



**HAL**  
open science

# Structuration géo-temporelle incrémentale hiérarchique d'une collection d'images pour sa gestion sur un mobile

Antoine Pigeau, Marc Gelgon

► **To cite this version:**

Antoine Pigeau, Marc Gelgon. Structuration géo-temporelle incrémentale hiérarchique d'une collection d'images pour sa gestion sur un mobile. Congrès Reconnaissance des Formes et Intelligence Artificielle (RFIA'2006), Jan 2006, France. pp.122-131. hal-00416000

**HAL Id: hal-00416000**

**<https://hal.science/hal-00416000v1>**

Submitted on 15 Sep 2009

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Structuration géo-temporelle incrémentale hiérarchique d'une collection d'images pour sa gestion sur un mobile

## Building and tracking hierarchical geo-temporal partitions for image collection management on mobile devices

A. Pigeau M. Gelgon  
LINA Fre 2927 CNRS / groupe INRIA ATLAS

2, rue de la Houssinière - BP 92208, 44322 Nantes cedex 03 - France

Mail : prénom.nom@univ-nantes.fr

### Résumé

*L'usage de mobiles équipés d'appareil photographique dans la vie courante entraîne un besoin d'organisation de larges collections d'images personnelles. En tenant compte d'études réalisées sur les besoins des utilisateurs, nous proposons un critère statistique associé à un algorithme d'optimisation pour construire une classification géo-temporelle hiérarchique d'une collection. En se basant sur les modèles de mélange, les particularités de notre application et la configuration des données sont prises en compte pour définir notre technique (incrémentalité, robustesse face aux données gaussiennes et gestion des petits échantillons). Des expériences sur une collection d'images réelles sont proposées pour valider notre approche.*

### Mots Clef

Recherche d'image, application sur des mobiles, métadonnées temporelles et spatiales, classification statistique.

### Abstract

*Usage of mobile devices (phones, digital cameras) raises the need for organizing large personal image collections. In accordance with studies on user needs, we propose a statistical criterion and an associated optimization technique, relying on the geo-temporal metadata, for building and tracking a hierarchical structure on the image collection. In a mixture model framework, particularities of the application and typical data sets are taken into account in the design of the scheme (incrementality in the optimization phase, non-Gaussianity and small and large samples in the modelling phase). Results are reported on a real data set.*

### Keywords

Image retrieval, mobile applications, spatio-temporal metadata, statistical clustering.

## 1 Introduction

L'usage de mobiles équipés d'appareil photographique (téléphone portable, appareil photographique numérique)

dans la vie courante permet la constitution de larges collections d'images. Il est donc nécessaire de fournir des outils aux utilisateurs afin de pouvoir facilement rechercher des images parmi plusieurs milliers dans leur collection. Ce domaine de recherche est maintenant actif et a été l'objet de plusieurs travaux intéressants. Les systèmes Nokia LiFeblog et Microsoft MyLifeBits sont deux solutions industrielles récentes. La particularité de la tâche, comparée aux systèmes classiques de recherche de données multimédias basées sur le contenu, vient de la disponibilité de nouvelles métadonnées accessibles via le support d'acquisition (date, lieu, paramètres optiques) et des différences sur les critères de recherches préférés par les utilisateurs. Des études sur ce dernier point, rapportées dans [16, 20], ont montré que les utilisateurs recherchaient leurs images selon les événements, relations sociales, ou les lieux de prises de vues des images.

Sur ce thème, des travaux proposent des techniques de classification supervisée basées sur le contenu des images (détection des images prises à l'intérieur/extérieur [11], détection de visages [7], tâche maintenant raisonnable à implémenter sur des PDAs avec des algorithmes de faible complexité [18]). Dans notre cas, nous utilisons seulement les métadonnées temporelles et spatiales associées à chaque image. Nous supposons que la localisation des images est fournie par un système du type GPS/E-OTD, les métadonnées sont donc un flux de triplets  $\{(t, (x, y)) \in \mathbb{R} \times \mathbb{R}^2\}$ . Bien que illustrée ici à une collection d'images, notre proposition a une application plus générale. Notre contribution est une technique automatique, incrémentale et hiérarchique pour structurer une collection d'images, en se basant sur les métadonnées temporelles et spatiales. Notre hypothèse est que le processus de génération des images présente des classes géo-temporelles sous forme de hiérarchie. La tâche consiste ainsi à organiser les différents épisodes de la vie de l'utilisateur, en les résumant avec les images associées. Les classifications hiérarchiques sont mises à jour incrémentalement, au fur et à mesure de l'ajout de nouvelles images dans la collection. L'extraction

d'épisodes spatiaux et temporels est motivée par les raisons suivantes :

1. ces axes temporels et spatiaux sont simples et familiers aux utilisateurs (similaire à un calendrier et une carte géographique), plus que ne peuvent l'être la couleur ou la texture des images ;
2. l'utilisateur se rappelle généralement d'au moins une des deux métadonnées pour l'image qu'il recherche [19]. Il peut alors combiner les partitions temporelles et spatiales, en passant de l'une à l'autre, pour obtenir l'épisode recherché ;
3. le parcours, plutôt que des requêtes, permet à l'utilisateur de naviguer dans sa collection personnelle sans avoir d'objectif précis (pour le plaisir ou pour avoir un résumé des événements) ;
4. parcourir selon ces axes est aisé à partir d'un mobile. Malgré les contraintes d'interaction homme-machine, un mobile présente l'avantage d'être toujours disponible.

Nous formulons notre recherche des partitions hiérarchiques temporelle et spatiale comme une tâche de classification non-supervisée. Deux classifications hiérarchiques distinctes sont construites, une temporelle et une spatiale, respectivement à partir de la date de prise de vue des images et de leurs coordonnées géographiques mesurées.

Nous avons choisi une approche basée sur les modèles de mélange. Ce choix permet d'éviter le problème d'explosion combinatoire inhérente aux problèmes de regroupement de données et convient bien à la nature incrémentale de la tâche. Les affectations des données aux classes sont flexibles et permettent une évolution souple des partitions au fur et à mesure de l'ajout de nouvelles données. Plus précisément, la technique présente les caractéristiques suivantes :

1. la recherche des épisodes géo-temporels est basée sur un critère statistique (la vraisemblance complétée intégrée) présentant de bonnes propriétés pour notre objectif (détermination de la complexité du modèle, robustesse face aux données non gaussiennes et gestion des petits échantillons) ;
2. l'algorithme est incrémental. Une procédure de recherche, basée sur l'algorithme EM et des tests de fusions et divisions de composantes, permet d'éviter de mauvais optima locaux du critère et de mettre à jour les partitions (en particulier le nombre de classes) ;
3. la classification hiérarchique est construite de bas en haut et mise à jour localement ;
4. dans toutes ces phases, nous évitons de fixer des paramètres critiques arbitrairement

Ce papier est organisé comme suit. La section 2 propose un bref état de l'art sur la question que nous traitons. La section 3 détaille notre proposition : la modélisation probabiliste et sa phase d'optimisation associée et le processus de construction de la classification hiérarchique sont

présentés. La section 4 fournit des résultats expérimentaux. Et enfin, la section 5 résume nos travaux et propose quelques perspectives.

