



Université du Québec

**École de technologie supérieure**

# BD Image

# Plan de la présentation

- **Les technologies du traitement de l'image (11.2)**
- Extractions de traits (*caractéristiques*) (11.3)
- Les requêtes basées sur les traits (*caractéristiques*) (11.4)
- Processus des requêtes d'images
- Le traitement de l'image avec Oracle *interMedia* et nouveautés SQL/MM

# Problématique avec les images

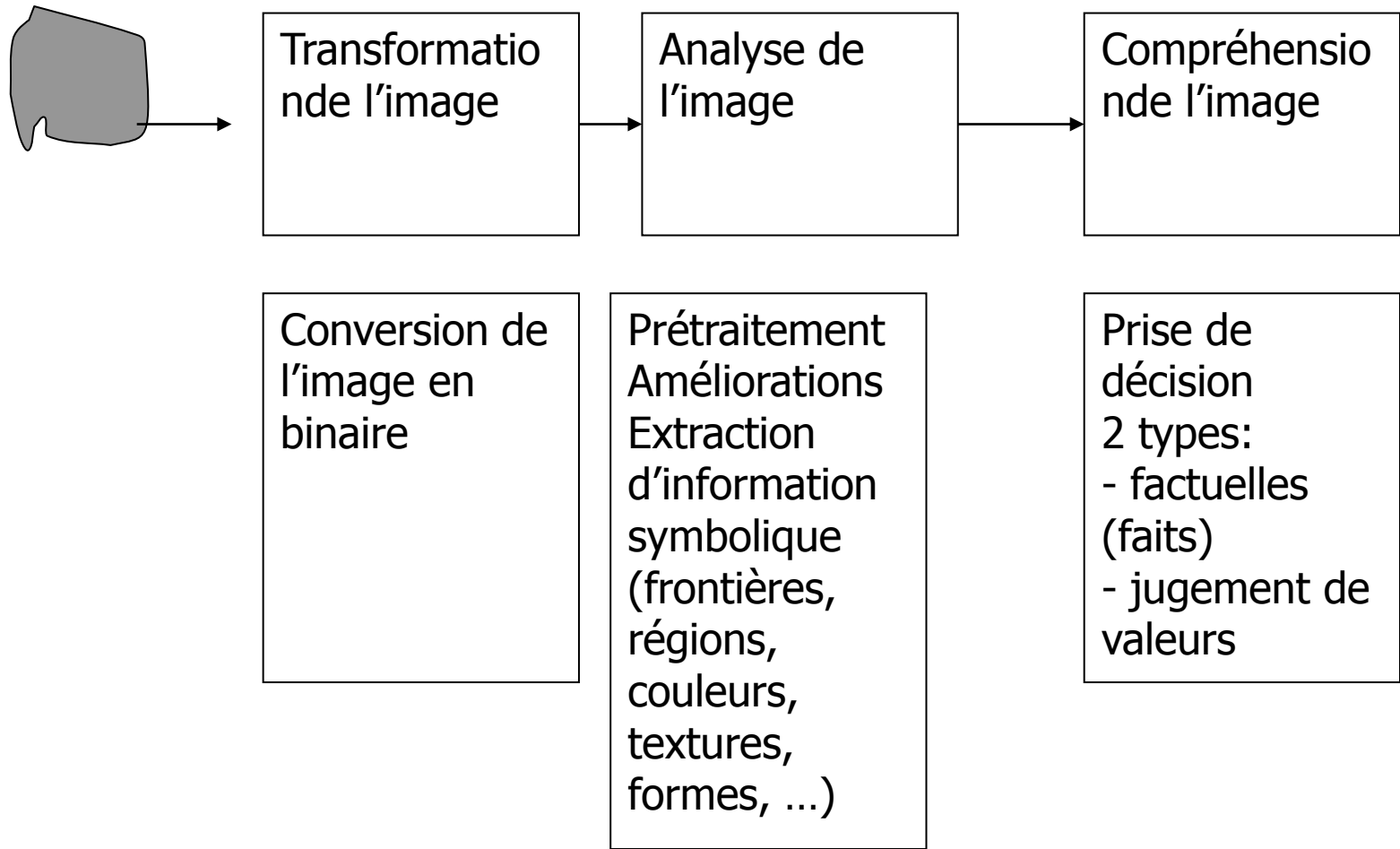
- Sémantique intrinsèque à l'image et à ses associations avec d'autres objets visuels
- Caractériser un contenu visuel est plus complexe et approximatif
- Similarité entre deux objets visuels basée sur la similitude apparente
- Encoder une description de chaque image
- Appariement par la similarité entre les descriptions

# Technologies du traitement des images

## Vision artificielle

- Discipline de l'intelligence artificielle et du traitement des images
- Recherche des méthodes pour extraire et encoder l'information visuelle
- Traitement multi-niveaux des images
  - Bas-niveau: améliorer la qualité de l'image (élimination du bruit, augmentation des contrastes)
  - Haut-niveau: identification de caractéristiques par la reconnaissance de patrons

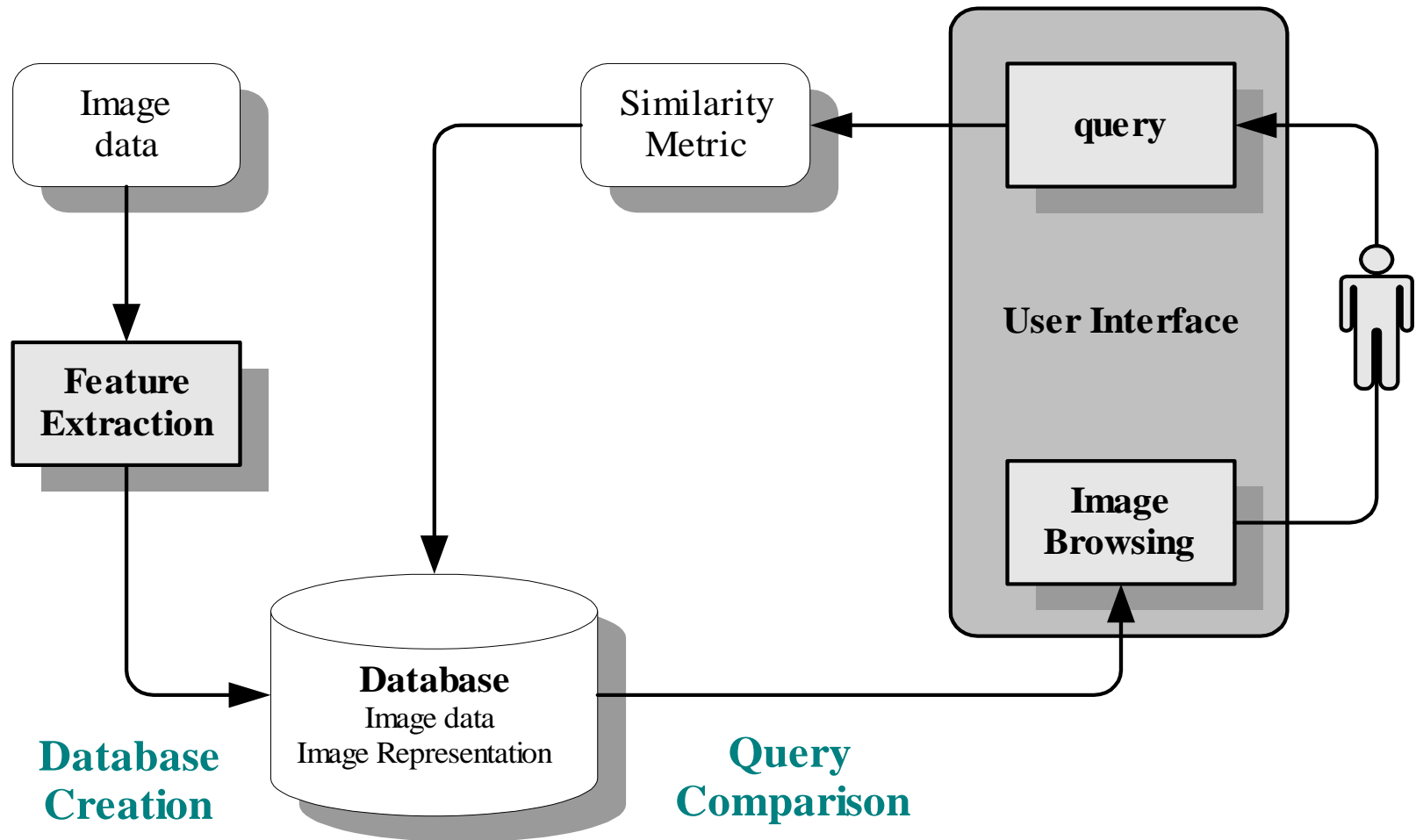
# Processus du traitement des images



# CBIR et CBR

- Content-based Retrieval (CBR) se concentre sur les caractéristiques de bas niveau
- Content-based Image Retrieval (CBIR) ajoute l'extraction de caractéristiques de haut niveau
  - Représente une image avec un ensemble de descripteurs de caractéristiques (**feature descriptors**)
  - Définit des **mesures de similitude** des descripteurs
  - La BDMM trie les images, selon la similitude avec chaque requête d'utilisateur

