SnapToTell
Accès ubiquitaire à de l'information multi-média à partir d'un téléphone portable

Application sur une base d'images de Singapour

Jean-Pierre Chevallet — **Joo-Hwee Lim**

Image Processing Lab and Application IPAL-CNRS
Institute for Infocomm Research I2R A-STAR
21 Heng Mui Keng Terrace,
Singapore 119613

viscjp@i2r.a-star.edu.sg
joohwee@i2r.a-star.edu.sg

RÉSUMÉ. *Avec la prolifération des téléphones portables munis d'appareils photo, beaucoup de nouvelles applications et services vont émerger : nous présentons le système SnapToTell, qui permet de fournir de l'information à partir de requêtes images prises d'un téléphone portable. Nous présentons également des résultats expérimentaux sur l'identification de scènes, basés sur une collection test d'images originales et réalistes de scènes à Singapour.*

ABSTRACT. *With the proliferation of camera phones, many novel applications and services will emerge. In this paper, we present the SnapToTell system, which provides information directory service to tourists based on pictures taken by the camera phones and location information. Next we present experimental results on scene recognition based on a realistic data set of scenes and locations in Singapore which form a new original application oriented image test bed freely available.*

MOTS-CLÉS : *Recherche d'Information (RI) par l'image, RI ubiquitaire, utilisation du contexte en RI, Terminaux mobiles de RI.*

KEYWORDS: *Image Information Retrieval (IR), ubiquitous Information Retrieval, using context for IR, mobile device for IR.*

1. Introduction

Imaginez-vous à Singapour, en train de vous promener sur "l'Esplanade" au milieu de sculptures et de monuments historiques. Imaginez-vous en situation de "touriste" en demande d'informations sur ces objets et monuments que vous admirez. Vous pouvez vous renseigner en compulsant vos livres et guides de voyage que vous avez peut être sur vous, mais vous pouvez faire mieux : directement prendre un monument en photo avec votre téléphone portable et l'envoyer à un fournisseur de service par l'intermédiaire du service de messages multimédia (MMS). Peu de temps après, vous recevez un message audio (MMS) et un message texte (SMS) qui vous décrit le monument en question et vous fournit quelques informations supplémentaires. Vous pouvez ainsi continuer à apprécier tranquillement le paysage sans vous surcharger de vos guides ! C'est ce genre de scénario que proposons dans le système *SnapToTell*[1] : un accès à de l'information, depuis n'importe où (ubiquitaire), sur un terminal mobile et à partir d'une requête visuelle.

Selon le dicton *"une image vaut mieux qu'un long discours"*, un touriste peut éviter de rechercher la description du monument ou d'un lieu dans son guide de voyage. Il peut même ignorer le nom des monuments ou le lieu où il se trouve, car il utilise directement l'image pour accéder à de l'information. Nous détournons alors le dicton pour : *"une image pour mieux accéder à un petit discours"*. Le développement de l'informatique mobile est indéniable. Elle va de pair avec le développement des infrastructures de communication sans fils. Cette technologie rend possible de nouveaux services, en particulier pour la Recherche d'Information (RI) mobile, multimédia et ubiquitaire. La particularité de ce type de RI tient d'une part, dans l'utilisation de terminaux de faible puissance, mais aussi dans le rôle prépondérant du contexte de la recherche. Dans le cas de SnapToTell, le contexte concerne au minimum la localisation du lieu de l'appel, réduisant le nombre de "documents" monuments qui sont de possibles réponses à une requête.

D'un point de vue technologique, l'obtention d'une information de localisation est dores et déjà possible avec un système GPS[2]. Elle peut également être obtenue par l'infrastructure téléphonique cellulaire du réseau GSM[3]. Cependant, connaître la position d'un téléphone, et donc de son utilisateur, n'est pas suffisant pour déterminer ce qui l'intéresse. En effet, il peut très bien admirer un monument dans son ensemble et en être éloigné de plusieurs centaines de mètres. L'information de localisation aide certainement à affiner le contexte de la requête, mais n'en capture pas l'intention. Une requête plus explicite est nécessaire, comme une photo prise depuis le téléphone portable. C'est exactement ce que nous proposons dans SnapToTell.

Informer ou guider un piéton dans un environnement urbain n'est pas une idée nouvelle [FEI 97]. Ce qui est nouveau c'est la technologie qui permet enfin la réalisation et l'expérimentation de prototypes crédibles et vraiment portables. Par exemple,

1. "Snap" signifiant ici "prendre un cliché", et "Tell" signifiant "décrire".
2. Global Positioning System http ://www.gpsworld.com
3. http ://www.gsmworld.com/

le système Infoscope [HAR 01] propose un service expérimental de traduction des informations inscrites sur le mobilier urbain à partir d'images prises sur un PDA et envoyées à un serveur de traitement et de traduction. La traduction est directement replacée dans l'image originale.

Nous présentons dans la partie suivante, les systèmes similaires à SnapToTell. Dans la partie 3, nous donnons l'architecture de notre système. Avant la conclusion, la partie 4 est dédiée à une étude expérimentale de l'accès à de l'information sur des objets à travers un ensemble d'images qui leur servent d'index.

2. Les approches similaires

Actuellement le système prototype SnapToTell [LIM 04b, CHE 05] est une application touristique en plein air, mais il est possible d'envisager d'autres types d'applications, comme par exemple, une aide à la localisation en zone urbaine par le repérage de bâtiments typiques [RAM 05]. Dans le secteur de l'éducation, une image partielle d'un végétal (ex : la forme d'une feuille, un champignon, etc.) peut servir de requête pour obtenir une identification de cette plante ou de cet animal. Il faut noter que la difficulté réside essentiellement dans l'algorithme d'appariement entre la requête et le document image.