## 2 Etat de l'art

L'exploration par la date ou le lieu des images d'une collection personnelle est proposée par de nombreux logiciels (le système Microsoft World Wide Media eXchange, Picasa "hello" ([www.hello.com](http://www.hello.com)), Cognima ([www.cognima.com](http://www.cognima.com))). Néanmoins aucun ne propose d'organisation automatique des images, celles-ci devant être organisées manuellement. Que le processus d'organisation soit manuel ou automatique (avec des erreurs possibles) est encore sujet à débat [12]. Nous pensons que l'automatisation de cette tâche est bénéfique, notamment dans le cas où le nombre d'images devient assez grand.

Parmi les méthodes automatiques existantes, l'information temporelle est un critère d'organisation pertinent : il est facilement interprétable et disponible. La segmentation temporelle est proposée dans [9, 10, 15], où les changements d'événements sont détectés incrémentalement. Des limites, fixées manuellement, définissent la notion de changement d'événements. La structuration temporelle peut aussi être combinée avec le contenu des images [3, 10] ou les caractéristiques de la prise de vue de l'image [6] (focale, temps d'exposition,...). La structuration basée sur le lieu est proposée dans [8], mais en mesurant continuellement les déplacements de l'utilisateur dans le temps (plutôt que des lieux d'acquisition d'images ponctuelles). Des partitions sont extraites à plusieurs niveaux de détails, basés sur un modèle de trajectoire paramétrique. On tente ainsi de récupérer des épisodes temporels et des lieux significatifs. Les travaux récents décrits dans [1] proposent des techniques d'organisation automatiques, mais contribuent surtout aux mécanismes d'exploration.

Le travail le plus proche de notre article, décrit dans [13], organise une collection d'images hiérarchiquement, en se basant sur la date et le lieu. Leur proposition est basée sur une série de règles dérivées des attentes des utilisateurs. Bien que ces règles soient judicieuses, elle semble impliquer plus de réglages arbitraires que notre proposition, où, par exemple, les distances inter-événements sont apprises automatiquement. De plus, leur technique n'est pas incrémentale. Un système incrémental nous semble indispensable pour toujours garder une organisation de la collection sans que l'utilisateur soit sollicité. Notre algorithme présentant une complexité nettement plus faible que ceux utilisés pour afficher des vidéos sur des mobiles, il est ainsi possible de l'exécuter en tâche de fond en permanence sur ce type de plate-forme.

## 3 Technique proposée

Nous formulons la recherche de structure hiérarchique géo-temporelle comme une tâche de classification non-supervisée. Deux classifications hiérarchiques distinctes sont construites pour les événements temporels et les lieux,

respectivement à partir des métadonnées temporelles et spatiales. Le même algorithme est employé dans ces deux espaces, indépendamment. L'utilisateur a ensuite la possibilité de combiner les deux partitions pour parcourir sa collection, en fonction des souvenirs qu'il lui reste des événements recherchés. Alternativement, nous avons proposé dans [14] une combinaison des partitions (non hiérarchiques) temporelle et spatiale en une partition hybride géo-temporelle.

### 3.1 Modélisation et critère d'optimisation

**Modèle de mélange.** Nous optons pour une approche basée sur les modèles de mélange, les modèles probabilistes permettant d'obtenir les classes et les affectations des données aux classes. Ce dernier point est pertinent pour notre aspect incrémental, puisque les affectations des données aux classes sont flexibles et peuvent évoluer facilement lorsque l'ajout de nouvelles données dans la classification suggère une reconsidération de la structure précédemment trouvée.

Les données  $D$  (soit les lieux  $(x, y)$  ou les dates  $t$ ) sont ainsi supposées être générées aléatoirement à partir d'une distribution de densité :

$$p(D) = \sum_{k=1}^K p_k \cdot \mathcal{N}(D | \mu_k, \Sigma_k) \quad (1)$$

où les probabilités  $p_k$  sont les proportions de mélanges et  $\mathcal{N}(D | \mu, \Sigma)$  est une distribution gaussienne de centre  $\mu$  et de covariance  $\Sigma$ .

Dans le cadre des modèles de mélange, un bon critère de comparaison entre plusieurs hypothèses de classification ayant un nombre différent de classes est la vraisemblance complétée intégrée (ICL) [2]. En comparaison avec le critère BIC approximant la vraisemblance marginalisée des données, ce critère optimise conjointement la vraisemblance des données et les labels inconnus  $z$  d'affectations des données au modèle. L'introduction de ces variables auxiliaires permet de prendre en compte la qualité de la partition lors de l'estimation des paramètres. Pour une hypothèse  $H_k$ , le critère ICL est défini par :

$$p(D, Z | H_K) = \int p(D, Z | \Theta_K, H_K) p(\Theta_K | H_K) d\Theta_K \quad (2)$$

où  $\Theta_K = (\theta_1, \dots, \theta_K)$  est le vecteur de paramètres de  $H_K$  et  $\theta_i = (p_i, \mu_i, \Sigma_i)$ ,  $1 \leq i \leq K$ . Une approximation de (2) est définie par :

$$ICL = -ML + \frac{\nu_K \log(n)}{2} - \sum_{k=1}^K \sum_{i=1}^n t_{ik} \cdot \log(t_{ik}) \quad (3)$$

où  $ML$  est la log-vraisemblance du modèle de mélange optimisé,  $\nu_K$  est le nombre de paramètres libres du modèle à  $K$  composantes,  $n$  est le nombre de données et  $t_{ik}$  est la probabilité à posteriori pour une observation  $i$  d'avoir pour origine la composante  $k$ . Les  $t_{ik}$  sont en fait les espérances

des affectations binaires probabilisées  $z_{ik}$ . En pratique, les  $z_{ik}$  sont déterminés pendant l'étape E de l'algorithme EM, décrit dans la suite.

L'expression (3) est une variation du critère BIC : le terme de droite ajouté a un comportement entropique et favorise les classes bien séparées [2]. Il permet d'améliorer l'aptitude de ce critère pour identifier correctement les classes non gaussiennes, souvent rencontrées dans notre contexte. La complexité de l'optimisation de ce critère est modérée, comparée à l'estimation de modèles de mélange définis comme robuste face aux données non gaussiennes (par exemple le modèle de mélange de Student).

Il est fréquent qu'une classe soit assignée à un petit échantillon de données, entraînant une mauvaise estimation de sa covariance. Nous traitons le problème en introduisant, à l'étape M de l'algorithme EM, des estimations de covariances régularisées calculées à partir des espérances des distributions à posteriori des matrices de covariances initiales (utilisant les modèles à priori conjugués de Gamma pour le temps (une dimension) et de Wishart pour le lieu (deux dimensions)).