# L'architecture CBIR



# Bases de la manipulation d'images

- Les images se caractérisent par un vecteur de *caractéristiques (feature vectors)*
- Chaque image possède ses valeurs propres pour chaque caractéristique dans la base de donnée
- Le traitement d'image de type CBR n'utilise pas la logique booléenne d'une manière générale
- Des opérateurs numériques sont utilisés afin de calculer la similitude des images pour une requête



# MPEG-7 Visuel (Traits normalisés)

- Basic structures (5):
  - Grid layout, 2D-3D multiple view, time series, spatial 2D coordinates, temporal interpolation
- **Color Descriptors (8):**
  - **Color space, color quantization, dominant colors, scalable color, color-structure descriptor, color layout, GoF/GoP Color**
- **Texture Descriptors (3):**
  - **Edge histogram, homogeneous texture, texture browsing**
- **Shape Descriptors (4):**
  - **object- region-based shape, contour-based shape, 3D shape, 2D-3D multiple view**
- Motion Descriptors (4):
  - Camera motion, object motion trajectory, parametric object motion, motion activity
- Localization (2): Region locator, spatio temporal locator
- Others: Face recognition

# Plan de la présentation

- Les technologies du traitement de l'image
- **Extractions de caractéristiques (11.3)**
- Les requêtes basées sur les caractéristiques
- Processus des requêtes d'images
- Le traitement de l'image avec Oracle *inter*MEDIA et nouveautés SQL/MM

# Le rôle de l'extraction des caractéristiques

- Un trait caractérise une propriété spécifique d'une image (ex: sa couleur)
- Un trait est représenté par un petit ensemble fini de chiffres que l'on nomme un vecteur de caractéristiques
- Objectif de l'extraction des caractéristiques
  - Déterminer un ensemble de traits numériques
  - Minimiser le nombre de dimensions
  - Maximiser la discrimination entre les images

# CBIR de papillons

- Permettre aux utilisateurs non experts de trouver une espèce spécifique de papillon à partir d'un spécimen
- Décrire son apparence :
  - Couleur, Texture, Forme



# Objectif de l'extraction des caractéristiques

- Calculs simples sur les traits de bas niveaux: couleurs, frontières des couleurs pour reconnaître le contenu
- Utilisés pour calculer la similitude entre l'image recherchée et les images de la base de donnée d'images

# Problèmes typiques rencontrés

- Différents utilisateurs = perceptions différentes
- Ne se souviennent pas toujours clairement de l'apparence du papillon recherché
- N'ont pas toujours une connaissance approfondie pour décrire des papillons
- N'ont pas toujours la patience de consulter beaucoup de résultats pour retrouver leur spécimen

# Solutions à ces problèmes

- Processus de requêtes interactives de type QBF/QBE query process
  - Query By Features and Query By Example
- Description floue des traits d'un papillon
- Interface pour des requêtes de type wysiwyg ("Ce que vous voyez c'est ce que vous obtenez" )
- Un ensemble (une collection) représentative de papillons

# Processus de requêtes QBF/QBE(1)

- Requête QBF

- Permet de choisir quelques traits de papillons
  - Couleurs dominantes, permutations de textures, formes
- La BD doit retrouver l'ensemble des images qui y correspondent

- Requête QBE

- Soumet une image type
- La BD doit retrouver l'ensemble des images qui y correspondent



# Processus de requêtes QBF/QBE(2)

- Propriétés d'une requête QBF
  - Recherche brute
  - Utilisée en première requête pour agrandir la frontière de recherche
- Propriétés d'une requête QBE
  - Recherche raffinée
  - Utilisée typiquement aux étapes finales de recherche pour trouver des objets voisins du spécimen

# Processus de requêtes QBF/QBE(3)

- Page de présentation des résultats  
(toujours en deux parties distinctes)
  - Images résultantes
    - Papillons qui rencontrent la condition de la requête
    - Peuvent être utilisées pour formuler une nouvelle requête de type QBE
  - Traits associés
    - Caractéristiques qui décrivent la dernière requête
    - Peuvent être utilisées pour formuler une nouvelle requête de type QBF

# Description de traits (1)

- Description des traits (*features*)
  - Méta données qui décrivent l'apparence
  - Consiste en quelques descripteurs
  - Rend possible la requête de type QBF
- Descripteur de traits (*feature descriptor*)
  - Une paire ( "valeur du trait" , "valeur de correspondance" )

# Description des traits (2)



Type de Trait	Valeur du Trait	Degré de correspondance
Couleur	mixed_with_black_and_orange	52/57
	orange_yellow	12/42
	orange_red	3/38
Texture	many_spots	58/62
	fore_half_different_color	27/33
	horizontal_bands	41/60
	edge_with_different_color	10/74
Forme	wave	98/110










# CBIR classique avec traits de couleurs










- Majorité des BD de type CBR utilisent la notion de couleur, avec quelques nuances
  - Couleur dominante
  - Basé sur l'histogramme des couleurs
    - Globale: pour toute l'image; affecté par distorsions locales
    - Locale: par sous-région de l'image; plus long à calculer
  - Descripteurs de la structure de la couleur, incorpore la structure spatiale de la couleur
- 90% de CBIR est fondé sur la couleur et non sur les autres types de traits !



# Description des traits (3)

## Couleur

Figure	Feature Value
	black
	brown
	bister
	orange_red
	orange_yellow
	yellow
	green
	blue
	purple









	gray
	white
	mixed_with_black_and_white
	mixed_with_black_and_yellow
	mixed_with_black_and_orange
	mixed_with_black_and_blue
	mixed_with_black_and_red
	mixed_with_wood_and_white
	mixed_with_many_colors





# L'importance de la texture et de la forme des images

- Éléments de traits pour la texture
  - Régularité : patron ou aléatoire
  - Périodicité : répétitivité d'un patron
  - Directionnelle : patrons orientés dans un sens identifiable

# Description des traits (4)

## Texture








Figure	Feature Value
	vertical_bands
	horizontal_bands
	many_bands
	two_lines
	many_lines
	Obvious_vein
	grids
	eyes

	few_spot
	some_spots
	many_spots
	color_blocks
	grainy
	edge_with_different_color
	starlike
	fore_half_different_color



# Description des traits (5)

## Forme

Figure	Feature Value
	swallowlike
	swallowtail
	broken
	wave
	like_leaf
	like_moth
	with_little_tails

# Description des traits (6)

- Requête de type QBF
  - Requête avec un seul trait
    - Images résultantes avec degré de similitude  $> 0$
    - Trié en ordre décroissant de degré de similitude
    - Appelé la séquence ordonnée "*feature sequence*"
  - Requête avec plusieurs traits:
    - Degré de similitude calculé pour chaque trait
    - Fusion des séquences ordonnées

# Vecteur d'index des traits

- But
  - Rendre la recherche efficace
- Problèmes d'indexation en CBIR
  - Dimensionnalité élevée de l'espace des traits
  - Structure d'index doit supporter des mesures de similitude non-Euclidienne
- Solution
  - Réduction de dimension (~PCA)
    - KLT (Karhunen-Loeve Transformation)
    - DFT (Discrete Fourier Transform)
    - DCT (Discrete Cosine Transform)
    - DWT (Discrete Wavelet Transform)
  - Indexes de similitude: R\*-tree, SS-tree, SR-tree

# Extraction des traits d'une manière Semi-Automatique

- Segmentation
  - Segmentation du plan arrière ou contours
  - Segmentation des points spécifiques
- Extraction des traits
  - Couleur: histogramme de couleurs
  - Texture: Annotation manuelle
  - Forme: Annotation manuelle

# Operations sur les vecteurs de traits

- Projection
  - Crée un vecteur plus petit en demandant un sous-ensemble à l'utilisateur
- Appliquer une fonction
  - Fonction de filtre sur les couleurs pour extraire les vecteurs contenant du rouge
- Mesures de distance
  - Entre deux objets à partir de leurs vecteurs de traits

# Espace de traits (Feature Space)

- Espace de traits
  - Espace engendré par les valeurs des variables de traits
  - Dimensionnalité = nombre de variables
  - Chaque objet est représenté comme un point dans l'espace de traits

# Opérations sur espace de traits

- Localisation d'un point
- Localisation des points d'une région
- Union
- Différence
- Appartiens à un ensemble
- Addition, Décompte
- Trier

# Opérations avancées sur espace de traits

- Délimiter la frontière d'un objet... pour ensuite trouver les objets appartenant à la même région
- Sélection par contrainte spatiale
  - Trouver les objets contenus dans une région
  - Utile en GIS quand il y a seulement 2 ou 3 dimensions
- Sélection par distance
  - Trouver les objets 'proches' d'un point dans l'espace
  - Trouver les objets 'proches' d'un autre objet (*k-nearest neighbor*)
- Partitionner l'espace en sous-régions (*cluster*)
- Agrégation : décompte, moyenne, écart type, etc.



