image:

13 100005 (score = 0.94)

Mask image:

14 100006 (score = 0.94)

Mask image:

14 100006 (score = 0.94)

Mask image:

Comparisons to Global Histogram

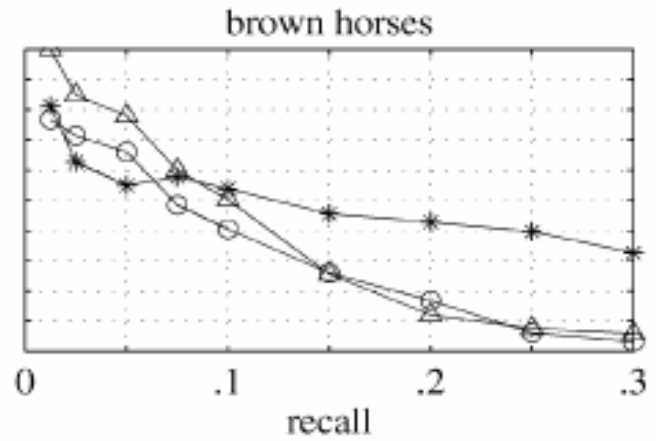
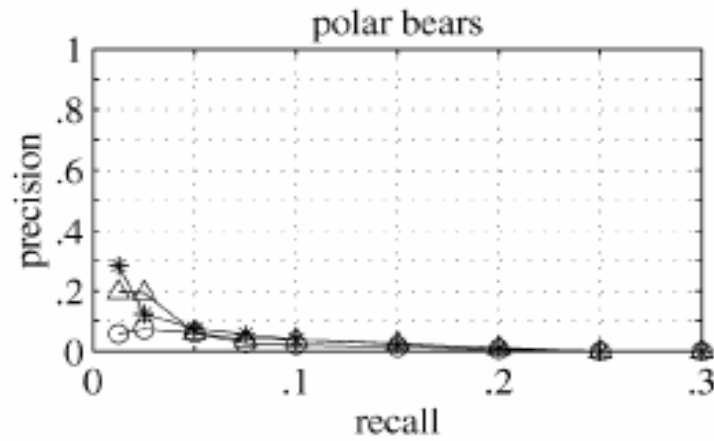
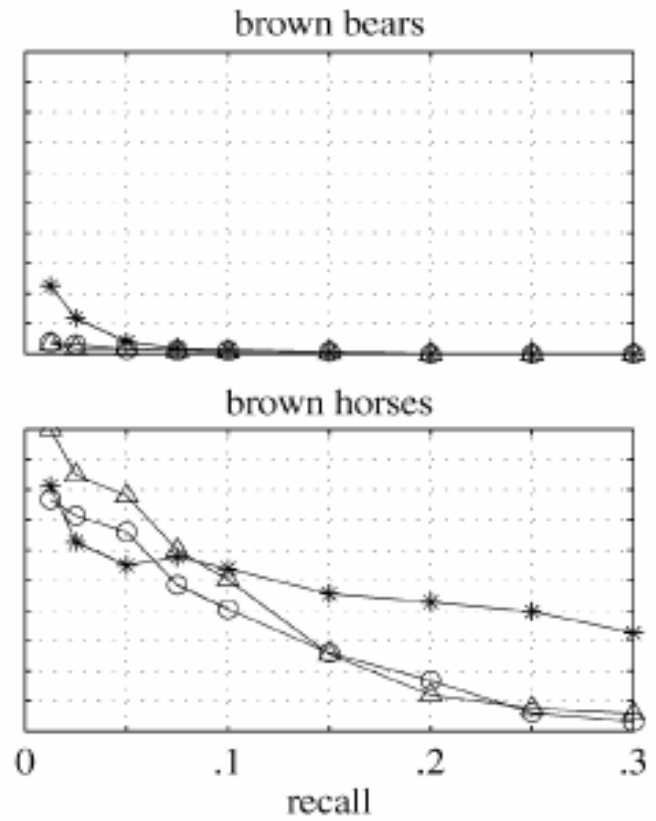
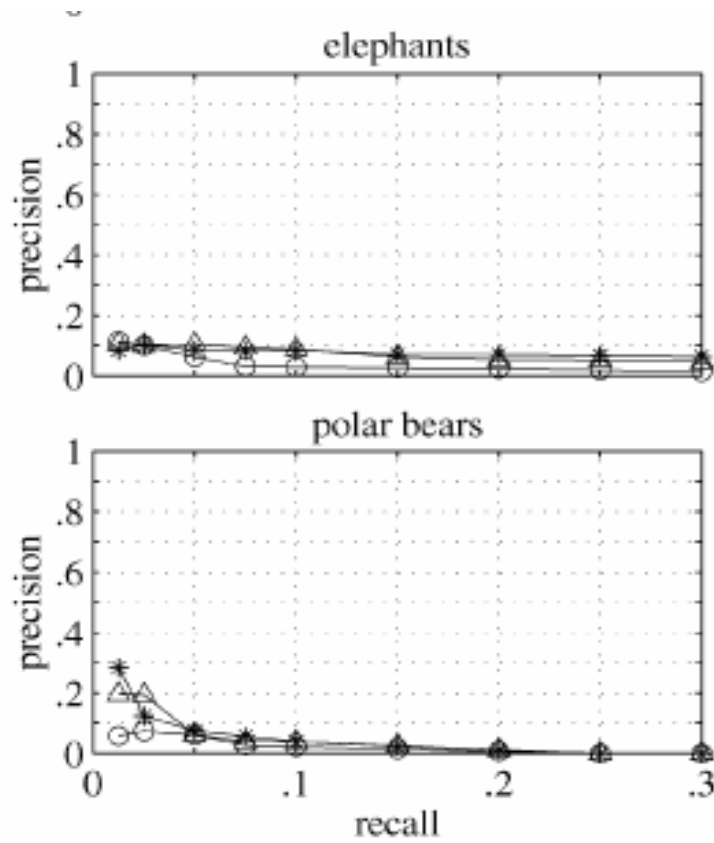
- Works great when object of search is “very distinctive”.
- Global Histograms observations:
 - Color carried most of the information
 - Ranking algo:
 - 218 L*a*b bins as in Blobworld.
 - 2 texture features into 21 bins each (equally spaced).

[Comparisons ...]

- Query: 2 blobs, 1-blob and background
- average precision = $\frac{\# \text{ relevant images}}{\# \text{ images retrieved}}$
- Recall = $\frac{\# \text{ relevant images retrieved}}{\text{total } \# \text{ of relevant images}}$
- Categorization of results:
 - Distinctive objects
 - Distinctive scenes
 - Distinctive objects and scenes
 - Other

“ Blobworld performs better when querying for distinctive objects ”

[Graphs contd.]



Inferences

- Blobworld advantages:
 - Interactive user-based query.
 - Allows query based on shape.
- Harder queries:
 - Objects and scenes are not distinctive.
 - Query with high score and lot of nearby neighbors will have a low precision.
 - **Hard queries => many “near” neighbors!**

[Discussion]

- How to avoid over-segmentation?
...using Gestalt factors!
- Has Shape feature been fully utilized?
- Is better segmentation going to make a better retrieval system?