Contrairement à [13], toutes les propriétés recherchées de l'approche sont fournies par un modèle probabiliste et un critère d'optimisation, plutôt qu'une série de règles arbitraires.

**Optimisation du critère proposé.** La recherche de partitions de données incrémentale suppose de pouvoir modifier facilement l'affectation des données aux classes et d'ajuster le nombre de composantes en fonction des nouvelles données. Notre proposition consiste à utiliser la partition obtenue au temps  $t$  comme initialisation de l'optimisation du critère ICL pour la partition à  $t + 1$  : cela garantit la stabilité de la partition au cours du temps et facilite ainsi son exploration par l'utilisateur.

L'algorithme Expectation-Maximization (EM) [4] permet d'optimiser localement le critère ICL à un nombre constant de composantes. Il se décompose en deux étapes, l'étape E, dans laquelle les probabilités d'affectation des données à chaque composante sont calculées, conditionnellement aux paramètres des modèles, et l'étape M, dans laquelle les paramètres des modèles sont estimés en se basant sur l'estimation courante des affectations des données aux modèles. Il reste deux problèmes à résoudre :

- l'évolution du nombre de composantes ;
- l'optimisation du critère ICL (équation 3) : l'utilisateur prenant généralement ses images par paquets, le flux de données ne peut pas être modélisé comme une série de données indépendantes. De plus grands efforts de réorganisation des classes sont ainsi nécessaires que si les données étaient totalement indépendantes. Par exemple, au fur et à mesure que les données du même lieu sont ajoutées, les classes sont optimisées localement sur ce lieu spécifique et peuvent être ainsi plus difficiles à modifier si un nouveau lieu apparaît. L'optimisation au cours du temps est ainsi délicate et des minima locaux sont souvent obtenus si seul un algorithme EM classique

---

**Algorithme 1** algorithme d'optimisation

---

1. ajout d'une nouvelle donnée dans l'ensemble  $D$

Itérer l'algorithme EM, initialisé avec le modèle obtenu après l'ajout de la nouvelle donnée précédente.

Soit  $ICL_1$  le critère ICL obtenu à la convergence et  $\mathcal{M}_1$  le modèle correspondant.

2. phase de division : ordonner les candidats  $\{S_1, \dots, S_d\}$  à diviser selon le critère (5).

**pour** les  $\alpha$  premières composantes ordonnées par entropie décroissante (pratiquement  $\alpha = 5$ ) **faire**

- diviser la composante et mettre à jour le modèle à partir de  $\mathcal{M}_1$

- itérer l'algorithme EM jusqu'à convergence. On obtient un modèle  $\mathcal{M}_2$  avec un critère  $ICL_2$

**si**  $ICL_2 < ICL_1$  **alors**

$\mathcal{M}_1 \leftarrow \mathcal{M}_2, ICL_1 \leftarrow ICL_2$  et aller à l'instruction 2

**fin si**

**fin pour**

3. phase de fusion : ordonner les paires de composantes à la fusion en fonction du critère (4).

**pour** les  $\alpha$  premières composantes ordonnées selon une distance croissante de Mahalanobis (pratiquement  $\alpha = 5$ ) **faire**

- fusionner les composantes et mettre à jour le modèle à partir de  $\mathcal{M}_1$

- itérer l'algorithme EM jusqu'à convergence. On obtient un modèle  $\mathcal{M}_2$  avec un critère  $ICL_2$

**si**  $ICL_2 < ICL_1$  **alors**

$\mathcal{M}_1 \leftarrow \mathcal{M}_2, ICL_1 \leftarrow ICL_2$  et aller à l'instruction 3

**fin si**

**fin pour**

---

est utilisé.

Nous proposons de traiter ces difficultés comme suit. Notre procédure d'optimisation permet des sauts locaux dans l'espace des paramètres en appliquant des fusions et divisions de composantes. Ceci diffère de la proposition [17], celle-ci ne présentant pas d'aspect incrémental et gardant le nombre de composantes constant. On alterne ainsi les phases de recherche semi-locale avec des itérations de l'algorithme EM, jusqu'à convergence. Les deux étapes permettent de minimiser le critère ICL et servent le même objectif : éviter les minima locaux et permettre l'évolution du nombre de composantes au cours du temps.

Critères de fusion et de division :

Le nombre de possibilités de fusions et divisions étant élevé, nous proposons des critères pour ordonner les candidats, les critères proposés dans [17] n'étant pas pertinents dans le cas de petit échantillons. Nous proposons ainsi de se baser sur la distance de Mahalanobis pour ordonner les candidats à la fusion :

$$J_{merge}(i, j, \Theta) = \min\{D(\mu_i, \Sigma_i, \mu_j), D(\mu_j, \Sigma_j, \mu_i)\} \quad (4)$$

où  $D(\mu_j, \Sigma_j, \mu_i) = (\mu_i - \mu_j)^T \cdot \Sigma_j^{-1} \cdot (\mu_i - \mu_j)$ .

Le critère de division est basé sur l'entropie des affectations des données aux composantes définie par (5). Une composante avec une entropie forte suggère que ses données sont aussi associées avec d'autres composantes proches.

$$J_{split}(k, \Theta) = - \sum_{i=1}^n t_{ik} \cdot \log(t_{ik}) \quad (5)$$

Initialisation des paramètres :

Les paramètres initiaux d'une composante de paramètre  $\theta_{i'}$  résultant d'une fusion de deux composantes  $i$  et  $j$  de paramètres  $\theta_i$  et  $\theta_j$  sont définis par :

$$p_{i'} = p_i + p_j \quad \text{et} \quad [\mu_{i'} \quad \Sigma_{i'}]^T = \frac{p_i [\mu_i \quad \Sigma_i]^T + p_j [\mu_j \quad \Sigma_j]^T}{p_i + p_j} \quad (6)$$

Pour une division d'une composante  $\theta_k$  en deux composantes  $\theta_{j'}$  et  $\theta_{k'}$  :

$$p_{j'} = p_{k'} = \frac{p_k}{2}, \quad \mu_{j'} = \mu_k + \epsilon, \quad \mu_{k'} = \mu_k - \epsilon \quad (7)$$

$$\Sigma_{j'} = \Sigma_{k'} = \det(\Sigma_k)^{(1/d)} / I_d \quad (8)$$

où  $\epsilon$  est un vecteur de faible norme colinéaire au vecteur propre de  $\Sigma_k$  de plus grande valeur,  $\det(\Sigma)$  est le déterminant de  $\Sigma$  et  $I_d$  est la matrice d'identité de dimension  $d$  ( $d = 1$  ou  $2$  respectivement pour les données temporelles et spatiales).

Procédure d'optimisation

Quand une nouvelle donnée est ajoutée, notre algorithme incrémental commence d'abord par tester plusieurs divisions de composantes, suivies de fusions de composantes. Chaque test est accompagné d'itérations de l'algorithme EM. Si le critère ICL est amélioré lors d'un test, on retient le nouveau modèle. Dans ce cas, la liste des candidats à diviser et fusionner est recalculée et les tests ré-itérés.