Le scénario de l'application que nous proposons, est semblable à celui de [MAI 03]. Dans cette étude, le dispositif client utilisé est un PDA (Personnal Digital Assitant) relié à Internet par communication sans fil (WLAN). Cette architecture suppose qu'un point d'accès sans fil est installé dans les environs. Le système utilise un iPAQ 3870, un appareil photo NexiCam, une sonde d'orientation, et un récepteur GPS. La détection de la position est assurée par un GPS relié au PDA. Cependant, la direction et la sonde d'inclinaison est reliée au PDA par l'intermédiaire d'un ordinateur portable, dû à une contrainte technique. Dans notre cas, nous avons choisi un téléphone mobile avec appareil photo intégré car sa diffusion est plus importante (en tout cas pour Singapour) que les PDA. Nous l'avons choisi également pour l'encombrement minimum de cette solution : l'appareil photo est intégré dans le dispositif de communication et la localisation est fournie par l'opérateur de télécommunication.

Dans le prototype de [MAI 03], l'image prise de l'appareil photo, ainsi que les données GPS et d'orientation sont envoyées à un serveur. Le serveur exécute alors un programme de synthèse d'image (3DMax) pour produire une image de référence à la même position en utilisant un modèle 3D construit par avance. L'appariement entre l'image requête et le document est effectué en faisant correspondre l'image réelle avec l'image 3D reconstruite. Cette solution nous parait trop complexe à mettre en oeuvre, en particulier, à cause d'une modélisation précise en trois dimensions des monuments à indexer qu'elle implique.

Dans SnapToTell, notre approche est donc différente : au lieu d'un appariement entre une vraie image et une image synthétisée d'un modèle 3D, nous proposons de pré enregistrer dans une base d'images, un grand nombre de prises de vues réelles des

monuments à indexer. L'objectif est ainsi de couvrir "raisonnablement" les différentes positions possibles de prises de vues, ainsi que les différents éclairements de la scène. Cette notion de nombre "raisonnable" de prises de vues dépend en fait de la qualité de l'algorithme d'appariement et de sa tolérance à la variation. Pour qu'une scène soit sélectionnée comme répondant à la requête, il suffit *qu'une seule de ses prises de vue* apparaisse comme la plus proche de la requête. L'indexation d'une scène (monument, bâtiment, statue, etc.) consiste alors fournir cet ensemble de prise de vues, puis à détecter dans les multiples images, les parties discriminantes des autres scènes pour garantir la précision de l'appariement. Notre stratégie d'indexation de scène suppose donc qu'un ensemble d'images de la scène aient en commun des caractéristiques assez distinctives pour détecter correctement la scène parmi les autres possibles dans un secteur géographique donné. L'information du lieu où la photo a été prise réduit l'espace de recherche et tend donc à contourner le difficile problème de la recherche d'images dans le cas général.

Le système IDeixis [TOL 04] propose également un accès à une information à partir d'une photo prise d'un terminal mobile. A la différence de notre approche, ce système utilise des images prises sur le WEB. Il n'a donc pas de contrôle sur les images. Dans notre approche, la base d'images sert exclusivement à la description des objets, pour servir d'indexation. La question que nous posons incidemment est alors relative au nombre minimum d'images et à leur variété pour obtenir des résultats satisfaisants. De plus, ce système a été assez peu évalué (50 requêtes et 3 lieux).Finalement, les auteurs de ce travail n'ont pas présenté le problème comme un accès à un objet, par l'image la plus proche de la requête, comme nous le proposons ici : ils mesurent la précision à 16 images.

3. Architecture de SnapToTell

SnapToTell est réalisé suivant une architecture client/serveur typique comme représenté dans la figure 1. Le client est un téléphone mobile avec appareil photo intégré. En l'occurrence nous utilisons dans notre développement et nos tests le modèle Nokia 7650. Avec ce téléphone, un utilisateur peut lancer l'application SnapToTell pour envoyer une demande sous forme de MMS au serveur d'application. La demande de MMS est soit l'image de la scène ou cette image prétraitée (ex : un histogramme de couleurs par zone ; cf. partie 4.1).

Lors de la réception du MMS, le serveur d'application obtient également une information de positionnement par l'opérateur de réseau mobile. Il existe plusieurs solutions techniques de localisation. La plus simple et la plus utilisée consiste à noter le numéro d'identification de la cellule de réception (GSM Location Base Service, GSM network cell ID). La précision de la localisation dépend de la taille de la cellule de réception : cela va de 200 m en zone urbaine à 50 km en campagne. Cette information peut être affinée en mesurant le décalage de temps sur le signal radio (Timing Advance) ou par analyse de la puissance de l'émission de l'antenne de la cellule courante. Des techniques plus sophistiquées de positionnement par triangulation par