# 5 principaux types de traits

<b>Trait</b>	<b>Technique principale</b>
Couleur	Histogramme de couleur
Texture	Luminosité, intensité
Forme	Rapport hauteur/largeur, moments, frontières
Position	Coordonnées spatiales
Apparence	Courbure et orientation



# Plan de la présentation

- Les technologies du traitement de l'image
- Extractions de traits (*caractéristiques*)
- **Les requêtes basées sur les traits (*caractéristiques*)**
- Processus des requêtes d'images
- Le traitement de l'image avec Oracle *interMedia* et nouveautés SQL/MM

# Requêtes sur la couleur (11.4.1)

- Le contenu des couleurs est l'élément majeur du repérage d'image
- Exprimé dans le modèle HSI
  - Hue: teinte spectrale dominante  $[0^{\circ}:360^{\circ}]$
  - Saturation: profondeur de la couleur  $[0:100\%]$
  - Intensity: brillance de la couleur  $[0:100\%]$  vue comme des niveaux de gris

Voir l'article [Histogram-Based Color Image Retrieval](#) de Sangoh Jeong

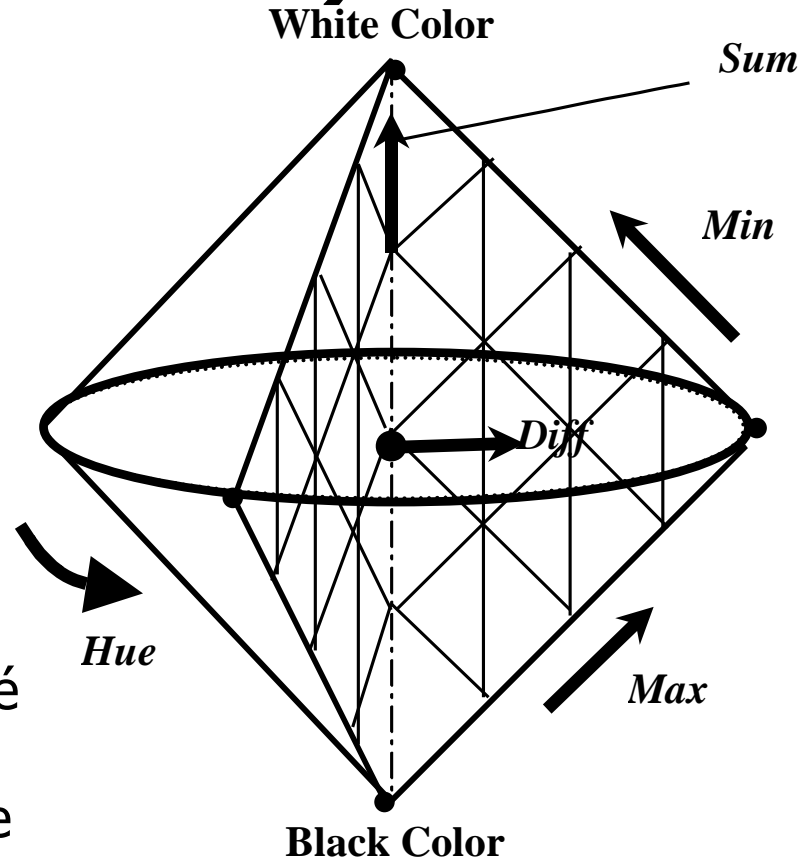
[\[http://scien.stanford.edu/class/psych221/projects/02/sojeong\]](http://scien.stanford.edu/class/psych221/projects/02/sojeong)

# Suite logique du cours 7

- On a vu que l'humain peut voir une fraction du spectre électromagnétique
- Les chercheurs ont réussi à créer des normes industrielles pour encoder les couleurs afin de mieux les sauvegarder et les extraire des médias logiciels (Espaces de couleur – modèles de couleurs)
- Modèle de couleur RGB
- Modèle de couleur HSB/V
- Modèle de couleur YCbCr

# Modèle de couleur HMMD (norme MPEG-7)

- HMMD (Hue-Max-Min-Diff)  
plus près d'une perception  
uniforme
  - **Hue** = même que dans le  
modèle HSI
  - **Max** =  $\max(R,G,B)$  indique la  
quantité de noir
  - **Min** =  $\min(R,G,B)$  indique la  
quantité de blanc
  - **Diff** =  $(\text{Max} - \text{Min})$  indique la  
quantité de gris et la proximité  
à une couleur pure
  - **Sum** =  $(\text{Max} + \text{Min})/2$ , simule  
l'intensité dans le modèle HSI



# Histogramme de couleurs

- Enregistre la proportion des pixels de chaque catégorie de couleur (*bin*) pour chaque image
- Même encodage de couleurs pour toutes les images
- Requête par une spécification des proportions de couleur
- Requête par l'exemple: construction d'un histogramme de couleurs pour l'image-requête
- Appariement avec les images de la collection selon une fonction de similarité sur les histogrammes
- Technique la plus utilisée: intersection d'histogrammes

# Similarité entre histogrammes

- Minkowski Similarity

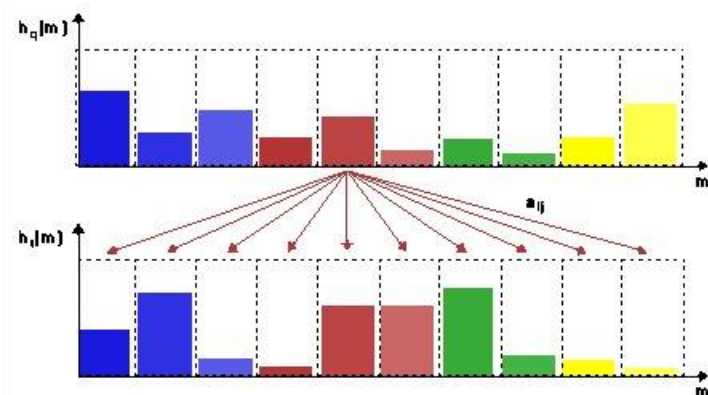
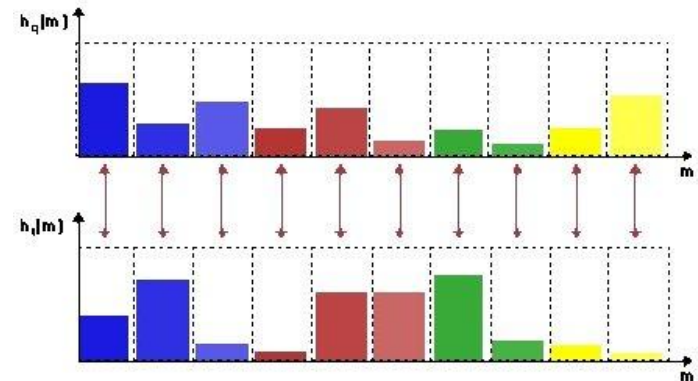
$$d(h_q, h_t) = \left( \sum_{k=0}^{M-1} |h_q[k] - h_t[k]|^r \right)^{1/r}, \quad r \geq 1$$

- Quadratic Similarity

$$d(h_q, h_t) = (h_q - h_t)^T Q Q^T (h_q - h_t)$$

- Intersection Similarity (Swain et Ballard 1991)

$$d_{q,t} = \frac{\sum_{k=0}^{M-1} \min(h_q[k], h_t[k])}{\sum_{k=0}^{M-1} h_t[k]}$$