L'approche proposée est un compromis entre la recherche locale dans l'espace des paramètres des modèles et le coût de calcul. Celui-ci est lié au niveau de la re-structuration de la partition après l'ajout d'une nouvelle donnée, qui reste généralement faible.

## 3.2 Algorithme hiérarchique

**Principes généraux.** Quand on parcourt une collection de plusieurs centaines d'images, il peut être utile

d'organiser hiérarchiquement les composantes temporelles et spatiales trouvées précédemment. Une telle organisation est aussi pertinente pour parcourir une collection d'images sur un terminal mobile. Cette section décrit notre proposition pour construire une hiérarchie de modèles de mélange. Comme précédemment, l'arbre est construit incrémentalement : sa structure évolue en fonction de l'ajout des nouvelles données.

Un algorithme hiérarchique de modèles de mélange a été décrit dans [5], mais n'est pas incrémental et fournit des arbres binaires. Notre contribution consiste ainsi à :

- introduire une nouvelle procédure pour mettre à jour la hiérarchie, détaillée dans le paragraphe suivant ;
- garantir la qualité de l'arbre. Une fois l'arbre binaire construit, on fait une sélection sur les niveaux de l'arbre, afin d'éviter les niveaux redondants et inintéressants. Encore une fois, le critère ICL fournit une solution adaptée. La figure 2 décrit ce processus de sélection. En procédant de la racine vers les feuilles, nous cherchons des minima locaux du critère ICL dans l'espace des partitions obtenues à chaque niveau de l'arbre binaire. Ces minima correspondent à des hypothèses de classification plausibles.

**Procédure de mise à jour de l'arbre.** L'idée est de propager les nouvelles données à partir de la racine jusqu'aux feuilles, en mettant à jour chaque niveau de l'arbre et en ne réorganisant entièrement que les sous-arbres où des modifications importantes ont lieu. Au fur et à mesure que la nouvelle donnée descend dans l'arbre, notre algorithme d'optimisation (section 3.1) permet de détecter à quel niveau une modification des partitions a lieu.

L'algorithme, quand il opère à un niveau différent du niveau des feuilles, doit permettre une évolution flexible des classes (en permettant des fusions et des divisions), mais sans glisser vers un minimum local correspondant à une partition déjà existante dans un niveau plus fin de la hiérarchie. Cela est pris en compte en utilisant la procédure suivante. Tout d'abord, on définit une composante comme *modifiée* si l'ensemble de ses données associées est modifié après l'ajout d'une nouvelle donnée et la mise à jour avec notre algorithme 1. La mise à jour d'un noeud  $q$  consiste à (1) reconstruire le modèle de mélanges associé à ses fils (noté  $c_q$ ), (2) appliquer notre algorithme d'optimisation, en limitant le nombre de divisions à 1, afin d'éviter de glisser vers une partition existante dans un niveau inférieur de la hiérarchie. Cette étape permet de déterminer si l'introduction de la nouvelle donnée entraîne la re-structuration du noeud mis à jour. Suivant les modifications obtenues, nous appliquons une de ces règles :

1. si la nouvelle donnée est associée à une composante non-modifiée  $q' \in c_q$ , le sous-arbre de racine  $q'$  est mis à jour ;
2. si la nouvelle donnée est associée à une nouvelle composante  $q'$ , on ajoute la composante  $q'$  à l'ensemble  $c_q$  ;

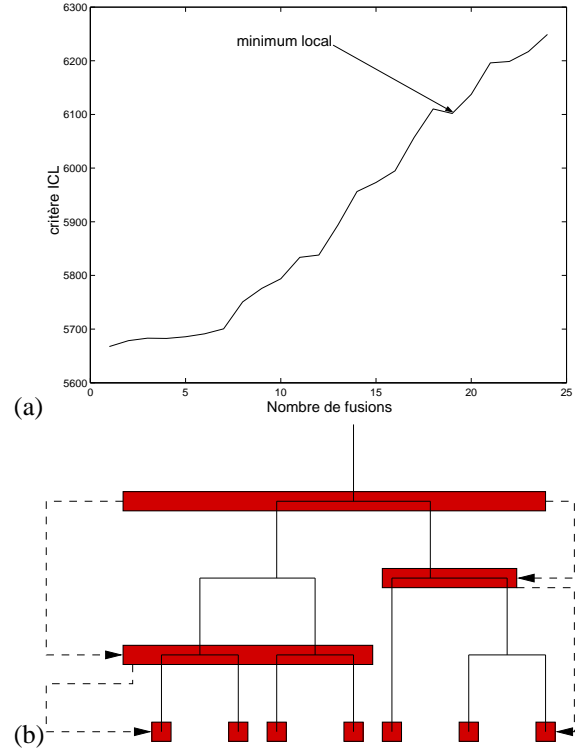


FIG. 2 – Selection de niveaux correspondant à un minimum local du critère ICL : la figure (a) représente la courbe du critère ICL obtenu à chaque niveau de l'arbre binaire représenté par la figure (b). Les rectangles gris indiquent les niveaux sélectionnés. Quand un optimum est trouvé à un niveau donné, nous cherchons d'autres minima locaux dans chaque sous-arbre à partir de ce niveau.

3. si la nouvelle donnée est associée à une composante modifiée, cela implique une re-structuration plus importante des données. Les différentes étapes sont :
  - sélectionner les feuilles des noeuds modifiés et mise à jour du modèle obtenu avec l'algorithme 1 ;
  - reconstruction d'un arbre binaire  $t$  selon la méthode décrite dans [5] à partir des feuilles mises à jour et des noeuds non-modifiés appartenant à  $c_q$  ;
  - optimisation de l'arbre  $t$  avec notre sélection de niveaux (figure 2) ;
  - mise à jour de l'arbre avec le nouveau sous-arbre.

La figure 1 présente un exemple de mise à jour de l'arbre. Une propriété intéressante de notre algorithme est que les propriétés de l'arbre (nombre de fils par noeud, ...) sont automatiquement déterminées par notre algorithme, au lieu d'être définies par des paramètres arbitraires. Le coût de calcul de mise à jour reste modéré puisque nous mettons à jour seulement les sections appropriées de l'arbre.