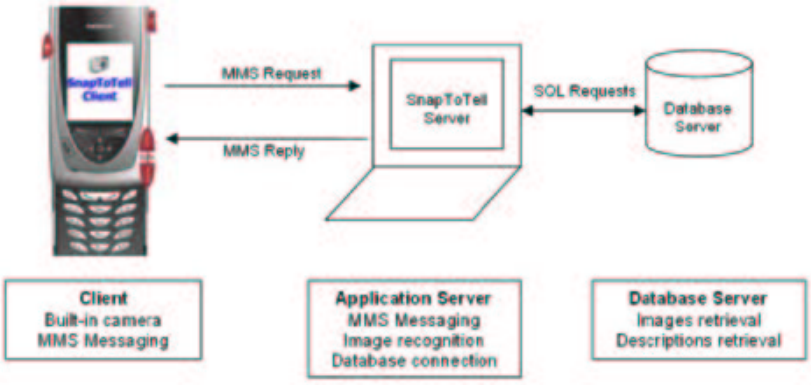

Figure 1. *SnapToTell : une architecture client/serveur*

rapport aux antennes du maillage sont possibles, mais nécessitent une modification du protocole GSM. Elles ne sont pas disponibles pour l'instant. Pour SnapToTell, nous utilisons uniquement le numéro d'identification de la cellule car elle est disponible sur tous les réseaux GSM. Si le téléphone mobile est équipé d'un récepteur GPS, cette information de positionnement peut également être envoyée du client au serveur de SnapToTell. Bien que le GPS soit potentiellement la méthode la plus précise pour obtenir un positionnement, cette méthode a plus de risques d'invisibilité que l'information cellulaire. En effet, pour avoir un signal de positionnement correct, il faut se trouver dans une zone bien dégagée, et donc loin des bâtiments, alors que la réception GSM a très peu de zones d'ombre surtout en zone urbaine où le nombre d'antennes est important. Le GPS a également un coût plus élevé, et nécessite un temps de mise en route (warming up) parfois important (ex : 10 minutes).

Le positionnement de l'appareil étant établi, le serveur de SnapToTell envoie une requête à la base de données pour rechercher l'image la plus proche visuellement. Une valeur de seuil permet de détecter une situation où le monument ne se trouve pas dans la base.

3.1. *Le client : Nokia 7650*

Le téléphone Nokia 7650 est basé sur la plateforme de la série 60 de Nokia (système d'exploitation Symbian OS v6.1). C'est l'un des premiers appareils mobiles à intégrer un téléphone, un appareil photo numérique et des fonctions de PDA. Nokia fournit un émulateur pour chaque série, pour faciliter le développement, en simulant son fonctionnement avant de déployer l'application sur le vrai dispositif. La version actuelle du client SnapToTell est écrite en C++, le langage de développement préconisé par Nokia. Une version Java serait plus portable, mais ce langage ne permet pas actuellement d'accéder à tous les modes de communication (MMS, SMS, Bluetooth).

La figure 2 est une copie des écrans en fonctionnement. Le mode opératoire est assez simple et direct : une photo est prise, puis envoyée, ou bien une photo en mémoire peut être envoyée. Dans ce cas, l'information de localisation doit être fournie manuellement. Le menu "Get Description" déclanche l'analyse locale de l'image et son envoi au serveur.

Figure 2. *Les écrans de SnapToTell côté client sur Nokia 7650*

Dans cette illustration, l'utilisateur a choisi l'image image stockée "SupremeCourt-16.jpg" comme requête (cinquième écran). Cette requête est envoyée comme un MMS au serveur d'application de SnapToTell (décrit ci-après). Le serveur renvoie en réponse, la description du monument, à la fois sous forme de texte et sous forme audio (synthèse vocale de ce texte, statiquement calculée sur le serveur).

3.2. *Le serveur SnapToTell*

Le serveur d'application SnapToTell est le noyau fonctionnel du système. Il est développé en Java. Il est architecturé selon quatre composants comme indiqué dans la figure 3.

Le composant *Communication/Messaging* se charge de la communication avec le dispositif client par MMS. Dans la version actuelle de test, ce composant dialogue avec le téléphone par une interface Bluetooth. Ce protocole de transmission sans fil de courte distance est employé uniquement pour des tests de fonctionnement en laboratoire. Une couche générique de communication encapsule le protocole de communication qui devient indépendant du support physique (MMS, Internet ou Bluetooth). Le composant *database connection* traite les accès à la base de données image par l'intermédiaire de requêtes SQL. Le composant *image matching algorithm* est responsable

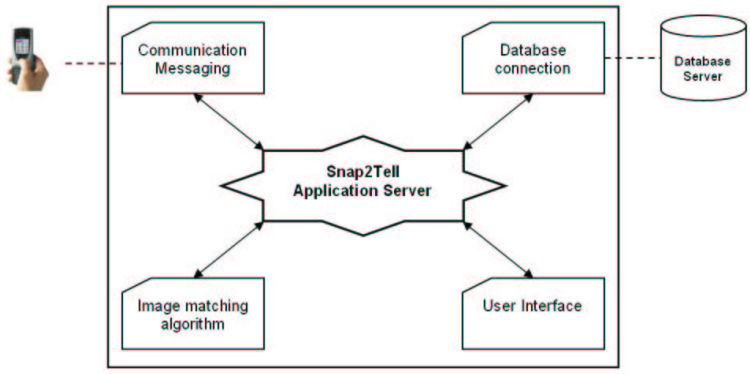

Figure 3. *Composants du serveur de SnapToTell*

de l'appariement entre une image requête et une image de base de données. Il décide du meilleur appariement et du cas où aucune solution n'est trouvée. Enfin, le serveur possède une interface utilisateur (*user interface*) pour configurer le serveur d'application, comme le choix du serveur de base de données, la modification (ajout) d'images, leur ré-indexation, etc. Une fois le serveur d'application configuré, l'administrateur l'active : il passe en mode "écoute".

