Bases de données multimédia

Structures d'index - LSH, applications

Michel Crucianu (prenom.nom@cnam.fr)
http://cedric.cnam.fr/~crucianm/bdm.html

Conservatoire National des Arts & Métiers, Paris, France

25 janvier 2016

Plan du cours

Contenu de la séance

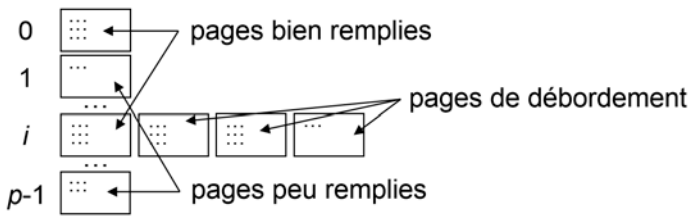
- Hachage sensible à la similarité (*Locality Sensitive Hashing*, LSH)
  - Principe d'indexation
  - Fonctions de hachage et mesures de similarité
  - *Multi-probe* LSH, *A posteriori multi-probe* LSH
  - Hachage et configurations géométriques
- Jointure par similarité
  - Problématique
  - Principes des méthodes de passage à léchelle
- Applications : surveillance de flux vidéo, fouille de bases vidéo
  - Qu'est-ce qu'une copie vidéo ?
  - Détection de copies par le contenu et recherche par similarité
  - Surveillance de flux vidéo par détection de copies
  - Fouille de grandes bases vidéo par détection de copies

le cnam

Hachage

(`http://cedric.cnam.fr/vertigo/Cours/RCP216/coursReductionComplexite.html#le-hachage`)

- Objectif dans ce contexte : trouver directement où est stockée une donnée
- Principe (BD relationnelle) : p pages disque, attribut qui prend des valeurs v dans un domaine $\mathscr{D}$, fonction de hachage $h : \mathscr{D} \rightarrow \{0, 1, 2, \ldots, p-1\}$
- Exemple (BD relationnelle) :
  - Table `film(titre, realisateur, annee)`, hachage sur l'attribut `titre`
  - Fonction de hachage (de préférence, p nombre premier) : h(`titre`) = (somme_codes_ascii_caractères(`titre`)) modulo p



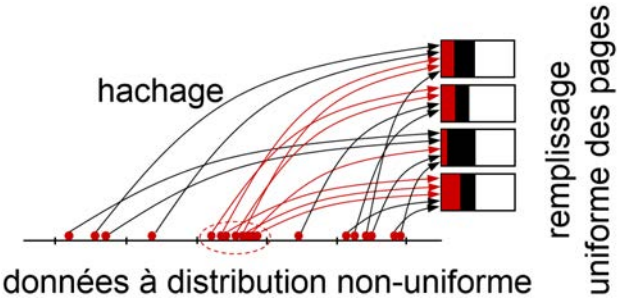
- Propriété recherchée : « disperser » uniformément des données dont la distribution peut être très non-uniforme au départ

le cnam

Hachage (2)

- Recherche par identité : retrouver les autres informations concernant un film à partir de son titre
  - Hachage du titre à rechercher
  - Lecture de la page disque identifiée par le *hash*

  $\Rightarrow$ complexité $O(1)$
- Dispersion uniforme de données non-uniformes $\rightarrow$ destruction de la structure de similarité des données $\Rightarrow$ méthode inadaptée à la recherche par intervalle ou par similarité

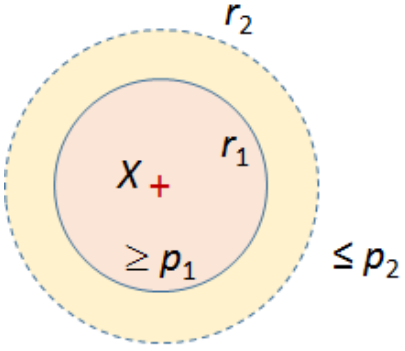


le cnam

Hachage sensible à la similarité (*Locality-Sensitive Hashing, LSH*)

- Données dans un domaine $\mathscr{D}$, ensemble de *hash* (empreinte, condensé) $\mathscr{Q}$, métrique $d_{\mathscr{H}} : \mathscr{D} \times \mathscr{D} \rightarrow \mathbb{R}^+$
- $\mathscr{H} = \{h : \mathscr{D} \rightarrow \mathscr{Q}\}$ est un ensemble de fonctions de hachage (r_1, r_2, p_1, p_2)-sensibles (avec $r_2 > r_1 > 0$, $p_1 > p_2 > 0$) si (voir par ex. [1])

$$\forall x, y \in \mathscr{D}, \quad \begin{array}{l} d_{\mathscr{H}}(x,y) \leq r_1 \;\Rightarrow\; P_{h \in \mathscr{H}} h(x) = h(y) \geq p_1 \\ d_{\mathscr{H}}(x,y) > r_2 \;\Rightarrow\; P_{h \in \mathscr{H}} h(x) = h(y) \leq p_2 \end{array} \tag{1}$$

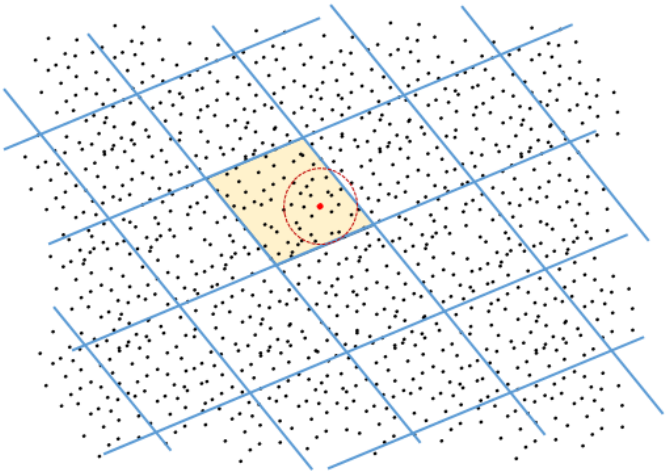


le cnam

Recherche par similarité avec LSH

- Avant toute requête :
  1. Calculer la valeur de *hash* pour chacune des données de la base
  2. Stocker les données de même valeur de *hash* dans une même page (*bucket*)
- Pour chaque donnée-requête :
  1. Hachage de la donnée-requête, lecture de la page (*bucket*) associée au *hash*
  2. Retour des données de cette page
  3. Éventuellement, filtrage de ces données par calcul des distances