## 4 Résultats expérimentaux

Nous proposons des classifications hiérarchiques spatiale et temporelle obtenues avec notre technique, à partir d'une

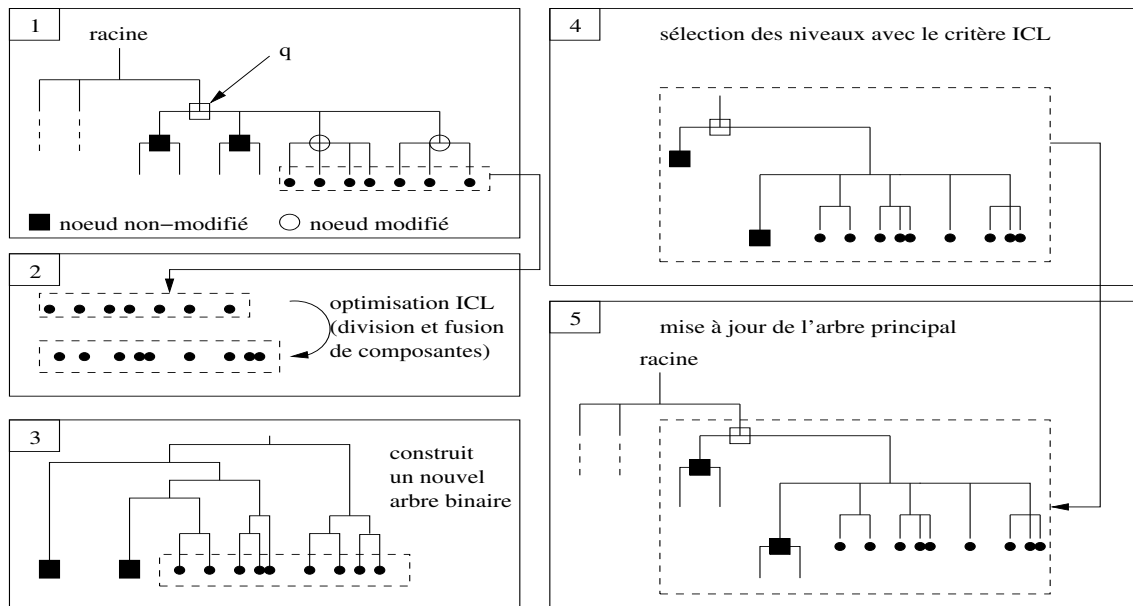


FIG. 1 – Exemple d’une mise à jour de la hiérarchie avec une nouvelle donnée : nous ajoutons une nouvelle donnée par la racine, reconstruisons le modèle  $c_{root}$  et appliquons notre algorithme d’optimisation du critère ICL. Dans cet exemple, aucune composante n’est modifiée et la nouvelle donnée est affectée à la composante non modifiée  $q$ . La mise à jour est donc propagée à ce noeud. Nous reconstruisons donc le modèle  $c_q$  et appliquons notre algorithme d’optimisation du critère ICL. La figure 1 montre les noeuds modifiés et non-modifiés obtenus. Nous supposons ici que la nouvelle donnée est associée à une composante modifiée. Nous récupérons les feuilles des noeuds modifiés et nous appliquons notre algorithme d’optimisation (figure 2). Une arbre binaire est ensuite reconstruit avec [5] à partir des feuilles mises à jour et des noeuds non-modifiés (figure 3), et les niveaux présentant un minimum local du critère ICL sont sélectionnés. La figure 4 présente le nouveau sous-arbre obtenu. Enfin, l’arbre principal est mis à jour en remplaçant le sous-arbre de racine  $q$  par le sous-arbre obtenu précédemment. Notons que les fils des noeuds non-modifiés appartenant à  $c_q$  sont conservés.

collection réelle personnelle composées de 721 images prises sur une durée de trois ans. L'utilisateur a pris les images essentiellement en France, mais aussi au Canada, en Turquie et aux USA. Les métadonnées temporelles étaient directement incluses par l'appareil photographique dans chaque image (métadonnées EXIF) et le lieu a été rajouté manuellement en fonction des coordonnées réelles. Les figures 3 et 4 présentent respectivement les métadonnées temporelles et spatiales.

Nous examinons tout d'abord la structure temporelle obtenue. La figure 5 montre les différents arbres obtenus après l'ajout de 300, 450, 600 et de toutes les images. L'arbre final est représenté par la figure 5(d).

L'arbre obtenu est composé de 4 niveaux et est bien équilibré. Le nombre de fils par noeud varie entre 2 et 14. Sur la figure 5, la classification obtenue croît en largeur et en profondeur au fur et à mesure que l'on rajoute de nouvelles images. La stabilité de l'arbre au cours du processus de classification est correcte ; les arbres obtenus présentent des similarités (carrés en pointillés). Seule une minorité d'images implique une sérieuse re-structuration de l'arbre. La figure 6 présente des partitions obtenues à différents niveaux de notre arbre. La figure 6(a) montre le niveau le plus général. Toutes les composantes sont bien délimitées

et fournissent un bon résumé de la collection. Les composantes 2, 3 et 4 de la figure 6(a) sont respectivement détaillées par les figures 6(d), (b) et (c). Les fils des composantes 4 et 3 fournissent des partitions bien définies puisque les événements temporels sont bien mis en valeur. Pour la composante 2, les épisodes sont correctement trouvés mais on peut noter une sur-segmentation, par exemple les groupes 2.9 et 2.10, due certainement à une trop grande vraisemblance des petits échantillons associés à une composante.

Nos premières expériences ont consisté à classer directement l'ensemble des métadonnées spatiales mais les résultats obtenus étaient peu pertinents à cause des caractéristiques particulières des données. L'utilisateur prend souvent des images sur un même lieu et cela entraîne des groupes compacts souvent concentrés en un point. Nous obtenions ainsi des arbres composés d'un seul niveau (la racine et directement des feuilles) où chaque feuille représentait un lieu de la collection. Notre algorithme ne réussissait pas à obtenir un bon résumé des lieux de la collection à cause de la structure particulière des données. Pour régler ce problème, nous avons décidé de résumer les métadonnées spatiales en ne conservant qu'une seule valeur de chaque coordonnée distincte  $(x, y)$ . Ainsi une co-

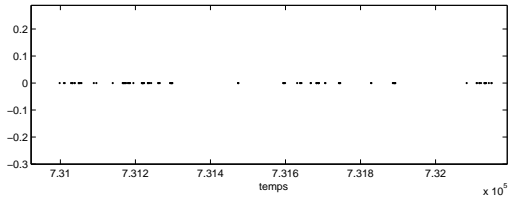


FIG. 3 – Métadonnées temporelles de la collection. Chaque point représente la métadonnée temporelle d’une image.

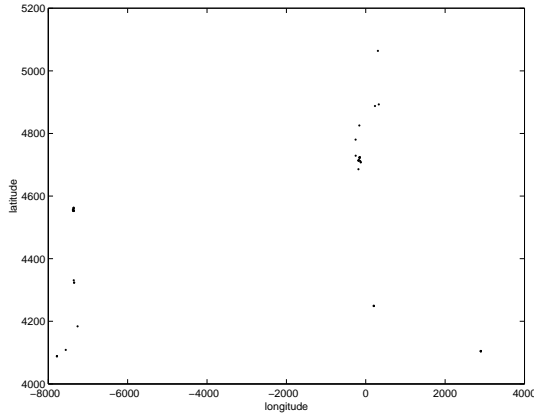


FIG. 4 – Métadonnées spatiales de la collection. Chaque point représente la métadonnée spatiale d’une image. Notons que les données sont souvent concentrées en un seul point (images prises sur un même lieu).

ordonnée  $(x, y)$  peut représenter plusieurs images. Cette étape ramène à 135 le nombre de lieux/images distinct(e)s à classer.