# Exemple calcul de similarité

- Avec les 3 images du cours précédent

- $H_1 = (8, 8, 8, 8, 8, 8, 8, 8)$

- $H_2 = (7, 7, 7, 7, 9, 9, 9, 9)$

- $H_3 = (2, 2, 10, 10, 10, 10, 10, 10)$

- Minkowski Similarity 
$$D_{r=1}(H_I, H_J) = \sum_{k=1}^n |I_k - J_k|$$

- $D(H_1, H_2) = 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 = 8$

- $D(H_1, H_3) = 6 + 6 + 2 + 2 + 2 + 2 + 2 + 2 = 24$

- $D(H_2, H_3) = 5 + 5 + 3 + 3 + 1 + 1 + 1 + 1 = 23$

- Avec  $r = 2$  :  $D_{1,2} = 2,8$      $D_{1,3} = 9,8$      $D_{2,3} = 8,5$



# Exemple calcul de similarité

- Quadratic Similarity  $d(h_q, h_t) = (h_q - h_t)^T (h_q - h_t)$ 
  - $D(H_1, H_2) = 4*(8*7) + 4*(8*9) = 512$
  - $D(H_1, H_3) = 2*(8*2) + 6*(8*10) = 512$
  - $D(H_2, H_3) = 2*(7*2) + 2*(7*10) + 4*(9*10) = 528$
- Intersection Similarity  $d_{q,t} = \sum_{k=0}^{M-1} \min(h_q[k], h_t[k]) / \sum_{k=0}^{M-1} h_t[k]$ 
  - $D(H_1, H_2) = [4*(7) + 4*(8)] / 64 = 60 / 64$
  - $D(H_1, H_3) = [2*(2) + 6*(8)] / 64 = 52 / 64$
  - $D(H_2, H_3) = [2*(2) + 2*(7) + 4*(9)] / 64 = 54 / 64$

# Exemple calcul de similarité

	Minkowski	Quadratic	Intersection
$D(H_1, H_2)$	8	512	<b>60</b>
$D(H_1, H_3)$	<b>24</b>	512	52
$D(H_2, H_3)$	23	<b>528</b>	54

- Minkowski Similarity
  - distance sur couleurs correspondantes (max:  $D(H_1, H_3)$ )
- Quadratic Similarity
  - distance sur couleurs corrélées (max:  $D(H_2, H_3)$ )
- Intersection Similarity
  - distance sur base commune de couleurs (max:  $D(H_1, H_2)$ )

# Requêtes sur la forme (11.4.3)

- Objectif principal : quantifier la forme des objets principaux
- Extraction des traits de l'image en comparant avec une collection de traits types enregistrée dans la BDMM
- Traits types sont indépendant de la taille et de l'orientation (*invariance spatiale*)

# Requêtes sur la forme

- Deux approches (Globale et Locale)
- Caractéristiques globales
  - Rapport hauteur/largeur (*aspect ratio*)
  - Circularité
  - Moments invariants
- Caractéristiques locales
  - Utilise la segmentation
  - Ensembles de segments de frontières consécutifs

# Techniques de segmentation

- Problèmes du global

- L'arrière-plan peut causer une extraction imprécise des traits globaux
- Dans certaines applications, on veut retrouver des objets précis dans l'image

- Solution

- Segmentation par région
- Segmentation par objet

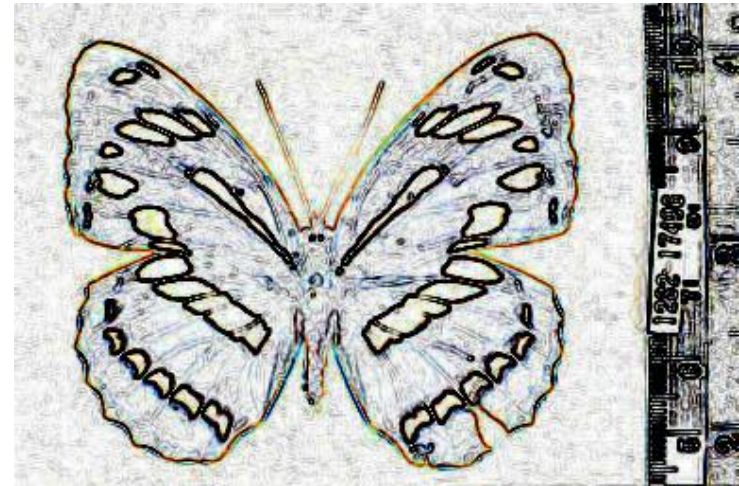
# Exemple 1 de segmentation régionale

- Détection des contours
  - Trouver les frontières de chaque région



# Exemple 2 de segmentation régionale

- Détection des contours
  - Trouver les frontières de chaque région



# Détection des contours par le filtre de Sobel

- Filtre de détection Sobel : accentuation des contrastes par la méthode du gradient maximum

Filtre vertical Gy

$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix}$$

$$g(x, y) = \begin{matrix} -1 f(x-1, y-1) + -2 f(x, y-1) + -1 f(x+1, y-1) + \\ 0 f(x-1, y) + 0 f(x, y) + 0 f(x+1, y) + \\ 1 f(x-1, y+1) + 2 f(x, y+1) + 1 f(x+1, y+1) \end{matrix}$$

Filtre horizontal Gx

$$\begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix}$$

$$g(x, y) = \begin{matrix} -1 f(x-1, y-1) + 0 f(x, y-1) + 1 f(x+1, y-1) + \\ -2 f(x-1, y) + 0 f(x, y) + 2 f(x+1, y) + \\ -1 f(x-1, y+1) + 0 f(x, y+1) + 1 f(x+1, y+1) \end{matrix}$$

$f(x, y)$  : niveau de gris du pixel  $(x, y)$  dans l'image initiale  
Gx et Gy : images après détection des contours



# Détection des contours par le filtre de Sobel (suite)

- Combinaison de la magnitude des deux gradients

$$G(x, y) = \sqrt{Gx^2 + Gy^2}$$

- Utilise un seuil  $h$  pour déclarer le pixel de type contour
  - $G(x,y) > h \Rightarrow$  pixel  $(x,y)$  est un contour



image source



gradient x

+



gradient y

=



image contour

[source: [Wikipedia](#)]



# Exemple de segmentation d'objet (1)

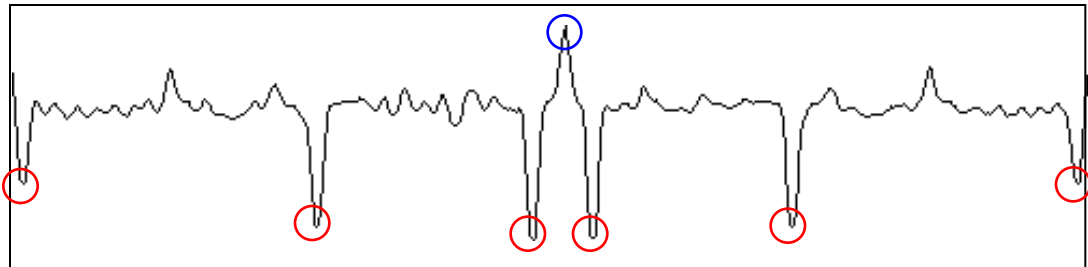
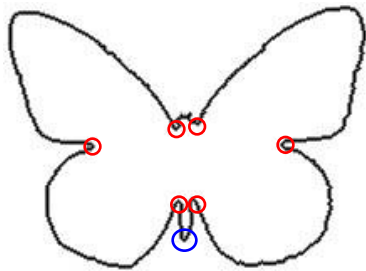
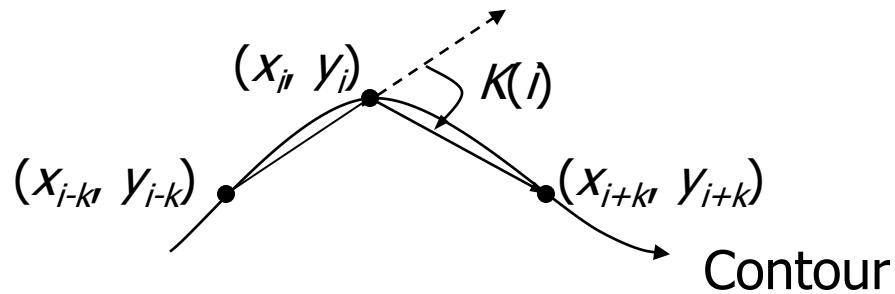
- Segmentation de l'objet papillon
  - Trouver des points particuliers (inflexions)
    - Ailes antérieures
    - Ailes postérieures
    - Corps