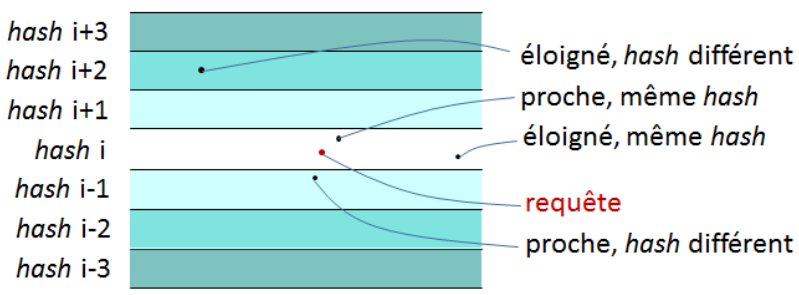
  $\Rightarrow$ Complexité $O(1)$

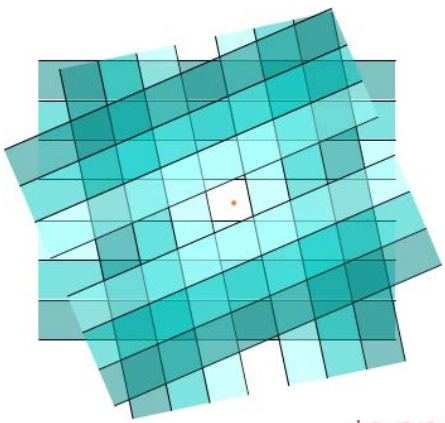


le cnam

LSH pour la métrique euclidienne

- $\mathscr{D} = \mathbb{R}^m$, $\mathscr{Q} = \mathbb{Z}$, $d_{\mathscr{H}}$ est la métrique L_2
- Fonctions élémentaires $h : \mathbb{R}^m \to \mathbb{Z}$, $h_{\mathbf{a},b,w}(\mathbf{x}) = \left\lfloor \frac{\mathbf{a}^T \cdot \mathbf{x} + b}{w} \right\rfloor$ avec $\mathbf{a} \in \mathbb{R}^m$ de composantes tirées indépendamment suivant $\mathscr{N}(0,1)$ et $b \in \mathbb{R}$ tiré suivant la loi uniforme dans $[0,1)$, $w \in \mathbb{R}$
- Table de hachage (ou fonction de hachage composée) : concaténation de (ET logique entre) n fonctions élémentaires indépendantes $\to$ *hash* de n entiers
- Exemple dans $\mathbb{R}^2$, avec 3 fonctions élémentaires :

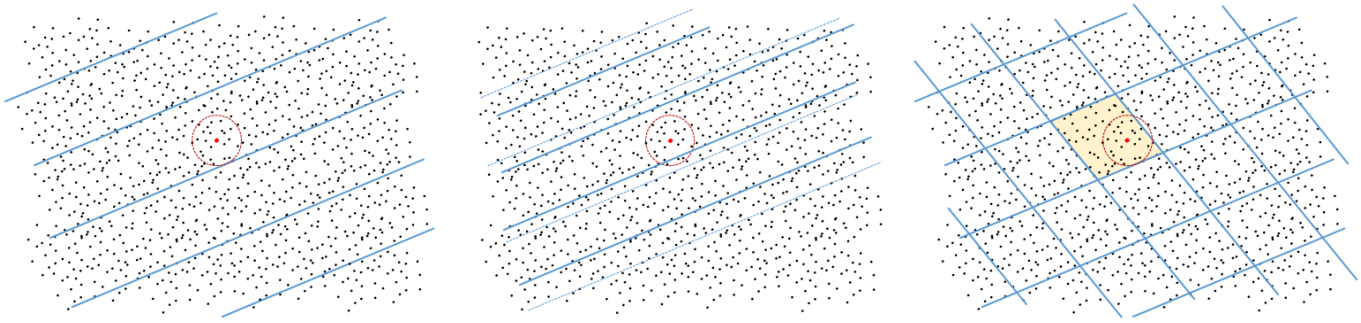




le cnam

LSH : pourquoi regrouper des fonctions de hachage

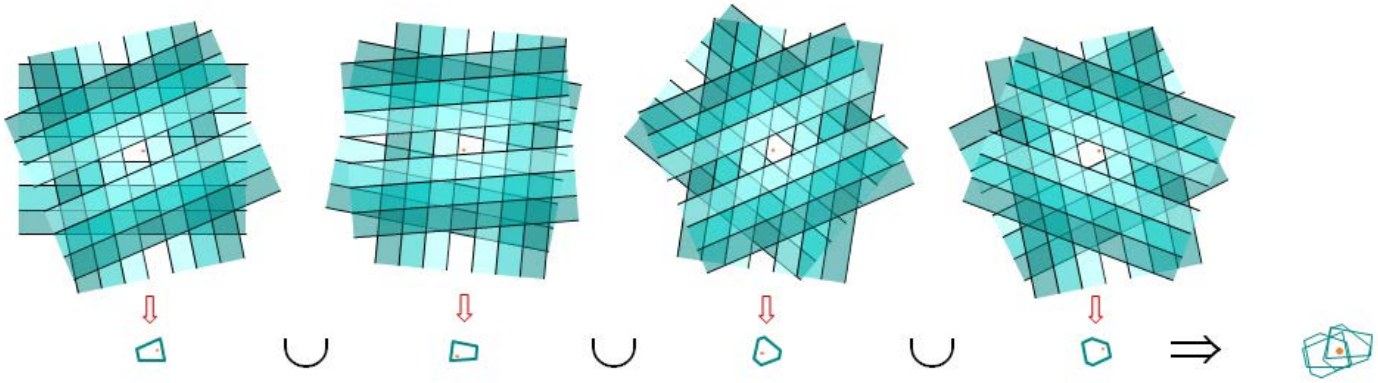
- 1 fonction élémentaire n'est pas assez sélective ⇒ précision insuffisante (trop de faux positifs à filtrer ensuite par de coûteux calculs de distance)
- Grille plus fine ⇒ meilleure précision mais forte réduction du rappel
- Concaténation de (ET logique entre) n (> 1) fonctions de hachage indépendantes ($\rightarrow$ 1 table de hachage) ⇒ améliorer la précision sans diminuer trop fortement le rappel
- Attention, contrairement au cas présenté dans la figure, en général $n \ll$ dimension de l'espace !