Les figures 7 et 8 montrent respectivement la classification hiérarchique spatiale et des exemples de partitions obtenues à différents niveaux.

L’arbre obtenu est composé de 3 niveaux et le nombre de fils par noeud reste modéré, variant de 2 à 6. Le niveau le plus général est présenté dans la figure 8(a), et la figure 8(b) est un zoom sur les composantes 1, 2 et 3. Les composantes sont distinctes et compactes ce qui est conforme aux caractéristiques des données. Notre algorithme d’optimisation a tendance à regrouper les données isolées, comme on peut le voir avec la composante 3. Cet aspect peut être néanmoins pertinent pour ensuite parcourir la collection (regroupement des lieux isolés adjacents). Les fils des composantes 2 et 7 sont respectivement présentés dans les figures 8(d) et (c). Les deux classifications sont pertinentes puisque les groupes semblent visuellement justifiés.

Afin d’évaluer pratiquement les classifications hiérarchiques obtenues, nous utilisons la même heuristique que [13], la précision et le rappel :

$$précision = \frac{\text{limites détectées correctement}}{\text{total des limites détectées}} \quad (9)$$

$$rappel = \frac{\text{limites détectées correctement}}{\text{total des limites réelles}} \quad (10)$$

L’utilisateur a classé manuellement ses images en 58 épisodes temporels. Nous avons noté qu’il regroupait généralement des images de vacances ou des images isolées successives dans un même groupe (même si le groupe obtenu se déroule sur plusieurs semaines). Pour comparer avec notre résultat, nous avons récupéré toutes les feuilles de notre classification temporelle pour obtenir la classification la plus détaillée. Elle est composée de 107 classes et nous avons obtenu 57 limites correctes :  $rappel = 98\%$ . Nous avons ainsi réussi à retrouver les différents événements de la collection. Puisque nous fournissons des classifications hiérarchiques, nous ne calculons pas la *précision* à partir de la partition des feuilles. Nous proposons de vérifier dans l’arbre temporel si les feuilles sont correctement regroupées dans un sous arbre approprié (si tous les groupes obtenus manuellement sont représentés par un noeud de l’arbre). Nous avons retrouvé 50 épisodes correctement défini dans notre arbre :  $précision = 86\%$ . Les erreurs sont dues aux vacances ou aux images isolées successives classées dans des feuilles séparées.

L’utilisateur a divisé sa collection en 25 lieux différents. Le nombre élevé de feuilles dans notre arbre est dû essentiellement à des images prises lors de balades, qui entraînent de nombreuses classes. Nous avons néanmoins retrouvé 23 lieux correctement regroupés dans des noeuds distincts de notre arbre spatial. Nous avons ainsi retrouvé les différents lieux de la collection. Deux erreurs subsistent : une feuille regroupe deux lieux proches et un lieu est divisé dans deux feuilles différentes. Cette dernière erreur est due à des images prises lors d’une balade, où les données présentent peu de structures. La hiérarchie obtenue est correcte, les noeuds étant facilement interprétables. Par exemple, la composante 2 représente les différents lieux pertinents dans la ville de l’utilisateur tandis que la composante 1 regroupe les lieux aux alentours.

La complexité de notre algorithme peut être évaluée comme suit. Pendant la construction de notre classification temporelle, l’algorithme agglomératif permettant de régénérer un arbre binaire a été appelé dans 25% de cas, et pour chaque cas concernait en moyenne 27% des données. Pour la classification spatiale, il a été appelé dans 59% des cas, et concernait pour chaque cas en moyenne 29% des données (pour les premières données du flux, l’algorithme agglomératif est souvent appelé à cause du manque de stabilité dans les partitions).

## 5 Conclusion

Ce papier propose une technique d’organisation d’une collection d’images personnelles acquises sur un appareil mobile, à partir des métadonnées temporelle et spatiale. Le choix de ces métadonnées est motivé par leur disponibilité et leur bonne interprétation par les utilisateurs. D’autres critères, par exemple sur le contenu de l’image, peuvent bien sûr être utilisés en conjonction avec ces métadonnées. La contribution de ce papier est une technique de classification automatique, limitant les paramètres arbitraires



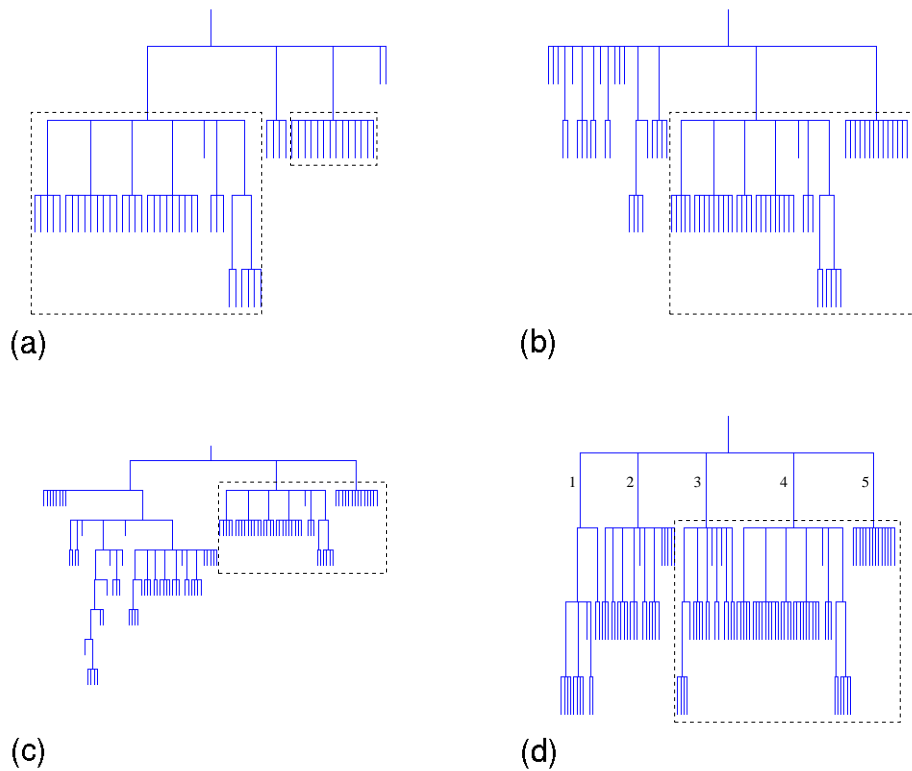


FIG. 5 – Classification hiérarchique temporelle obtenue après 300 (a), 450 (b), 600 (c) et 721 (d) données. La classification croît en largeur et en profondeur au fur et à mesure que l'on rajoute de nouvelles données. Les rectangles en pointillés indiquent les sections des arbres similaires. Les noeuds sur (d) sont numérotés arbitrairement et correspondent à une classe de la figure 6.