# Exemple de segmentation d'objet (2)

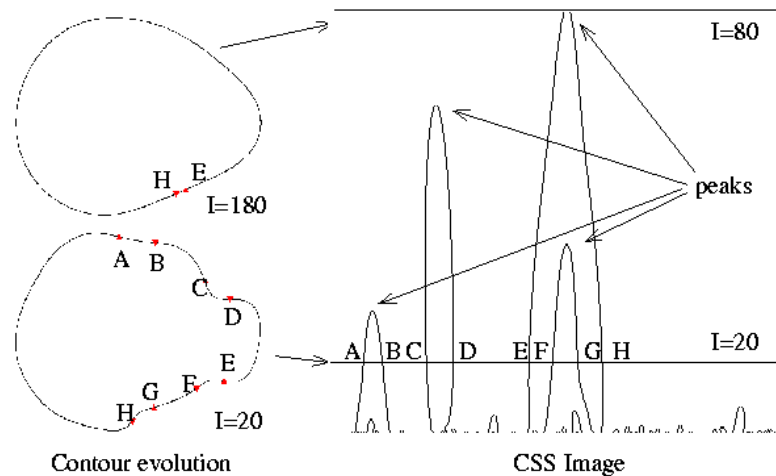
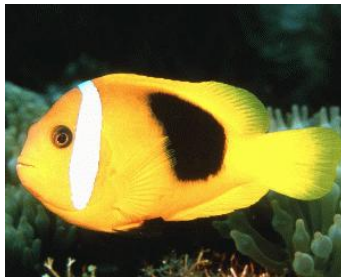
- Représentation de Courbures  $K(i)$

- concavités
- convexités



# Exemple de contour d'objet (3)

- Représentation d'un Contour fermé [source: Irek Defee]



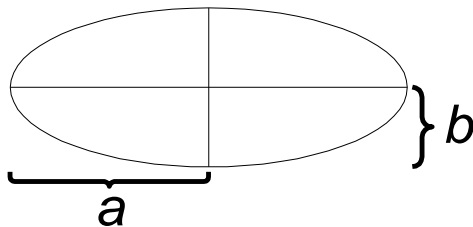
- Relève  $n$  points sur le contour en naviguant dans le sens horaire
- Coordonnées  $(x,y)$  organisées en vecteurs  $X$  et  $Y$
- Adoucit le contour par l'application répétée d'un filtre passe-bas
- Mesure la circularité et l'excentricité du contour résultant

# Circularité et excentricité

- Circularité : mesure du rapport entre le périmètre et la surface

$$circularité\ é = \frac{\text{périmètre}^2}{surface}$$

- Excentricité : mesure du rapport entre le plus grand axe et le plus petit axe



$$excentricité = \frac{a - b}{a}$$

- Formules d'approximation pour forme générale

# Requêtes sur la texture (11.4.2)

- L'appariement par similtude sur la texture permet souvent de distinguer des régions d'une image ayant des couleurs similaires (ciel et océan, feuilles et gazon)
- La méthode utilise les valeurs d'intensité des pixels qui varient selon la réflexion ou la réfraction de la lumière sur des surfaces
- L'arrangement des intensités définit la texture visuelle de l'image

Exemple de Qazim Iqbal (Université du Texas)

[source: [http://amazon.ece.utexas.edu/~qasim/samples/sample\\_textures.htm](http://amazon.ece.utexas.edu/~qasim/samples/sample_textures.htm)]

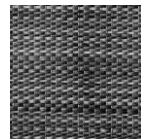
# Régularité, direction et échelle

MPEG-7 définit un descripteur de texture à 3 dimensions

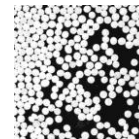
- Régularité : la texture est régulière si son patron est
  - périodique
  - orienté
  - de granularité uniforme
- Direction : orientation dominante du patron
- Échelle : taille des granules du patron dans l'orientation dominante



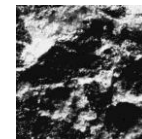
**11**



**10**



**01**



**00**

[source: Irek Defee]



Université du Québec

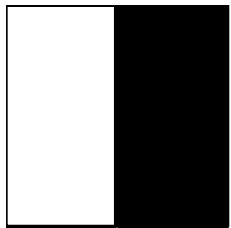
École de technologie supérieure

Département de génie logiciel et des TI

# Histogramme de patrons

MPEG-7 définit 5 catégories de patrons

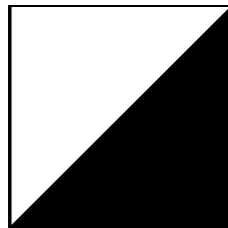
- 4 directionnelles et une non directionnelle
- Utilise la transformée de Fourier (analyse spectrale)
- Un histogramme de patrons décrit les patrons d'une image



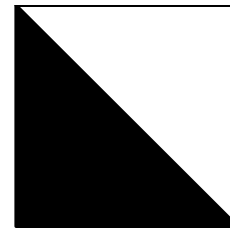
a) vertical



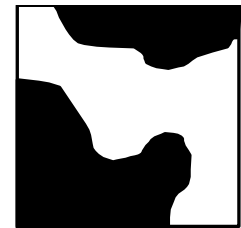
b) horizontal



c) 45°



d) 135°



e) non directionnel

[source: Irek Defee : <http://www.cs.tut.fi/~defee/mmsp/mmsp.8.ppt>]



Université du Québec

École de technologie supérieure

Département de génie logiciel et des TI



# Requêtes sur la texture

- Formulée par la sélection d'une texture type enregistré dans la BD
- Formulée par un exemple tirée d'une image
- Appariement par similarité des textures
- Possibilité d'utiliser un thesaurus de textures pour identifier des classes de textures dans une collection d'images

# Mesures des caractéristiques

Trait	Mesures	Utilisation
Couleur	Histogramme	Indexation
Texture	Intensité de pixel Régularité, Direction, Échelle	Indexation Thesaurus
Forme	<u>Globale</u> : Rapport hauteur/largeur, Circularité, Moments invariants <u>Locale</u> : Frontières de segments	Indexation Reconnaissance d'objets (local)
Apparence	<u>Globale</u> : Courbure, orientation <u>Locale</u> : Courbure, orientation	Classification d'images
Position	Relations spatiales	Reconnaissance d'objets

# Avantages / Désavantages

Trait	Avantages	Désavantages
Couleur	Applicable aux images 2D-3D	
Texture	Capable de distinguer des régions de couleur similaire	Dimensionnalité des vecteurs de traits (~4000)
Forme	Utile pour segmentation Classifier sous différentes formes	Représentation difficile Dépendant du point de vue
Apparence	Utile pour évaluer la similarité Génération de mesures invariantes Description à différents niveaux de détail	
Position	Applicable aux images 2D-3D	Doit définir des relations spatiales Indexage spatiale en combinaison avec couleur et texture

# Analyse d'images & reconnaissance (11.5)

- Deux approches
  - Analyse globale de l'image
    - Histogramme de couleurs
    - Texture
    - Moments invariants aux rotations et aux translations
  - Segmentation en régions ou objets
    - Géométrie (formes des objets)
    - Topologie (relations spatiales entre les objets)
    - Taille
    - Forme

# Plan de la présentation

- Les technologies du traitement de l'image
- Extractions de traits (*caractéristiques*)
- Les requêtes basées sur les traits (*caractéristiques*)
- **Processus des requêtes d'images**
- Le traitement de l'image avec Oracle *interMedia* et nouveautés SQL/MM

# Problèmes du processus d'appariement d'images (11.6.2)

- a) Difficulté à exprimer la requête, à formuler des indices visuels
- b) Difficulté à exprimer la qualité d'un appariement (ex. mesure de similarité affectée par l'arrière-plan)
- c) Difficulté à ordonner les résultats d'appariement; difficulté à pondérer l'importance relative des traits



# Processus d'appariement

- Deux approches

- Utiliser une image-requête (QBE)
  - Image fournie par l'utilisateur
  - Images sélectionnées par l'utilisateur à partir d'une collection d'images types présentées
- Utiliser des traits (QBIC)
  - Sélection de traits et de valeurs à partir de listes de traits/valeurs
  - Pondération de l'importance relative de chaque trait

# Stratégies de repérage

- *Ad hoc (One-stage)*
  - Repérage unique à partir d'une requête
  - Précision faible en repérage d'images
- *Reformulation unique (Two-stage)*
  - Utilise les résultats d'un premier repérage et les jugements de pertinence de l'utilisateur pour améliorer la requête
  - Utilisation de techniques statistiques (PCA, LSI, CA) pour recalibrer la pondération des traits
- *Reformulation multiple (Multi-stage)*
  - Même que précédent, mais à répétition