3.3. *La base de données des scènes*

La base de données des scènes contient la description de tous les monuments, statues, batiments, etc., connus du système avec leur descriptif et l'ensemble de leurs images index associées. Cette base est actuellement gérée par un système Access. Nous utilisons la ville de Singapour comme banc d'essai. Cette base contient trois types d'informations de localisation : d'une part la localisation GPS, les numéros de cellules GSM les plus proches, et d'autre part une localisation hiérarchique à trois niveaux. Nous avons divisé Singapour en zones. Une zone inclut plusieurs lieux (location), dont chacun peut contenir un certain nombre de scènes. Une scène est caractérisée par des images prises de différents points de vue, de distances, et de conditions de luminosité. Par exemple, dans la zone sud de Singapour, nous avons le lieu "Sentosa", où nous pouvons admirer le "Merlion", une sculpture emblématique de Singapour. En plus des descriptions de localisation et des images exemples, une scène est associée à une description textuelle et une description audio. C'est en fait la partie "information" de la base d'image. Finalement, une scène est assignée à une catégorie, information qui pour l'instant n'est pas utilisée dans nos expérimentations (par exemple " Monument", "Art", etc..). Les prises de vues elles mêmes sont associées à des méta-informations : date et heure de la prise de vue, auteur, modèle de l'appareil photo, type de temps, etc. Les prises de vues sont au format JPEG avec les informations de type EXIF disponibles en fonction des modèles d'appareils.

La figure 4 illustre un exemple de relations entre une zone, un lieu et une scène, avec des catégories et des exemples. Pour le lieu No 11 dans la zone 4, nous avons deux scènes : "Chinatown" et "Thian Hock Keng Temple". Trois scènes de nom "Indian National Monument", "Court suprême", et "Rafle Statue" sont situées dans le lieu 14 de la zone 5.

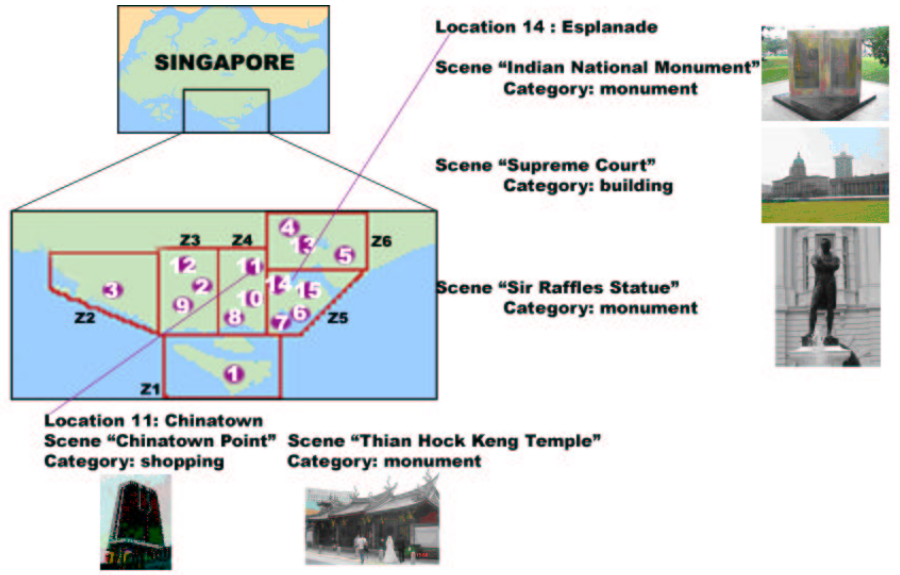

Figure 4. *Zones, lieux et scènes de la base de Singapour*

Notre base de données actuelle possède 6 zones, 15 lieux, 88 scènes, et 1550 images. En moyenne, il y a 2.5 lieux par zone, 5.8 scènes par lieu, et 17 exemples d'images pour décrire une scène. La base totale contient actuellement des prises de vues issues de 8 appareils photos différents : 2 téléphones (Sony Ericsson T610, Nokia 7650), 1 PDA (HP iPAQ 6365), 3 appareils photo bas de gamme (Nikon 3300, Canon G1, Sony Cybershot), 1 de moyenne gamme (Nikon 5700) et 1 reflex haut de gamme (Canon EOS300D) avec 4 auteurs différents. La diversité des modèles d'appareil photos est importante car elle permet de rendre compte des variations des caractéristiques techniques et des dérives colorimétriques.

4. Reconnaissance de scènes

Dans notre approche, la reconnaissance d'une scène passe par la sélection d'une des images associées à cette scène. Nous réduisons de cette manière, la complexité du problème de correspondance d'images en associant une variété de prises de vues de la scène, ce qui augmente les chances de succès d'un appariement avec une requête image. Le problème de Recherche d'Information est alors différent des moteurs

de recherche classiques : il ne s'agit plus de trouver toutes les images pertinentes à une image requête, mais de trouver la scène la plus proche d'une image requête, par l'intermédiaire *d'une seule image* issue de la liste des images de cette scène.

Pour la résolution de la requête nous avons le choix suivant : soit envoyer l'image complète au serveur, soit envoyer une forme pré-analysée de l'image. Cette forme pré-analysée doit être de plus petite taille que l'image initiale. Nous avons opté pour cette seconde solution pour réduire le coût des transferts. Comme le système SnapToTell fonctionne sur un téléphone mobile, nous ne disposons que de peu de puissance de calcul. Il faut donc trouver un équilibre entre le temps de calcul (augmentant la consommation électrique et réduisant l'autonomie) et le coût de transfert des données vers le serveur. Il nous faut alors choisir une technique de calcul de correspondance demandant un traitement minimum de la part de l'ordinateur portable. Pour cette première étude, nous avons adopté un calcul d'histogrammes de couleur [SWA 91] pour caractériser et classer les images. Cette technique est connue pour être invariable à la translation et à la rotation autour de l'axe de la prise de vue. Elle change relativement lentement en fonction de l'angle de la prise de vue, de l'échelle, et de l'occlusion. Par contre elle est sensible au bruit de l'image, mais des méthodes de calculs d'estimations de densité comme décrit dans [GEV 03] limitent la dérive du calcul[4]. Par conséquent nous pensons que cette technique simple est un bon point de départ pour l'identification d'éléments invariables d'une scène, dans le cas de notre application particulière.