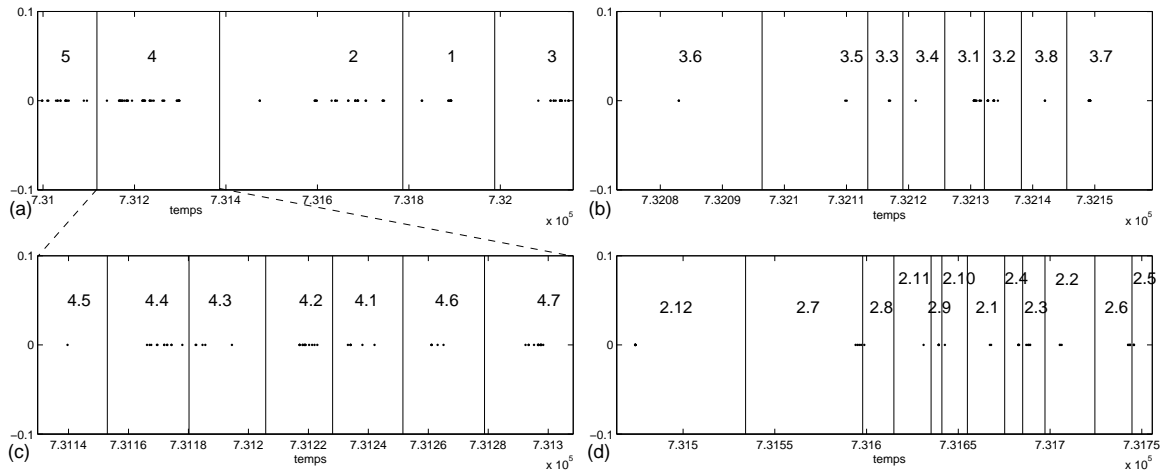


FIG. 6 – Exemple de partitions obtenues dans la classification hiérarchique temporelle. Chaque numéro fait référence à un noeud de l'arbre présenté dans la figure 5(d). Les lignes verticales représentent les limites entre les composantes et les points indiquent les métadonnées temporelles. La figure (a) représente le niveau le plus général de notre classification, et les figures (b), (c) et (d) montrent respectivement les fils des composantes 3, 4 et 2. Les partitions obtenues présentent en général des composantes distinctes avec des limites visuellement justifiées. On note aussi une sur-segmentation des données, par exemple les composantes 2.9 et 2.10.

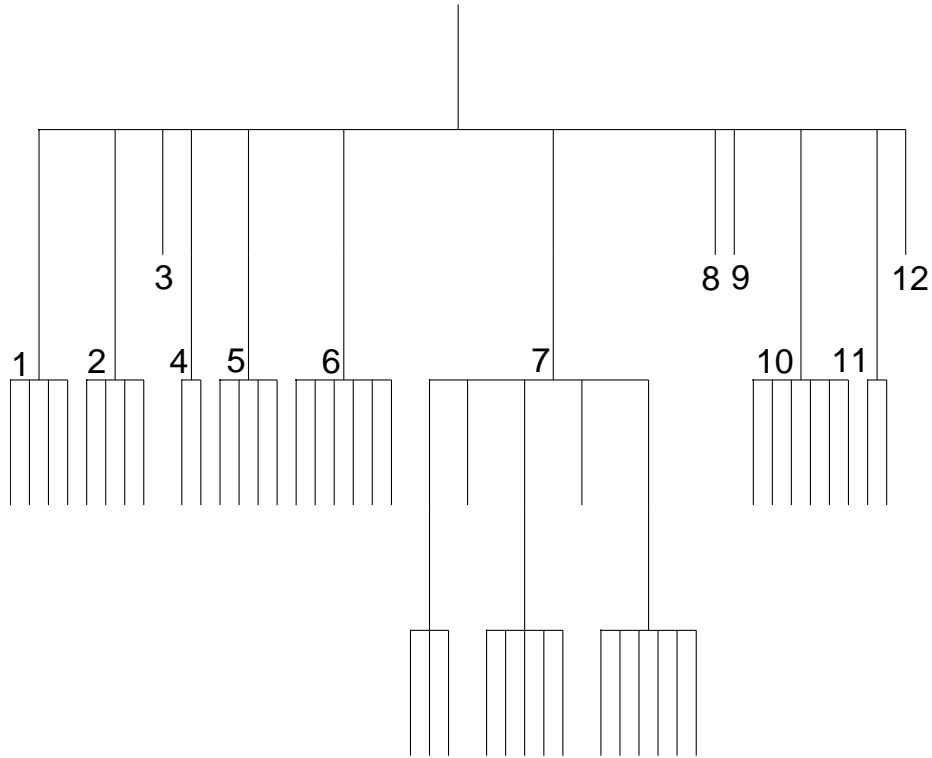


FIG. 7 – Classification hiérarchique spatiale : l’arbre obtenu est bien équilibré et le nombre de fils par noeud varie entre 2 et 6. Le niveau le plus général est présenté dans la figure 8(a). Les numéros des noeuds correspondent aux composantes de la figure 8.