le cnam

LSH : pourquoi regrouper des tables de hachage

- 1 table de hachage ⇒ si requête proche d'une frontière alors des données similaires sont de l'autre côté ⇒ faux négatifs (réduction du rappel)
- Réunion (OU logique entre) des résultats issus de t (> 1) tables de hachage indépendantes ⇒ meilleur rappel



le cnam

LSH pour la distance de Hamming

- Motifs binaires de longueur fixe m, $\mathscr{D} = \{0,1\}^m$, $\forall \mathbf{x}, \mathbf{y} \in \{0,1\}^m$, distance de Hamming $d_H(\mathbf{x},\mathbf{y}) =$ nombre de positions sur lesquelles $\mathbf{x}$ et $\mathbf{y}$ diffèrent
- Fonctions de hachage élémentaires : $h_i : \mathscr{D} \to \{0,1\}$, $h_i(\mathbf{x}) = x_i$ (i-ème bit de $\mathbf{x}$)
- Utilisé pour m très élevé et un nombre de fonctions de hachage comparativement faible

$$\begin{array}{c} \downarrow \text{position } i \\ 110\ldots011\ldots010 \\ \underbrace{100\ldots010\ldots100}_{m \text{ bits}} \end{array}$$

le cnam

Similarité entre ensembles finis

- Indice (ou similarité) de Jaccard :
  - Soit un ensemble $\mathscr{E}$, $\mathscr{D} = \mathscr{P}(\mathscr{E})$ est l'ensemble des parties de $\mathscr{E}$
  - Similarité entre deux sous-ensembles $\mathscr{A}, \mathscr{B} \in \mathscr{P}(\mathscr{E})$: $s_J(\mathscr{A},\mathscr{B}) = \frac{|\mathscr{A} \cap \mathscr{B}|}{|\mathscr{A} \cup \mathscr{B}|}$
  - $d_J(\mathscr{A},\mathscr{B}) = 1 - s_J(\mathscr{A},\mathscr{B})$ est une métrique sur $\mathscr{P}(\mathscr{E})$
- Définir les ensembles :
  - Texte : mots
  - Profil d'achat : ensemble d'articles achetés ou ensemble de *catégories* d'articles achetés
- Pour réduire l'ordre de complexité de la recherche par similarité (éviter de comparer une requête à chacune des N données de la base) : définir des fonctions LSH adaptées et s'en servir pour une recherche $O(1)$

le cnam

LSH pour ensembles

- Fonctions de hachage élémentaires : $h_\pi : \mathscr{P}(\mathscr{E}) \to \mathscr{E}$, $h_\pi(\mathscr{A}) = \min \pi(\mathscr{A})$
  - On fixe un ordre des éléments de $\mathscr{E}$
  - π est une permutation des éléments de $\mathscr{E}$
  - $\min \pi(\mathscr{A})$ est l'élément de $\mathscr{A}$ qui se retrouve premier après cette permutation
- Représentation des ensembles par leurs fonctions caractéristiques :
  - Exemple de permutation : $\pi = \begin{pmatrix} a & b & c & d & \dots \\ c & d & a & b & \dots \end{pmatrix}$

$\mathscr{E}$	$\mathscr{A}$	$\mathscr{B}$	$\mathscr{C}$	$\pi(\mathscr{E})$	$\pi(\mathscr{A})$	$\pi(\mathscr{B})$	$\pi(\mathscr{C})$
a	1	0	0	c	1	1	0
b	0	0	1	d	0	0	0
c	1	1	0	a	1	0	0
d	0	0	0	b	0	0	1
...				...			

le cnam

LSH pour ensembles (2)

- Probabilité de collision : pour toute fonction h_π et quels que soient les ensembles $\mathscr{A}, \mathscr{B} \in \mathscr{P}(\mathscr{E})$

$$p(h_\pi(\mathscr{A}) = h_\pi(\mathscr{B})) = s_J(\mathscr{A}, \mathscr{B})$$

- Justification :
  - Soit x le nombre d'éléments communs entre $\mathscr{A}$ et $\mathscr{B}$ (c'est à dire $|\mathscr{A} \cap \mathscr{B}|$)
  - Soit y le nombre d'éléments spécifiques à $\mathscr{A}$ ou $\mathscr{B}$ (c'est à dire $|(\mathscr{A} - \mathscr{B}) \cup (\mathscr{B} - \mathscr{A})|$)
  - Alors $s_J(\mathscr{A}, \mathscr{B}) = \frac{x}{x+y}$
  - Pour toute fonction h_π, la probabilité de trouver en premier après la permutation π un élément commun plutôt qu'un élément spécifique est $\frac{x}{x+y}$
  - Donc $p(h_\pi(\mathscr{A}) = h_\pi(\mathscr{B})) = \frac{x}{x+y}$

le cnam

LSH pour ensembles (3)