Dans la section suivante, nous présentons une étude expérimentale sur un tiers des images de la collection. Notre objectif est d'étudier le comportement du calcul des histogrammes pour sélectionner une seule image d'un groupe d'images d'une même scène.

4.1. *Etude empirique de la base d'images*

Pour valider notre système, nous pouvons expérimenter la recherche de scènes grandeur nature, avec un ensemble d'utilisateurs et sur un ensemble de scènes. Avant cette expérimentation, nous désirons évaluer l'efficacité de notre méthode de sélection de scènes avec les images que nous avons dans notre base. L'objectif de cette étude empirique est donc d'examiner le comportement des images de la base, pour la description des scènes avec la fonction d'appariement implantée dans la version actuelle du système SnapToTell.

En particulier, nous désirons vérifier deux éléments : l'influence du contexte, c'est-à-dire la localisation sur la précision de la recherche, et le centre d'intérêt des images. Pour les tests, nous avons utilisé 530 images de la base actuelle. Nous utilisons actuellement une mesure de correspondance basée sur des histogrammes de couleurs globaux à l'image et locaux par zones. La technique par histogramme de couleurs possède des avantages intéressants dans notre contexte : ils sont d'une part très simples

4. Nous n'avons pas encore testé cette méthode

à calculer et demandent donc peu de puissance de calcul. Par exemple, dans la version actuelle du client de SnapToTell, le client Nokia ne possède pas d'unité de calcul en virgule flottante. L'histogramme est tout de même calculé sur le téléphone, mais sans la division de normalisation qui est calculée sur le serveur. L'inconvénient des histogrammes est le manque d'informations spatiales. Ces informations spatiales sont parfois importantes pour identifier plus clairement les objets. Nous utilisons alors un calcul d'histogrammes par blocs avec une prépondérance pour le centre de l'image. En effet, nous faisons l'hypothèse que les images ont pour unique objectif de désigner une entité dans la scène (ex : un monument). Le centre de l'attention doit donc être situé au centre de l'image.

Nous calculons donc une mesure de similitude λ entre une requête q (avec m blocs locaux Z_j) et une image x (avec m blocs locaux X_j) de la manière suivante :

$$\lambda(q,x) = \frac{\sum_j \omega_j \cdot \lambda(Z_j, X_j)}{\sum_k \omega_k}, \qquad [1]$$

où ω_j sont des poids des blocs, et $\lambda(Z_j, X_j)$ est une mesure de similarité entre deux blocs d'images définie par :

$$\lambda(Z_j, X_j) = 1 - \frac{1}{2}\sum_i |H_i(Z_j) - H_i(X_j)|, \qquad [2]$$

avec $H_i(Y)$ qui représente la valeur de la classe i de l'histogramme de l'image ou la zone d'image Y. Les histogrammes sont normalisés, cela signifie que pour tout histogramme calculé sur Y, $\sum_i H_i(Y) = 1$. On peut noter que cette similarité correspond à une intersection d'histogrammes [SWA 91]. Les histogrammes sont calculés dans l'espace HSV [SCH 87] qui correspond à la couleur (Hue, la teinte), la saturation, et la valeur (l'intensité lumineuse). Cet espace est intéressant car il est plus proche de l'interprétation que l'on peut avoir des couleurs. Cependant, comme cet espace est juste une transformation réversible de l'espace RVB (intensité de Rouge, de Vert et de Bleu), on ne s'attend pas à des améliorations particulières par rapport à des histogrammes directement calculés sur ces composantes. Par contre, il est possible de pondérer les composantes HSV selon que l'on considère que la teinte par exemple, est plus importante que la saturation. Il faut noter cependant que la teinte est instable au voisinage de faibles saturations, ou bien dans le cas de faibles luminosités. Nous ne corrigeons pas ce problème actuellement.

Un histogramme est calculé globalement sur les 3 dimensions du codage des couleurs. Cela signifie que pour un ensemble de classes (bins) C_H, C_S, C_V de l'espace HSV, nous avons le nombre total $C_H \times C_S \times C_V$ de classes à calculer. Dans nos expérimentations nous avons conservé la même résolution de classes b pour les 3 dimensions : $C_H = C_S = C_V = b$. Par conséquent, le nombre de classes du calcul des histogrammes est égal à b^3 où b est le nombre d'intervalles égaux dans chacune des dimensions de H, S, et de V.

Nous avons également divisé l'image en blocs identiques dans les deux dimensions de X-Y (cad. une grille de $K \times K$). Lorsque deux images sont comparées, nous comparons les histogrammes locaux des blocs correspondants (équations (2) et (1)). Nous avons examiné un maximum de $K = 8$ blocs dans chaque dimension. Les images sont donc découpées en un maximum de 64 blocs. Dans ce cas extrême, nous devons donc calculer 64 histogrammes. Notez que $K = 1$ correspond à un histogramme global unique sur toute l'image.