# Plan de la présentation

- Les technologies du traitement de l'image
- Extractions de traits (*caractéristiques*)
- Les requêtes basées sur les traits (*caractéristiques*)
- Processus des requêtes d'images
- **Le traitement de l'image avec Oracle *inter*Media et nouveautés SQL/MM**

# Focus sur ORDImage

- ORDImage intègre l'entreposage, le repérage et la gestion d'images numérisées
- De même ORDVideo et ORDAudio pour les fichiers vidéo et audio

# Types **ORDImage** et **ORDSource**

```
CREATE TYPE ORDImage AS OBJECT  
(  
  source          ORDSource,  
  height          INTEGER,  
  width           INTEGER,  
  contentLength   INTEGER,  
  fileFormat      VARCHAR2(4000),  
  contentFormat   VARCHAR2(4000),  
  compressionFormat VARCHAR2(4000),  
  mimeType        VARCHAR2(4000)  
);
```

```
CREATE TYPE ORDSource AS OBJECT  
(  
  localData       BLOB,  
  srcType         VARCHAR2(4000),  
  srcLocation     VARCHAR2(4000),  
  srcName         VARCHAR2(4000),  
  updateTime      DATE,  
  Local           NUMBER  
);
```

# Les attributes d'ORDImage

- source: la source de l'image; contenu et autres attributs
- height: hauteur de l'image en pixels
- width: largeur de l'image en pixels
- contentLength: taille de l'image en octets
- fileFormat: le format d'encodage (TIFF, BMP, JPEG, etc.)
- contentFormat: le type d'image (monochrome , etc.)
- compressionFormat: l'algorithme de compression utilisé
- mimeType: le type MIME

# Quelques méthodes d'ORDImage (ORDSource)

- `init(srcType,srcLocation,srcName)` : constructeur
- `processCopy(command, ORDImage)` : copie avec traitement particulier précisé par un opérateur  
(voir *interMedia* reference annexe D)
- `set/getUpdateTime()` : date de dernière modification
- `set/getMimeType()` : type MIME
- `getCompressionFormat()` : Voir *interMedia* Reference B-9 à 12
- `readFromSource()` : lecture d'image par tranche
- `writeToSource()` : écriture d'image par tranche

# ORDImage – Create Table

## - Création d'une table d'image

```
CREATE TABLE stockphotos  
(  
    photo_id INTEGER,  
    photographer VARCHAR2(64),  
    annotation VARCHAR2(255),  
    photo ORDSYS.ORDImage  
);
```

# 3 approches pour charger des images avec *interMedia*

- PL/SQL
  - avec DBMS\_LOB.LOADFROMFILE
  - pour charger une image à la fois
- SQLLDR
  - avec SQL\*Loader
  - pour charger plusieurs images
- Java (*interMedia* Java Classes)
  - loadDataFromFile()
  - loadDataFromInputStream()

# Exemple SQLLDR

```
LOAD DATA
INFILE *
INTO TABLE SOUNDS
APPEND
FIELDS TERMINATED BY ','
  (Item_ID integer external,
   sound column object
   ( source column object
     ( localdata_fname FILLER CHAR(128),
       localdata LOBFILE (sound.source.localdata_fname)
         terminated by EOF
     )
   )
  )
)
)
BEGINDATA
55,the_grid.au,
33,engine.wav,
44,spacemusic.au
```



# ORDImage formats d'images -1-

- Plusieurs formats de fichiers sont supportés
- Tableau non exhaustif

Format	Pixel Format	Support
BMP	Monochrome	Read/Write
	4-bit LUT	Read
	8-bit LUT	Read/Write
	16-bit RGB	Read
	24-bit RGB	Read/Write
	32-bit RGB	Read
	<b>Compression Format</b>	<b>Support</b>
Choose one of these compression formats:	Uncompressed	Read/Write
	BMPRLE (for 8-bit LUT)	Read/Write
	<b>Data Description</b>	<b>Support</b>
Choose one or more of these content formats:	Inverse DIB	Read
	OS/2 format	Read

# ORDImage formats d'images -2-

Format	Pixel Format	Support
GIF	Monochrome	Read
	8-bit LUT	Read/Write
File Format: 'GIF'		
File Ext: .gif		
Mime: image/gif		
NOTE: interMedia image has limited support for animated GIF images; there is setProperties( ) support; however, processing using the process( ) and processCopy( ) (or Analyze) methods is not supported.		
	<b>Compression Format</b>	<b>Support</b>
	GIFLZW (LZW)	Read/Write
	<b>Data Description</b>	<b>Support</b>
	NA	NA

<https://cwisdb.cc.kuleuven.ac.be/ora10doc/mix.101/b12039/idx-i>

# Information visuelle: méthodes et opérations

## ( 1 ) Signature et similarité

- TYPE ORDImageSignature
- ORDSYS.ORDImageSignature.init();  
pour générer les traits de couleur, texture et forme
- IMGSimilar ( )  
Pour comparer la signature de deux images et  
déterminer leur similarité selon un ensemble de  
critères spécifiés par l'utilisateur

# Information visuelle: méthodes et opérations

## ( 2 ) Extraction des propriétés d'une image

- `checkProperties ( )`

Retourne 'vrai' si les propriétés de l'image correspondent aux attributs stockées dans la BD

*Pour récupérer des attributs spécifiques, ex. hauteur, largeur en pixels*

- *`getHeight ( )`*
- *`getWidth ( )`*



# Information visuelle: méthodes et opérations

## ( 3 ) Vérification des propriétés d'une image

- `checkProperties ( )`

Pour vérifier si les propriétés enregistrées correspondent à celles de l'image

## ( 4 ) Modification d'une image

- `Process ( ) / processCopy ( )`

Utilisé pour manipuler les propriétés de base

- conversion de format
- compression
- changement d'échelle
- troncature

# Information visuelle: méthodes et opérations

## ( 5 ) Déplacement d'images

- copy ( ) / processCopy ( )
- export ( ) / import ( ) / importFrom ( )

Pour déplacer/copier des images d'une source à une autre

## ( 6 ) Modification manuelle des caractéristiques

- setProperties ( )

Pour modifier explicitement les caractéristiques d'une image

# Concepts de repérage par contenu (CBR)

- Caractéristiques utilisées
- Repérage par image-requête
  - Calcul de similarité
- Index pour améliorer la performance

# Vecteur de traits et segmentation

- 1 signature par image
  - 1 vecteur global de traits
  - 1 vecteur de trait par région (segmentation)
  - 1 vecteur de trait 'background' pour les pixels n'appartenant à aucune région
- Vecteur de traits
  - Couleur : couleur dominante
  - Forme (*Structure*) : segmentation basée sur l'uniformité de couleur
  - Texture (granularité ?)
  - Localisation spatiale : des traits de couleur, de forme et de texture



# Génération des traits - 1-

## *Image initiale*



(image de Oracle corp)



Université du Québec

École de technologie supérieure

Département de génie logiciel et des TI

# Génération des traits - 2-

## *Image segmentée*



(image de Oracle corp)



Université du Québec

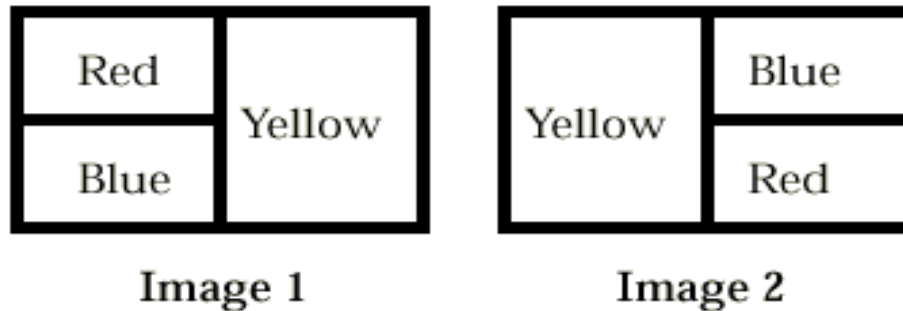
École de technologie supérieure

Département de génie logiciel et des TI

# Comparaison des couleurs -1

## ( 1 ) Couleur globale et locale

*Figure 2-1 Image Comparison: Global Color and Local Color*



- Couleur globale : similarité parfaite (score=100)
  - Proportion globale de chaque couleur: identique
- Couleur locale : aucune similarité (score=0)
  - Proportion locale de chaque couleur : aucun chevauchement