1. Construction d'une table de hachage en regroupant n fonctions de hachage choisies aléatoirement de façon indépendante :
   - Le *hash* donné par la table est la concaténation des *hash* des n fonctions
   - 2 données sont en collision dans la table $\Leftrightarrow$ elles sont en collision par rapport à la première fonction de hachage ET par rapport à la deuxième ET ... ET par rapport à la n-ème

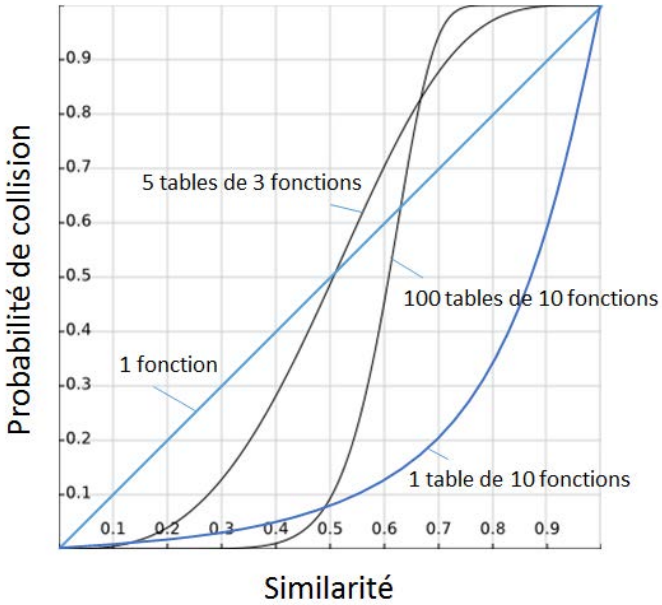
   $\Rightarrow$ probabilité de collision dans la table : s^n (s étant la similarité $s_J(\mathscr{A},\mathscr{B})$)
2. Utilisation de *plusieurs* (t) tables de hachage :
   - 2 données sont en collision si elles sont en collision dans la première table OU dans la deuxième OU ... OU dans la t-ème

   $\Rightarrow$ probabilité de collision dans au moins une table : $1-(1-s^n)^t$
   Justification : probabilité de non collision dans 1 table $= 1-s^n$ $\rightarrow$ probabilité de non collision dans les t tables $=(1-s^n)^t$ $\rightarrow$ probabilité de collision dans au moins 1 table $=1-(1-s^n)^t$

le cnam

LSH pour ensembles (4)

- Graphique probabilité de collision ($\in(0,1)$) fonction de $s_J(\mathscr{A},\mathscr{B})$ ($\in(0,1)$) :



le cnam

Généralisation : « amplification » de fonctions *LSH*

- Fonctions de hachage (r_1, r_2, p_1, p_2)-sensibles (avec $r_2 > r_1 > 0$, $p_1 > p_2 > 0$) :

$$\forall x, y \in \mathscr{D}, \begin{array}{rcl} d_{\mathscr{H}}(x,y) \leq r_1 & \Rightarrow & P_{h \in \mathscr{H}} h(x) = h(y) \geq p_1 \\ d_{\mathscr{H}}(x,y) > r_2 & \Rightarrow & P_{h \in \mathscr{H}} h(x) = h(y) \leq p_2 \end{array} \quad (2)$$

- Amplification : rapprocher p_2 (probabilité de collision pour données dissimilaires) de 0 et p_1 (probabilité de collision pour données similaires) de 1
  - ET logique entre n fonctions : $h_{\text{AND}}(x) = h_{\text{AND}}(y)$ si et seulement si $h_i(x) = h_i(y)$ pour tout i, $1 \leq i \leq n \Rightarrow h_{\text{AND}}$ est (r_1, r_2, p_1^n, p_2^n)-sensible
    → justifie le regroupement de plusieurs fonctions dans une table
  - OU logique entre t fonctions : $h_{\text{OR}}(x) = h_{\text{OR}}(y)$ si et seulement si $h_i(x) = h_i(y)$ pour au moins une valeur de i, $1 \leq i \leq n \Rightarrow h_{\text{OR}}$ est $(r_1, r_2, 1-(1-p_1)^t, 1-(1-p_2)^t)$-sensible
  - $h_{\text{OR,AND}}$ est $(r_1, r_2, 1-(1-p_1^n)^t, 1-(1-p_2^n)^t)$-sensible
    → justifie le regroupement de plusieurs tables

le cnam

Références I


A. Gionis, P. Indyk, and R. Motwani.
Similarity search in high dimensions via hashing.
In *Proceedings of the 25th International Conference on Very Large Data Bases*, VLDB '99, pages 518–529, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc.

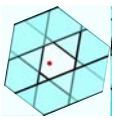
le cnam

le cnam

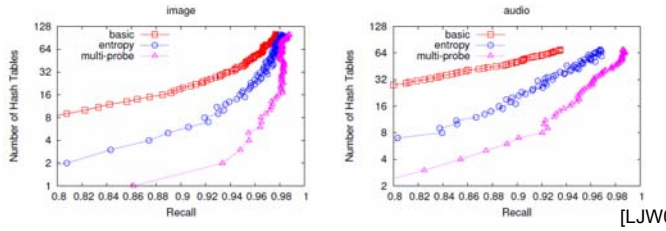
Multi-probe LSH

- Bon rappel avec LSH ⇐ nombreuses tables : occupation mémoire

- *Multi-probe* [LJW07] : dans chaque table échantillonner aussi les *buckets* voisins, avec une probabilité proportionnelle à la proximité