Dans cette étude empirique, nous désirons étudier l'effet de la connaissance d'un contexte simple : la localisation du téléphone dans une zone ou un lieu[5]. L'objectif est simple : réduire l'espace de recherche des images à apparier avec la requête. Nous étudions en fait ici, le *comportement global de la base d'images*, en considérant chaque image comme une requête potentielle. Nous avons volontairement laissé de côté, pour cette expérience, le fait que les images soient prises par différents appareils photos, pour ne garder qu'un groupe cohérent d'images par le même appareil photo.

Le problème de la variation entre appareil photo est réel et provoque une dérive des couleurs donc perturbe la correspondance. Ce problème important doit être traité séparément. La solution simple que nous avons mise en place actuellement, est de conserver la cohérence entre l'appareil qui est utilisé pour la requête, et celui qui est utilisé pour la prise de vue de l'indexation. Bien évidemment cette solution n'est pas optimale, car l'effort de constitution de l'index est important : il faut se rendre sur place et prendre une grande variété de photos d'un même monument sous plusieurs angles et sous différentes conditions de luminosité. Il n'est pas souhaitable de multiplier cet effort par le nombre de modèles d'appareils photos. Pour cette expérimentation nous n'avons donc utilisé que les 530 images de l'appareil Sony. Chaque image est prise comme une requête et est comparée à l'ensemble des 529 autres images. Les images sont ensuite classées en ordre décroissant de similitude.

La table 1, présente la précision obtenue par la recherche d'une scène par la sélection de la scène de l'image la plus proche de la requête. Dans ces résultats, nous constatons qu'une augmentation du nombre de classes de l'histogramme, augmente la précision des résultats. Un palier apparaît à partir de 9 classes dans les 3 dimensions HSV. Au delà de 11 classes, nous voyons apparaître[6] une baisse de la précision qui s'explique par le fait qu'une plus grande précision peut causer des décalages dans le décompte des pixels dans les classes adjacentes, et augmente donc les disparités des histogrammes. Sans employer d'information de localisation, la meilleure précision est obtenue globalement avec 11^3 classes (c'est à dire 1331 classes !). Le meilleur score de précision (73,4%) est obtenu pour 11 classes avec 4 et 9 blocs par images.

La table 2 montre la précision obtenue en utilisant l'information de localisation par zone. Nous choisissons la meilleure image à partir de l'ensemble des images qui partagent la même zone que l'image requête. Clairement, nous notons une augmenta-

5. nous n'utilisons pas l'information de la cellule GSM car la base actuelle est incomplète sur cette information
6. Assez peu nettement toutefois.

Tableau 1. *Précision d'une scène par l'image la plus proche*

	Nombre de blocs							
Classes	1×1	2×2	3×3	4×4	5×5	6×6	7×7	8×8
2	26.3	42.4	51.5	51	51.2	54.7	54	54.5
3	47.8	56.8	60.9	62.4	62.2	61.1	61.4	62.6
4	59.8	64.2	66.3	68.7	69.3	68.7	69.5	68.5
5	64.4	66.1	69.1	68.7	69.1	67.8	67.2	66.9
6	66.7	68.5	71.9	71.7	71.5	70.4	69.5	68.9
7	66.9	71.2	71.7	71.9	70.8	71.2	70.0	69.9
8	66.5	71.4	72.8	72.7	**73.0**	72.1	70.2	**70.6**
9	69.7	71.2	72.3	70.6	72.3	71.2	70.0	69.7
10	69.9	72.3	71.5	71.4	71.9	70.8	70.0	69.3
11	**71.9**	**73.4**	**73.4**	**73.0**	72.5	71.7	**71.0**	70.2
12	71.4	71.7	73.2	72.1	71.5	69.9	70.0	69.5
13	71.5	72.8	72.5	72.7	71.2	71.7	70.4	70.0
14	71.2	72.3	72.5	72.7	72.5	**72.3**	70.8	70.0

Tableau 2. *Précision de scènes avec filtrage par zone*

	Nombre de blocs							
Classes	1×1	2×2	3×3	4×4	5×5	6×6	7×7	8×8
2	44.4	59.4	62.8	63.3	65.0	66.9	64.2	65.2
3	62.8	67.1	69.1	73.2	71.7	71.0	70.8	71.0
4	68.2	73.6	74.7	76.4	76.0	77.0	76.4	76.0
5	72.5	73.2	76.2	76.0	75.8	75.3	74.9	75.1
6	72.5	75.1	76.8	76.8	77.0	76.2	76.4	76.8
7	73.2	77.7	77.1	78.5	76.4	76.2	76.0	75.8
8	73.0	77.5	78.8	**79.0**	78.6	77.5	77.5	77.7
9	75.3	76.8	**79.2**	77.9	78.6	78.6	77.5	77.3
10	75.7	77.3	78.1	77.7	78.3	76.6	76.2	76.0
11	76.8	**78.5**	79.0	78.8	78.3	78.8	**78.6**	**78.5**
12	76.0	77.7	79.0	78.3	78.1	76.4	77.0	76.4
13	77.0	78.3	79.0	78.5	78.5	77.7	77.7	77.9
14	**77.3**	78.3	78.8	**79.0**	**79.6**	**79.0**	77.9	77.5

tion prévisible de la précision. Nous notons également que les meilleurs résultats se déplacent vers des histogrammes plus détaillés. Ce phénomène est peut être un artefact statistique dû à la réduction du nombre d'images pris en compte à cause du filtrage. Le nombre de blocs optimum se situe entre 3×3 et 6×6. La classe optimum apparaît par contre moins nettement.