**(image de Oracle corp)**



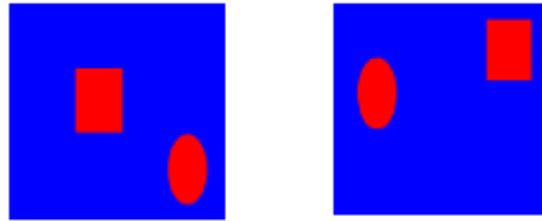
Université du Québec

École de technologie supérieure

Département de génie logiciel et des TI

# Comparaison des couleurs -2

*Figure 2-2 Images Very Similar in Global Color*



*Figure 2-3 Images Very Similar in Local Color*



- Figure 2-2: très similaires en couleur globale (score=0.0)
- Figure 2-3: très similaires en couleur locale (score=0.02461)

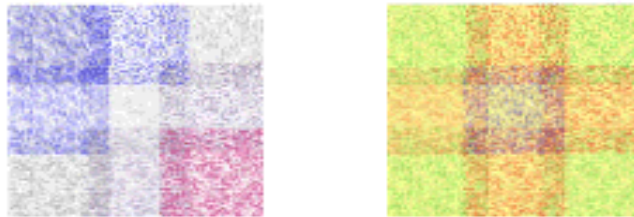
(image de Oracle corp)

# Comparaison de textures

## ( 2 ) Texture

Utile pour comparaison globale d'images  
(catalogues de grains de bois, marbre, sable, pierres)

*Figure 2-4 Fabric Images with Similar Texture*



- Figure 2-4: deux exemples de matériaux similaires (score=4.1)

# Comparaison par position

## ( 3 ) Structure (forme)

- L'uniformité de couleur détermine la taille et la localisation spatiale des régions
- Utile pour comparer des objets telles que les lignes d'horizon d'images de paysage
- Utile pour requête sur des formes simples

*Figure 2-5 Images with Very Similar Structure*



Figure 2-5: régions similaires

**(image de Oracle corp)**



Université du Québec

École de technologie supérieure

Département de génie logiciel et des TI

# Le poids de chaque trait

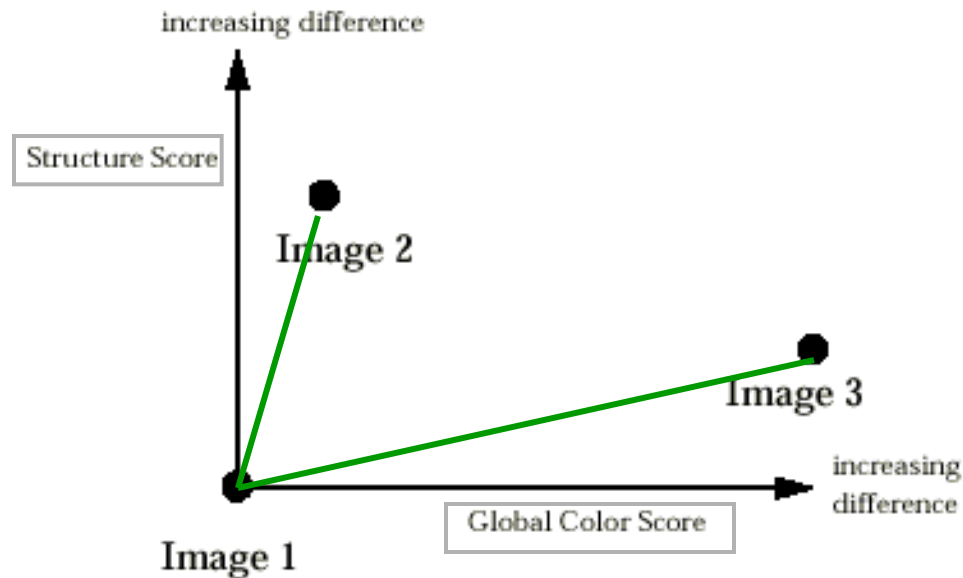
## ( 1 ) Poids accordé à un trait

- Importance accordée à un trait dans le calcul de similarité entre deux images
- Valeur du poids
  - Dans l'intervalle  $[0, 1]$
  - Somme des poids = 1
  - 0 = aucune importance, 1 = toute l'importance
  - Ex: couleur=0.6, texture=0.3, forme=0.1
- Valeur de similarité : SCORE ramené dans l'intervalle  $[0, 100\%]$

# Calcul de distance

## ( 2 ) Similarité

Figure 2-6 Score and Distance Relationship



(image de Oracle corp)



# Exemple

## ( 3 ) Calcul de similarité

Somme pondérée des poids et distances des attributs visuels

*Table 2-1 Distances for Visual Attributes Between Image1 and Image2*

Visual Attribute	Distance
Global color	15
Local color	90
Texture	5
Structure	50

[ poids ]

- Global color = 0.1
- Local color = 0.6
- Texture = 0.2
- Structure = 0.1

$$\text{SCORE} = (0.1 * 15) + (0.6 * 90) + (0.2 * 5) + (0.1 * 50) = 61.5$$

(image de Oracle corp)



# Choisir une frontière acceptable

- Appariement relatif à une valeur limite  $h$  (Threshold)
  - Appariement positif si distance calculée  $\leq h$
  - Dans l'exemple précédent, les images sont considérées similaires si  $h = 70$ , mais différentes si  $h = 60$
- Déterminer  $h$  par essais et erreurs
  - Éliminer les images non désirées
  - Accorder l'attention aux images recherchées
  - $h$  est dépendant du type d'application et de la collection d'images

# Exemple

## DECLARE

```
t_img ORDSYS.ORDImage;  
i INTEGER;  
image_sig ORDSYS.ORDImageSignature;  
query_signature ORDSYS.ORDImageSignature;  
BEGIN  
SELECT photo_id, photo, photo_sig INTO i, t_img, image_sig  
FROM stockphotos WHERE ORDSYS.IMGSimilar(photo_sig,  
query_signature, 'color="0.2" texture="0.1" shape="0.5"  
location="0.2"', 25) = 1;  
END;
```

Traits générés à partir de l'image-requête  
par un appel de generateSignature();



# Index pour la performance

```
CREATE INDEX imgindex ON stockphotos(photo_sig)  
INDEXTYPE IS ORDSYS.ORDImageIndex  
PARAMETERS(' ORDIMAGE_FILTER_TABLESPACE =  
ordimage_idx_tbs_1, ORDIMAGE_INDEX_TABLESPACE =  
ordimage_idx_tbs_2');
```

- Les tablespace ordimage\_idx\_tbs\_1 et ordimage\_idx\_tbs\_2 doivent préalablement être créés
- FILTER contient les données de l'index
- INDEX est un index interne sur FILTER
- IMGSimilar() utilisera cet index
- L'index est automatiquement mis à jour lors d'ajout/modification/suppression d'image dans la table

# *interMedia*: Exemples

- Exemple : base de données Livres
- Création du répertoire d'images

CREATE DIRECTORY IMGDIR AS 'C:\';

(À L'ÉTS) – » **CREATE OR REPLACE DIRECTORY IMGDIR AS  
'/home/bac/ag000000/IMGDIRGTI440';**

**'/home/bac/ag000000/IMGDIRGTI440'** représente le chemin du répertoire qui contient les fichiers images sur le serveur oracle ('la machine physique')

**IMGDIR** est l'alias assignée par Oracle à ce répertoire

# Création de la table d'images

```
CREATE TABLE livres (  
  numero      INTEGER PRIMARY KEY,  
  isbn        INTEGER,  
  titre       VARCHAR2(30),  
  auteur      VARCHAR2(30),  
  prix        INTEGER,  
  resume      VARCHAR2(200),  
  photo       ORDSYS.ORDImage,  
  photo_sig   ORDSYS.ORDImageSignature  
);
```

# Insertion d'images (1/2)

DECLARE

img ORDSYS.ORDImage;

sig ORDSYS.ORDImageSignature;

ctx RAW(4000):= NULL; /\* contexte utilisé par certains plug-in \*/

BEGIN

INSERT INTO livres (numero, isbn, titre, auteur, prix, resume, photo, photo\_sig)  
VALUES(1, 3544651, 'Java, la référence', 'David Flanagan', 60, 'Classes java',  
ORDSYS.ORDImage.init('file','IMGDIR', 'Sample.jpg'),  
ORDSYS.ORDImageSignature.init() );

# Insertion d'images (2/2)

- Chargement des images et génération des signatures

```
SELECT L.photo, L.photo_sig INTO img, sig  
FROM livres L WHERE L.numero = 1 FOR UPDATE;
```

```
img.import(ctx);  
sig.generateSignature(img);
```

```
UPDATE livres L  
SET L.photo = img,  
L.photo_sig = sig  
WHERE L.numero = 1;
```

```
END;
```