- Comparaison

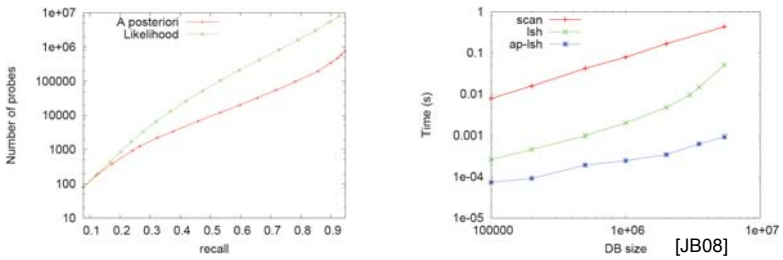


[LJW07]

le cnam

A posteriori multi-probe LSH

- *Multi-probe* LSH [LJW07] : la définition des fonctions de hachage définit le voisinage d'un *bucket* B (l'ensemble des *buckets* à visiter lorsque la clé de la requête correspond à B)
- *A posteriori multi-probe* LSH [JB08]
  - Estimer la probabilité *a posteriori* de trouver des réponses pertinentes dans chaque *bucket* voisin
  - Choisir le nombre minimal de *buckets* tels que la somme de leurs probabilités dépasse un seuil de rappel α

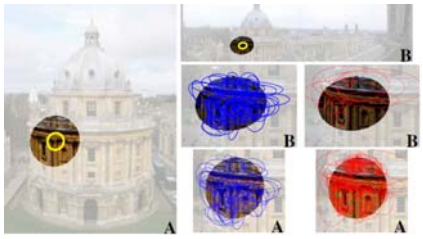


[JB08]

le cnam

Min-hachage géométrique

- [CPM09] : comparaison d'ensembles de points d'intérêt
  - Information de co-occurrence de points → meilleure précision
  - Ensembles de points voisins → meilleure robustesse aux occultations partielles → meilleur rappel
- Construction d'un ensemble : sélection des points en utilisant l'information d'échelle issue du descripteur (SIFT) du point central



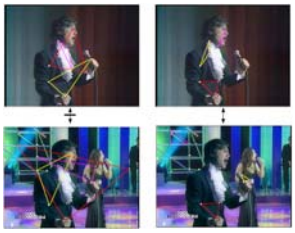
[CPM09]

le cnam

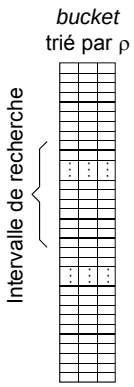
Min-hachage géométrique (2)

- [PCS09] : quantification de l'espace de description en cellules → chaque *bucket* est défini par un triplet de cellules
- Triplets de points définis par les *k*ppv dans l'image
- Information géométrique

$$\rho = \frac{d\left(q, 1^{\text{er}}\ \text{ppv}\right)}{d\left(q, 2^{\text{nd}}\ \text{ppv}\right)}$$



[PCS09]



Exemples de détections

[PCS09]

le cnam

Jointure par similarité

- Problématique
  - Jointure et auto-jointure
  - Rapport entre auto-jointure et classification automatique
- Approches
  - Parcours parallèle d'indexes
  - Partitionnement et jointures intra-partitions

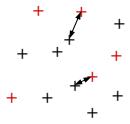
le cnam

Jointure par similarité : problématique

- Jointure avec seuil de distance θ, ensembles $\mathcal{D}_1, \mathcal{D}_2$ avec N_1 et respectivement N_2 éléments

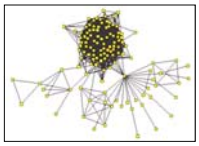
$$K_\theta = \{(\mathbf{x},\mathbf{y}) \mid \mathbf{x} \in \mathcal{D}_1, \mathbf{y} \in \mathcal{D}_2, d(\mathbf{x},\mathbf{y}) \leq \theta\}$$

⇒ complexité $O(N_1 \times N_2)$?



- Auto-jointure : $\mathcal{D}_1 \equiv \mathcal{D}_2$

⇒ complexité $O(N^2)$?

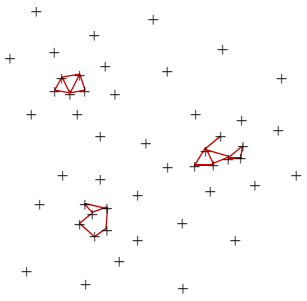


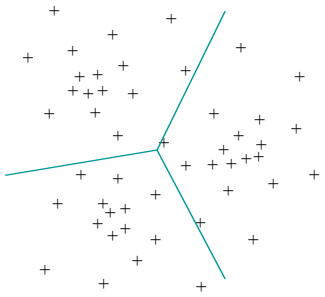
(variante sans seuil de distance : jointure *top-k*)

le cnam

Auto-jointure ↔ classification automatique

- Auto-jointure par similarité
  - Paires de points distance < θ
  - Possible extraction ultérieure de cliques, graphes connexes
  - Autres points : ignorés



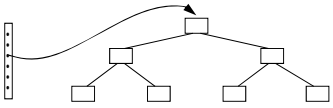


- Classification automatique
  - Regroupement par similarité
  - Chaque point appartient à une partition

le cnam

Jointure par similarité : approches

- Application directe de la recherche par similarité
  - Chaque objet devient une requête
  - Arbre → $O(N \log N)$; LSH → $O(N)$



- Regrouper le travail pour objets-requête similaires
  - Parcours en parallèle de deux arbres → $O(N)$
    mais construction d'arbre $O(N \log N)$!
  - Partitionnement (par ex. LSH) et jointure intra-partitions
    méthodes exactes ou approximatives

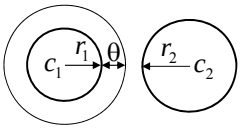
(rappel : ordre de complexité ≥ taille résultat !)