Lorsque l'on augmente la précision de la localisation (cf : table 3), nous notons encore une amélioration des résultats. La distribution des meilleurs résultats est assez

Tableau 3. *Précision de la scène en utilisant la localisation*

	Nombre de blocs							
Classes	1×1	2×2	3×3	4×4	5×5	6×6	7×7	8×8
2	54.7	65.7	68.0	68.9	70.2	72.3	69.9	71.2
3	68.9	71.7	74.7	77.7	76.2	75.5	76.2	76.2
4	72.3	76.6	78.6	80.1	80.0	81.1	79.6	79.6
5	77.1	77.7	79.8	79.6	79.6	78.8	78.6	79.0
6	76.8	78.8	80.7	80.7	80.3	79.8	80.3	80.1
7	77.1	81.1	81.8	82.2	80.5	80.7	79.8	79.8
8	77.5	80.3	**82.4**	82.4	**82.2**	81.4	80.7	80.9
9	78.6	80.3	**82.4**	81.6	82.0	81.4	80.5	80.3
10	78.8	79.8	81.3	80.9	81.3	80.0	79.6	79.6
11	**80.3**	**81.3**	81.4	82.2	81.3	**82.0**	**81.6**	**81.4**
12	79.0	80.5	81.3	81.4	81.4	80.3	79.8	80.0
13	**80.3**	81.1	82.0	81.4	81.4	80.7	80.7	**81.4**
14	80.1	81.1	81.8	**82.6**	**82.2**	**82.0**	81.4	80.7

Tableau 4. *Précision au taux de rappel maximum sans localisation*

	Nombre de blocs							
Classes	1×1	2×2	3×3	4×4	5×5	6×6	7×7	8×8
2	8.9	12.2	14.0	13.8	14.0	14.3	13.9	13.7
3	15.8	17.3	18.5	18.7	18.7	18.3	17.9	17.5
4	18.5	19.7	20.9	20.5	21.0	20.2	19.9	20.2
5	20.4	22.4	22.4	22.1	22.3	21.3	21.0	20.7
6	22.4	23.2	23.3	24.1	23.8	23.0	22.6	22.5
7	22.5	**23.7**	24.2	23.9	24.0	23.0	23.2	23.0
8	21.7	22.7	24.1	24.2	**24.6**	**23.9**	23.6	**23.5**
9	**22.7**	23.6	**24.4**	24.3	24.1	23.8	**23.7**	**23.5**
10	22.5	23.3	24.3	24.1	24.0	23.4	23.1	22.9
11	22.5	23.6	24.2	**24.7**	24.2	23.5	23.1	23.1
12	21.7	22.0	22.8	22.8	23.0	22.2	22.3	22.0
13	22.3	22.5	23.7	23.5	23.7	23.1	22.7	22.6
14	22.3	22.4	23.7	23.3	23.3	22.6	22.6	22.3

semblable à la table précédente. Une réduction de l'espace de recherche ne semble donc pas trop affecter les caractéristiques du calcul des histogrammes.

La table 4 donne la précision à 100% de rappel. C'est en fait le rapport des images dans la scène correcte, sur le total d'images recherchées quand toutes les images de cette scène sont recherchées. On constate d'une part une baisse importante des résultats : cela s'explique par le fait que l'on recherche dans ce cas toutes les images pertinentes, c'est-à-dire toutes les images d'une scène, à partir de chaque image de la

Tableau 5. *Précision avec une comparaison pondérée centrale (sans localisation)*

	Nombre de blocs									
	Poids 1 à 2					Poids 1 à 5				
Cls	3×3	4×4	5×5	6×6	7×7	3×3	4×4	5×5	6×6	7×7
2	51.2	50.8	52.1	54.9	53.0	49.9	47.6	53.0	53.4	51.4
3	62.8	63.3	64.8	63.1	62.8	62.4	62.6	66.3	64.8	63.5
4	68.9	69.1	70.8	69.3	69.9	69.7	67.6	69.7	68.0	69.7
5	70.0	70.4	71.2	70.0	70.0	69.9	70.8	71.4	71.0	71.2
6	73.0	73.0	71.7	72.3	70.6	71.5	72.3	73.8	73.2	72.5
7	72.1	71.9	73.2	72.1	71.7	71.4	71.2	72.3	71.2	**72.8**
8	73.6	74.2	73.0	72.7	**72.3**	73.6	72.7	73.4	72.3	72.1
9	73.8	73.0	73.6	**73.4**	**72.3**	73.0	73.6	73.2	72.1	71.5
10	73.4	72.5	**73.8**	72.8	71.2	74.2	**73.8**	73.8	**72.5**	71.4
11	**74.9**	73.8	73.0	72.7	71.0	74.2	72.5	73.0	**72.5**	71.0
12	74.5	**74.3**	72.8	71.7	71.2	73.2	72.7	73.0	71.5	72.5
13	74.0	73.6	73.4	71.9	70.4	**74.7**	**73.8**	**74.0**	72.1	70.4
14	74.0	**74.3**	73.6	71.7	71.4	73.8	73.4	73.2	71.4	71.0

collection. La précision est globalement assez stable en dessous de 24%, mais également très faible. Cela signifie que quelle que soit la manière de calculer les histogrammes, en moyenne sur un ensemble de réponses, seul un cinquième des images correspondent à la scène correcte. En fait, la distribution des résultats dépend des images de la base : une grande diversité des images index permet d'assurer qu'au moins une est nettement proche de l'image requête. Par contre, plus il y a de diversité, plus il devient difficile de retrouver avec précision toutes les images de la scène. On s'attend donc à ce que la précision par scène augmente lorsque la précision au taux de rappel maximum diminue. On peut finalement affirmer que cette dernière mesure n'est pas pertinente pour notre application, car notre but n'est pas de trouver toutes les images pertinentes, mais d'assurer de trouver au moins une image qui donne accès à une scène qui elle est pertinente.