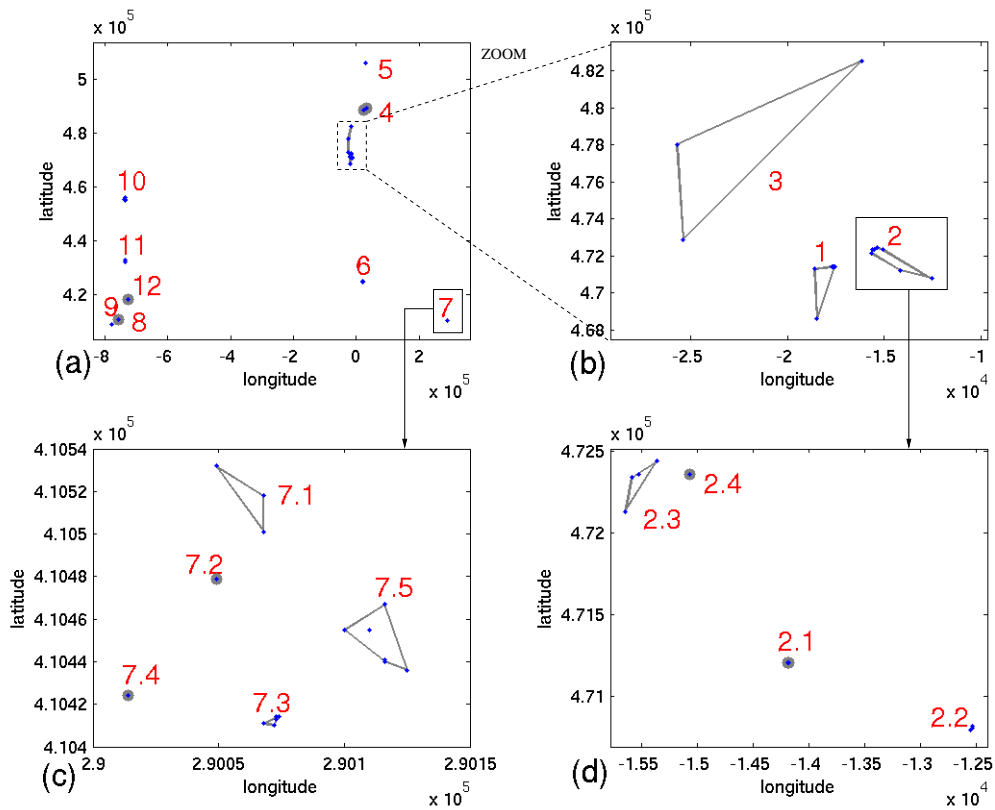


FIG. 8 – Exemple de partitions obtenues dans la classification spatiale hiérarchique : Chaque numéro fait référence à un noeud de l’arbre présenté dans la figure 7. Les points sont les métadonnées spatiales et les polygones englobants représentent l’association des données aux composantes selon le critère maximum a posteriori. Les flèches indiquent les liens de parentés entre les partitions. La figure (a) présente le niveau le plus général de la classification et la figure (b) est un zoom sur cette partition. Les figure (c) et (d) sont respectivement les fils des composantes 7 et 2.

critiques et dépendant ainsi seulement de la structure des données. Le principe est de construire progressivement une hiérarchie de modèle de mélanges en fonction de l'ajout de nouvelles images dans la collection. Le critère ICL maintient une définition uniforme de la qualité des partitions. La technique est aussi relativement robuste face aux classes non-gaussiennes et aux petits échantillons. Notons que le temps de calcul de notre algorithme incrémental est distribué dans le temps (sur plusieurs jours) comme une tâche de fond, consommant ainsi peu de ressources comparé à d'autres activités sur le terminal mobile. Les classifications obtenues sont ensuite dédiées à une interface pour parcourir la collection sur un mobile. Nous étudions en ce moment un moyen de combiner les partitions temporelles et spatiales en une seule partition hybride, afin de simplifier le parcours de la collection sur un appareil mobile.

## Références

- [1] ARIS, A., GEMMEL, J., AND LUEDER, R. Exploiting location and time for photo search and storytelling in MyLifeBits. Tech. Rep. MSR-TR-2004-102, Microsoft research, Septembre 2004.
- [2] BIERNACKI, C., CELEUX, G., AND GOVAERT, G. Assessing a mixture model for clustering with the integrated classification likelihood. In *IEEE Transaction on pattern analysis and machine intelligence* (Juillet 2000), vol. 22, pp. 719–725.
- [3] COOPER, M., FOOTE, J., GIRGENSOHN, A., AND WILCOX, L. Temporal event clustering for digital photo collections. In *Proc. ACM Multimedia* (Novembre 2003), pp. 364–373.
- [4] DEMPSTER, A. P., LAID, N. M., AND RUBIN, D. B. Maximum likelihood for incomplete data via the EM algorithm. *J. R. Stat. Soc.* (1977), 1–38.
- [5] FRALEY, C. Algorithms for model-based Gaussian hierarchical clustering. *SIAM Journal on Scientific Computing* 20, 1 (1999), 270–281.
- [6] GARGI, U., DENG, Y., AND TRETTER, D. R. Managing and searching personal photo collections. Tech. Rep. HPL-2002-67, HP Laboratories, Palo Alto, Mars 2002.
- [7] GELGON, M. Using face detection for browsing personal slow video in a small terminal and worn camera context. In *IEEE Int. Conf. on Image Processing (ICIP'2001)* (Thessaloniki, Greece, Septembre 2001), IEEE Signal Processing society, pp. 1062–1065.
- [8] GELGON, M., AND TILHOU, K. Structuring the personal multimedia collection of a mobile device user based on geolocation. In *IEEE Int. conf. on Multimedia and Expo (ICME'2002)* (Lausanne, Switzerland, Août 2002), pp. 248–252.
- [9] GRAHAM, A., GARCIA-MOLINA, H., PAEPCKE, A., AND WINOGRAD, T. Time as essence for photo browsing through personal digital libraries. In *ACM Joint Conference on Digital Libraries JCDL* (Juin 2002), pp. 326–335.
- [10] LOUI, A., AND SAVAKIS, A. E. Automatic image event segmentation and quality screening for albuming applications. In *IEEE Proceedings Int. Conf. on Multimedia and Expo (ICME'2000)* (New York, USA, Août 2000), pp. 1125–1128.
- [11] LUO, J., SAVAKIS, A., AND SINGHAL, A. A Bayesian network-based framework for semantic image understanding. *Pattern Recognition* 38, 6 (June 2005), 919–934.
- [12] MYKA, A. Nokia lifeblog - towards a truly personal multimedia information system. In *Proc. of Workshop des GI-Arbeitskreises "Mobile Datenbanken und Informationssysteme"* (Karlsruhe, Germany, Février 2005).
- [13] NAAMAN, M., SONG, Y. J., PAEPCKE, A., AND GARCIA-MOLINA, H. Automatic organization for digital photographs with geographic coordinates. In *Proc. of ACM/IEEE Conference on Digital libraries (JCDL'2004)* (Juin 2004), pp. 53–62.
- [14] PIGEAU, A., AND GELGON, M. Incremental statistical geo-temporal structuring of a personal camera phone image collection. In *Proc. of Int. Conf. on Pattern Recognition* (Cambridge, U.K., Août 2004), vol. 3, pp. 878–881.
- [15] PLATT, J. C., AND M. CZERWINSKI, B. A. F. PhotoTOC : Automatic clustering for browsing personal photographs. Tech. Rep. MSR-TR-2002-17, Microsoft Research, Février 2002.
- [16] RODDEN, K. How do people manage their digital photographs ? In *ACM Conference on Human Factors in Computing Systems* (Fort Lauderdale, Avril 2003), pp. 409 – 416.
- [17] UEDA, N., NAKANO, R., GHARHAMANI, Z., AND HINTON, G. SMEM algorithm for mixture models. *Neural computation* 12, 9 (2000), 2109–2128.
- [18] VIOLA, P., AND JONES, M. Robust real-time object detection. In *Proc. of Int. Workshop on Statistical and Computational Theories of Vision - Modeling, Learning, Computing, Sampling (with ICCV'2001)* (Vancouver, Juillet 2001).
- [19] WAGENAAR, W. My memory : a study of autobiographical memory over six years. *Cognitive psychology* 18 (1986), 225–252.
- [20] WILHELM, A., TAKHTEYEV, Y., SARVAS, R., VAN HOUSE, N., AND DAVIS, M. Photo annotation on a camera phone. In *Proc. of ACM Computer Human Interaction (CHI'2004)* (Vienna, Austria, 2004), pp. 234–238.