le cnam

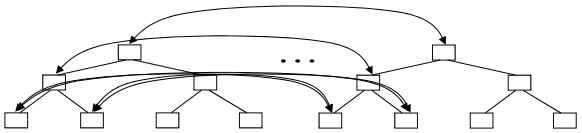
Approche à base de deux arbres

- Idée : parcourir en parallèle deux arbres de recherche
  - Test d'exclusion de 2 sous-arbres

$$d(c_1, c_2) > r_1 + r_1 + \theta$$



- Jointure : algorithme doublement récursif avec 2 arbres

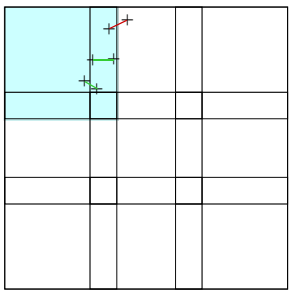


- Auto-jointure : algorithme doublement récursif avec 1 seul arbre

le cnam

Approche par partitionnement

- Idée (auto-jointure) : partitionnement du domaine en *k*, puis jointure intra-partitions ⇒ complexité $O(N^2)$ ↘ $O(N^2/k)$
- Méthode exacte : superposition nécessaire entre partitions voisines → redondance, qui augmente rapidement avec la dimension
- Méthodes approximatives : si partitionnement suffisamment fin, quels que soient 2 points d'une même partition, leur distance est < θ
  - Potentiellement, complexité $O(N)$!



le cnam

Jointure : conclusion

- Nombreux travaux sur le passage à l'échelle de la classification automatique, peu sur la jointure par similarité
- Diverses méthodes exploitent des spécificités des données (distribution particulière, données creuses)
- Extension possible aux ordres supérieurs
- Tenir compte dès le départ de la totalité des critères définissant la similarité (exemple : configurations géométriques)

le cnam

Applications

- Détection de copies vidéo par le contenu (DCPC)
  - Qu'est-ce qu'une copie vidéo ?
  - Détection de copies par le contenu et recherche par similarité
  - Quelle description vidéo pour la détection de copies ?
  - Problème du passage à l'échelle et solutions
- Application à la surveillance de flux vidéo
  - *Workflow* du processus de surveillance de flux vidéo
  - Évaluation de la qualité de détection et du passage à l'échelle
- Application à la fouille de bases vidéo
  - Fouille par DCPC et auto-jointure par similarité
  - Description compacte des images-clés
  - Indexation pour l'auto-jointure par similarité
  - Évaluation de la méthode et illustration des résultats

le cnam

Applications : détection copies vidéo

- Copie = version transformée d'un contenu original
- Transformations rencontrées
  - Photométriques : contraste, gamma, passage en N&B, bruit, flou
  - Géométriques : recadrage, changement d'échelle, étirement
  - Temporelles : changement de tempo, ajout et suppression d'images
  - Post-production : extraits, compilations, ajout de logos, de sous-titres, incrustations, bandes noires, ré-encodage, etc.
- Images-clés illustrant des exemples de copies

25 janvier 2016 Bases de données multimédia 13

le cnam

Détection copies par contenu (DCPC)

- Intérêt de la détection de copies
  - Protection des droits : détection de transmissions potentiellement illicites dans des flux vidéo (hertziens, satellitaires, par câble, Internet) ou dans des bases vidéo (Web2.0, pair-à-pair)
  - Suivi des contrats de diffusion et/ou facturation automatique : toute re-transmission d'un extrait est facturée automatiquement, de façon précise (durée)
  - Fouille vidéo basée sur la détection de copies

1. Détection de copies par tatouage (*watermarking*)
   - Uniquement pour originaux tatoués avant toute transmission…
   - Robustesse variable aux différents types d' « attaques »
   - Nombreuses méthodes de tatouage ⇒ surveillance difficile
2. Détection par le contenu : pour garder une partie de l'intérêt du contenu original, la copie doit conserver l'essentiel de « l'information visuelle » présente

25 janvier 2016 Bases de données multimédia 14

le cnam

DCPC et recherche par similarité

- « Conserver l'essentiel de l'information visuelle »
  ⇒ similarité (pas n'importe laquelle !) entre l'original et la « copie »
- Principe de détection de copie vidéo par le contenu

  La vidéo candidate est une copie de l'original si elle est suffisamment similaire (avec une représentation adéquate du contenu et une mesure de similarité adaptée) à l'original

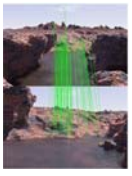
  → La représentation de la vidéo candidate est utilisée comme une requête par similarité dans une base contenant les représentations des vidéos originales

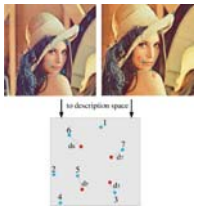
le cnam

Quelle description vidéo pour la DCPC

- Description globale des images ?
  - Peu robustes aux transformations attendues…

→ Description locale des images
  - SIFT, PCA-SIFT, GLOH, SURF ?
    - Trop invariants aux changements d'échelle, angle de vue, etc. ?
    - Très coûteux (extraction, recherche)
  - → Détecteur amélioré de Harris, description différentielle
    - Robustesse (dans des limites assez étroites) aux changements d'échelle, contraste
    - Assez léger (vecteurs de dimension 20)





le cnam

Problème du passage à l'échelle

- Chaque image-clé de la vidéo candidate contient env. 20 points d'intérêt, chaque point est une requête dans la base
- Taille de la base de signatures de référence
  - 60 000 heures → 3 433 853 921 signatures
  - 280 000 heures → 16 354 748 143 signatures…
- Méthode générale
  - Traitement en *batch* (on accumule un nombre élevé de requêtes, on charge successivement en mémoire des parties de la base, on traite toutes les requêtes accumulées par rapport à la partie chargée)

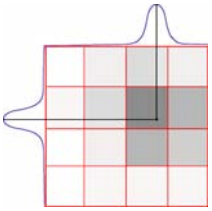
  ⇒ La structure d'index sert essentiellement à réduire le nombre de calculs de distance !