# Voir les métadonnées (1)

```
DECLARE
```

```
  image ORDSYS.ORDImage; titre_livre varchar2(30);
```

```
  verifie BOOLEAN;
```

```
BEGIN
```

```
  SELECT titre, photo INTO titre_livre, image FROM livres WHERE numero=1;  
  dbms_output.put_line(' Numéro : ' || num || ' - Titre : ' || titre_livre);
```

```
  verifie := image.checkProperties();
```

```
  IF verifie THEN
```

```
    dbms_output.put_line('Hauteur :      ' || image.getHeight() );
```

```
    dbms_output.put_line('Largeur :      ' || image.getWidth() );
```

```
    dbms_output.put_line('Type MIME :      ' || image.getMimeType() );
```

```
    dbms_output.put_line('Format de Fichier : ' || image.getFileFormat() );
```

```
    dbms_output.put_line('Taille :      ' ||
```

```
    TO_CHAR(image.getContentLength() ) );
```

```
  END IF;
```

```
END;
```

# Voir les métadonnées (2) en XML

SQL> begin

```
3      for c in (select id, prop from photos) loop
4          Dbms_Output.Put_Line('Le Id est : ' || c.id);
5          Dbms_Output.Put_Line('Le fichier XML des propriétés : ');
6          printClobOut(c.prop); -- Une procédure qui permet d'afficher
           le contenu de CLOB à l'écran (à définir)
7      end loop;
8 end;
9 /
```

Le Id est : 0

# XML de métadonnées d'une image

Le fichier XML des propriétés :

```
<?xml version="1.0"?>
<!-- Generated by Oracle interMedia -->
<ImageAnn>
  <Attributes>
    <MEDIA_SOURCE_MIME_TYPE desc="MIME Type of the media/its
      samples" dt="java.lang.String"> <![CDATA[image/jpeg]]>
    </MEDIA_SOURCE_MIME_TYPE>
    <MEDIA_SIZE desc="Image contentLength" dt="java.lang.Integer">
      <![CDATA[131347]]> </MEDIA_SIZE>
    <MEDIA_SOURCE_FILE_FORMAT desc="Image fileFormat"
      dt="java.lang.String"> <![CDATA[JFIF]]>
  </MEDIA_SOURCE_FILE_FORMAT>
    <IMAGE_CONTENT_FORMAT desc="Image contentFormat"
      dt="java.lang.String"> <![CDATA[24BITRGB]]>
  </IMAGE_CONTENT_FORMAT>
    <IMAGE_COMPRESSION_FORMAT desc="ImagecompressionFormat"
      dt="java.lang.String"> <![CDATA[JPEG]]>
  </IMAGE_COMPRESSION_FORMAT>
    <IMAGE_HEIGHT desc="Image Height" dt="java.lang.Integer">
      <![CDATA[1536]]> </IMAGE_HEIGHT>
    <IMAGE_WIDTH desc="Image width" dt="java.lang.Integer">
      <![CDATA[2048]]> </IMAGE_WIDTH>
  </Attributes>
</ImageAnn>
```

retourne les valeurs de tous les attributs du type ORDImage. – » la balise [CDATA[.....]].



# Repérage par similarité (1/2)

- Declaration d'un curseur

```
DECLARE
```

```
    seuil    NUMBER;
```

```
    compare_sig  ORDSYS.ORDImageSignature;
```

```
    photo       ORDSYS.ORDIMAGE;
```

```
    numero_photo  INTEGER;
```

```
    titre_photo   VARCHAR2(30);
```

```
CURSOR rech_photos IS
```

```
    SELECT L.numero, L.photo
```

```
    FROM livres L
```

```
    WHERE ORDSYS.IMGSimilar(L.photo_sig, compare_sig,  
'color="0,6" texture="0,2" shape="0,1" location="0,1"', seuil
```



# Repérage par similarité (2/2)

```
BEGIN
```

```
  -- choisir une image-requête
```

```
  SELECT L.photo_g_sig          INTO compare_sig FROM livres L  
  WHERE L.numero = 1;
```

```
  seuil := 30.0;
```

```
  -- repérage
```

```
  OPEN rech_photos;
```

```
  LOOP
```

```
    FETCH rech_photos INTO numero_photo, photo;
```

```
    EXIT WHEN rech_photos%NOTFOUND;
```

```
    -- output the result
```

```
    dbms_output.put_line(numero_photo);
```

```
  END LOOP;
```

```
  CLOSE rech_photos;
```

```
END;
```

# Requête mixte

```
SELECT Q.numero, Q.titre, Q.photo.getCompressionFormat()  
FROM Livres Q, Livres E  
WHERE E.numero=1 AND  
      ORDSYS.IMGSimilar(Q.photo_sig, E.photo_sig,  
                          'color=1', 50.0)=1  
AND Q.photo_g.getWidth() > 32  
AND Q.photo_g.getContentLength() > 10000;
```

# Nouveau dans 10/11g

[http://www.doag.org/pub/docs/sig/intermedia/2003-11/DOAG2003\\_TextNewFeatures\\_DE.ppt](http://www.doag.org/pub/docs/sig/intermedia/2003-11/DOAG2003_TextNewFeatures_DE.ppt)

- Support du standard SQL/MM Still Image
- Nouvelle version du JAI (Java Advanced Imaging) et ajout d'opérateurs pour traiter les images
- Support pour des formats multimédia additionnels
  - Microsoft ASF, MPEG2 & MPEG4
- Nouveaux serveurs de Plugin
  - Microsoft Windows Media Server Plugin
  - Real Server Plugin for Helix Server
- Intégration XML DB

# SQL/MM Still Image Standard

[http://www.doag.org/pub/docs/sig/intermedia/2003-11/DOAG2003\\_TextNewFeatures\\_DE.ppt](http://www.doag.org/pub/docs/sig/intermedia/2003-11/DOAG2003_TextNewFeatures_DE.ppt)

- Oracle10g supporte la première version du ISO/IEC 13249-5:2001 SQL/MM Part 5: Still Image Standard
- Cette norme définit des types objet-relationnel pour les images et leurs caractéristiques
- Chaque type inclut des attributs, des méthodes et leurs fonctions et procédures SQL associées



# SQL/MM StillImage

- SQL pour le multimédia (SQL/MM)
- Types objets (Oracle 10g – *interMedia* reference chapitre 6)
  - SI\_AverageColor: Couleur Moyenne
  - SI\_Color: Couleur en RGB
  - SI\_ColourHistogram: Histogramme de couleur
  - SI\_PositionalColor: Couleurs par région (rectangles)
  - SI\_Texture: Texture (régularité, direction, échelle)
  - SI\_FeatureList: Vecteur de traits (SI\_AverageColor, SI\_ColourHistogram, SI\_PositionalColor, SI\_Texture)
  - SI\_StillImage: Métadonnées de l'image

# JAI (Java Advanced Imaging)

[http://www.doag.org/pub/docs/sig/intermedia/2003-11/DOAG2003\\_TextNewFeatures\\_DE.ppt](http://www.doag.org/pub/docs/sig/intermedia/2003-11/DOAG2003_TextNewFeatures_DE.ppt)

- Support pour JAI 1.1.1\_01, la dernière version du SUN Open Standard for Image Processing
- Nouveaux opérateurs
  - Arbitrary Image Rotate
  - Flip & Mirror
  - Page extract from a multi page TIFF file
  - Contrast Enhancement
  - Quantize algorithm
  - Gamma Correction

# Microsoft ASF & Windows Media Server

[http://www.doag.org/pub/docs/sig/intermedia/2003-11/DOAG2003\\_TextNewFeatures\\_DE.ppt](http://www.doag.org/pub/docs/sig/intermedia/2003-11/DOAG2003_TextNewFeatures_DE.ppt)

- Advanced System Format est devenu un format de flu populaire sur le web
  - Oracle10g Database peut parser des fichiers de métadonnées en format ASF
- Windows Media Server
  - Un plugin développé par Oracle pour supporter les Microsoft Windows Media qui activent des flux ASF
  - Support équivalent à Oracle9i Database pour serveur de flux Real Networks
  - Disponible via OTN



# Travaux personnels et labo

- Faites les exercices du Chapitre 11
- Continuez la lecture de la documentation Oracle proposée
- Continuez votre laboratoire

# Sommaire du cours 8

- Problématiques du repérage d'images
- Différents traits caractéristiques peuvent être extraits des images
- Systèmes commerciaux utilisent des traits de
  - Couleur
  - Texture
  - Forme
- Futur : traits caractérisant l'apparence ?