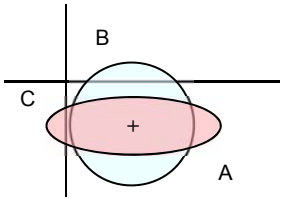
le cnam

Structure d'index et optimisations

- k-d-B-tree (ou LSDh-tree) : pour 60 000 heures de vidéo, 8Gb nécessaires pour stocker les frontières !
- Z-grille : version simplifiée d'un k-d-B-tree, 20 x 2 octets suffisent pour les frontières
- Quel type de recherche ?
  - ε-range : séparation nette, alors que la probabilité des transformations diminue de façon continue avec l'amplitude
  - *k*NN : les *k*NN peuvent être trop loin

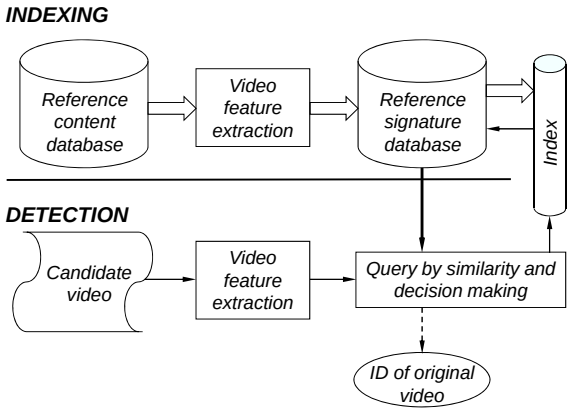
  → Probabiliste (puis ε-range et *k*NN…) : on retient les cellules telles que leur probabilité cumulée dépasse un seuil convenable, ensuite on applique une sélection ε-range et enfin *k*NN sur les résultats





le cnam

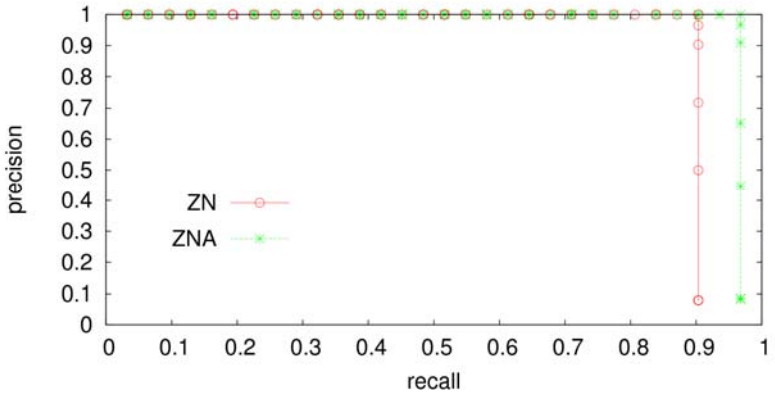
Surveillance de flux : *workflow*



le cnam

Surveillance de flux : qualité

→ Précision et rappel sur vérité terrain CIVR 2007 :

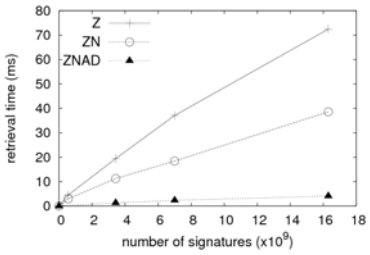


le cnam

Surveillance de flux : rapidité

→ Coût (en milisecondes) par requête

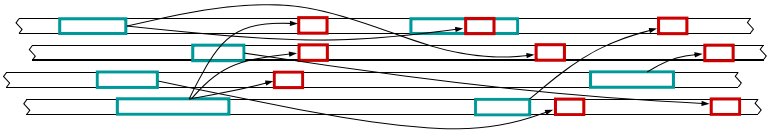
Méthode	ZN	ZNAD
60 000 h	11,6	1,3
120 000 h	18,4	2,35
280 000 h	38,6	4,1



le cnam

Fouille vidéo par DCPC

- Nombreuses applications potentielles :
  - Contexte grande archive de contenu professionnel (exemple : l'INA)
    - Segmentation et étiquetage du contenu (détection génériques)
    - Aide à l'activité des documentalistes (extension d'annotations)
    - Suivi de la diffusion de séquences publicitaires
    - Analyse des offres média (qui reprend quoi et de qui, etc.)
    - Analyse de bruit média et de notoriété, etc.
  - Contexte site Web2.0 (*User Generated Content*)
    - Nettoyage du contenu : élimination des doublons, sélection des meilleurs représentants, meilleure utilisation de la capacité de stockage
    - Mutualisation / filtrage de mots clés entre versions d'un même contenu
    - Réponses plus diversifiées aux requêtes des utilisateurs, etc.



le cnam

Fouille vidéo : description des images

- À la base, même méthode de description qu'auparavant
- Évolutions nécessaires pour améliorer l'efficacité
  1. Réduire le volume des données à traiter
  2. Rechercher par similarité entre signatures d'images clés plutôt que simplement entre signatures de points individuels (inclusion de la décision dans la recherche par similarité)

→ Signature vectorielle compacte d'image clé (*Glocal*)

le cnam

Fouille vidéo : passage à l'échelle

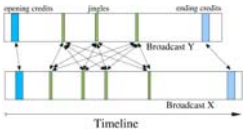
- Le problème : la fouille par détection de copies par le contenu peut être vue comme une auto-jointure par similarité sur la base de séquences vidéo → complexité quadratique (sans index) !
- Solutions envisagées
  1. Utilisation directe de la méthode de surveillance de flux : lecture séquentielle du contenu de la base, qui devient ainsi un flux, et recherche par similarité à partir de chaque image-clé
  2. Autre méthode
     - Segmentation de la base (avec redondance entre segments !) par similarité : à l'intérieur d'un même segment, la similarité est nécessairement supérieure à un seuil
     - Jointure par similarité à l'intérieur de chaque segment si la similarité exigée pour la détection de copies est supérieure au seuil employé pour la segmentation

le cnam

Fouille vidéo : qualité, post-traitements

→ Précision et rappel sur vérités terrain (durée fouille < 1 minute !) :
- Base CIVR 2007 (env. 80 h) : précision 0,96 pour rappel 0,8
- Base INA (env. 30 h) : précision 0,95 pour rappel 0,84

- Post-traitements : séparer les différents types de liens de contenu à partir de la structure des liens et des caractéristiques des séquences reliées
  - Habillage TV : séparation des génériques de début et de fin des autres séquences d'habillage



  - Séparation des séquences publicitaires
  - Identification des retransmissions de programmes
  - Identification des « compilations »

le cnam

Fouille vidéo : rapidité

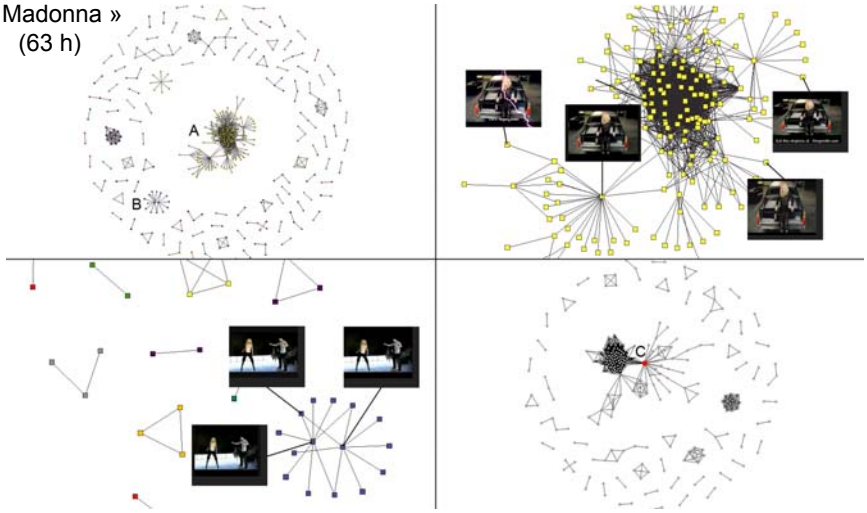
→ Rapidité sur plus grandes bases

Taille base	2 000 h	5 000 h	10 000 h
Nombre de trames-clés	$5{,}8 \times 10^6$	$14{,}5 \times 10^6$	$28{,}7 \times 10^6$
Construction base	2h 35min	3h 38min	7h 00min
Création liens entre trames	5h 40min	14h 59min	55h
Création liens entre séquences	1h 15min	7h 15min	20h 35min

le cnam

Fouille vidéo : illustration Web2.0

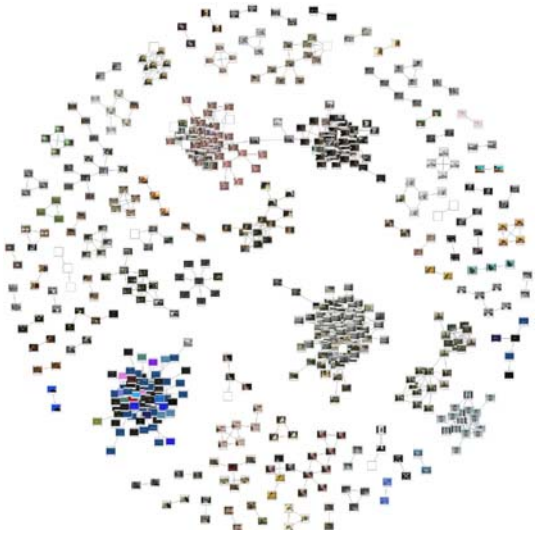
« Madonna »
(63 h)



le cnam

Fouille vidéo : illustration Web2.0

« *funny cats* »
(36 h)



le cnam

Références

[AMNSW98] S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman, A. Y. Wu. An optimal algorithm for approximate nearest neighbor searching in fixed dimensions. Journal of the ACM 45(6): 891-923, November 1998.

[Cha02] M. S. Charikar. Similarity estimation techniques from rounding algorithms. In STOC'02: Proc. 34th ACM Symposium on Theory Of Computing, pages 380–388, New York, NY, USA, 2002. ACM.

[CPIZ07] O. Chum, J. Philbin, M. Isard, and A. Zisserman. Scalable near identical image and shot detection. In Proc. 6th ACM intl. Conf. on Image and Video Retrieval (CIVR'07), pp. 549-556, Amsterdam, The Netherlands, 2007. ACM Press.

[CPM09] O. Chum, M. Perd'och, and J. Matas. Geometric min-hashing: Finding a (thick) needle in a haystack. In CVPR'09: IEEE conf. on Computer Vision and Pattern Recognition, pages 17–24, June 20–26, 2009.

[DII04] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. In SCG'04: Proc. 20th annual Symposium on Computational Geometry, pages 253–262, New York, NY, USA, 2004. ACM.

[IM98] P. Indyk and R. Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In STOC'98: Proc. ACM symp. on Theory of Computing, pages 604–613, New York, NY, USA, 1998. ACM.

le cnam

Références

[JB08] A. Joly and O. Buisson. A posteriori multi-probe locality sensitive hashing. In MM'08: Proc. ACM Multimedia, pages 209–218, New York, NY, USA, 2008. ACM.

[LJW07] Q. Lv, W. Josephson, Z. Wang, M. Charikar, and K. Li. Multi-probe LSH: efficient indexing for high-dimensional similarity search. In VLDB'07: Proc. intl. conf. on Very Large Data Bases, pages 950–961. VLDB Endowment, 2007.

[PCS09] S. Poullot, M. Crucianu, and S. Satoh. Indexing local configurations of features for scalable content-based video copy detection. In LS-MMRM: 1st Workshop on Large-Scale Multimedia Retrieval and Mining, with ACM Multimedia 2009, pp. 43–50, New York, NY, USA, 2009. ACM.

[Sam06] H. Samet. Foundations of Multidimensional and Metric Data Structures. Morgan Kaufmann Publishers, 2006, 1024 p.

[WTF09] Y. Weiss, A. Torralba, and R. Fergus. Spectral hashing. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, Advances in Neural Information Processing Systems 21, pages 1753–1760. 2009.