Finalement la table 5 détaille les résultats de la précision d'une comparaison pondérée par zone. Nous avons pondéré les blocs en privilégiant les blocs du centre. La répartition des pondérations est linéaire avec le minimum sur les bords et le maximum sur le ou les blocs du centre : selon que le nombre de blocs est respectivement impair ou pair. Nous avons testé deux répartitions : du simple au double (poids de 1 à 2), et une valeur cinq fois plus importante au centre. Les calculs sont réalisés sans tenir compte de la localisation. Il faut donc comparer ces résultats avec ceux de la table 1. Nous pouvons noter une augmentation systématique de la précision, mais somme toute dans un rapport assez faible : un point de pourcentage en moyenne. Nous constatons donc que notre hypothèse du rôle prépondérant du centre de l'image est valide, mais que le gain est assez peu intéressant. On note également que ce gain est toujours supérieur dans le cas d'un découpage impair de blocs. En effet, dans cette configuration l'unique bloc à l'extrême centre est nettement plus avantagé que les blocs sur les bords de l'image. Cette remarque nous met sur la voie d'un découpage de blocs peut être plus en rapport avec le contenu effectif de l'image, et donc sur une étude des éléments locaux discriminants d'un groupe d'image cohérents, c'est-à-dire d'une même

scène. Nous travaillons actuellement dans cette direction [LIM 04a]pour améliorer la technique classique de la correspondance par histogramme qui arrive rapidement à son maximum d'efficacité.

5. Conclusion

Dans cet article, nous avons présenté le système expérimental SnapToTell : une application de Recherche d'Informations multimédia ubiquitaire avec une application touristique sur un téléphone portable muni d'un appareil photo. Notre approche traite le problème en vraie grandeur et avec un véritable dispositif mobile léger d'accès à l'information. D'un point de vue purement technologique, nous sommes à la limite à la fois dans l'évolution du logiciel et du matériel, de ce que l'on peut réaliser avec un téléphone portable disponible actuellement. Cependant, nous sommes convaincus que l'informatique mobile va continuer son fort développement à court terme et que les contraintes techniques auxquelles nous sommes confrontés, seront peu à peu levées. Nous sommes convaincus que l'accès mobile à de l'information, en particulier l'accès contextuel à l'information basée sur l'image, est un thème de recherche prometteur pour la Recherche d'Information, qui avec ses contraintes originales, amène à une réflexion particulière et à des solutions innovantes.

Les résultats de l'analyse présentée ici sur une partie de la base d'images, prouvent que la technique de multiplication des prises de vue pour "indexer" une scène permet de contourner très simplement le problème toujours difficile de la reconnaissance d'entités dans l'image. D'autre part, nous pensons que la recherche d'information sur les images doit s'adapter à chaque type d'application, pour trouver un équilibre crédible et efficace entre la complexité des calculs d'indexation et d'appariement et le niveau de précision des résultats obtenu. Notre point de vue est que la sémantique de l'information se trouve et restera du coté de l'utilisateur humain : ce que l'ordinateur peut nous apporter en Recherche d'Information doit se trouver parmi ses capacités propres et originales : le calcul en masse. C'est pour cette raison que nous ne cherchons ni à "extraire de la sémantique" des images prises, ni à reconnaître explicitement les éléments de la scène (personnages, objets, etc.), pour effectuer la correspondance. Au contraire, nous avons privilégié l'utilisation d'éléments visuels de bas niveau : couleur, saturation et intensité et leur répartition par histogramme, pour proposer une solution calculatoire simple mais effective.

Il reste bien sûr à affiner notre approche : en particulier, l'adaptation à différents types d'appareils photos, pour limiter le nombre de prises de vue à l'indexation. Nous avons pu constater en effet, une dérive des couleurs pour les images réalisées avec des appareils bas de gamme, ce qui perturbe la correspondance d'images par histogramme. Nous travaillons alors, d'une part à une adaptation par calibration automatique des différents appareils photos, et d'autre part, à la détection simple d'éléments caractéristiques des groupes d'images décrivant une scène, pour augmenter la précision de la correspondance. La base d'image est disponible librement à des fins de recherche à l'adresse `http://ipal.imag.fr/SnapToTell`.

6. Bibliographie

[CHE 05] CHEVALLET J.-P., LIM J.-H., VASUDHA R., « SnapToTell : a Singapore Image Test Bed for Ubiquitous Information Access from Camera », Poster on European Conference on Information Retrieval ECIR'05, 2005.

[FEI 97] FEINER S., MACINTYRE B., HÖLLERER T., WEBSTER A., « A touring machine : Prototyping 3D mobile augmented reality systems for exploring the urban environment », *Proceedings ISWC '97 (First IEEE Int. Symp. on Wearable Computers)*, vol. 1(4) de *Personal Technologies*, October 13-14 1997, p. 208-217.

[GEV 03] GEVERS T., STOKMAN H., « Robust Histogram Construction for Color Invariants for Object Recognition », *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 25, nº 10, 2003.

[HAR 01] HARITAOGLU I., « InfoScope : Link from Real World to Digital Information Space », ABOWD G., BRUMITT B., SHAFER S., Eds., *Ubicomp 2001 : Ubiquitous Computing : Third International Conference Atlanta, Georgia, USA,*, vol. 2201/2001 de *Lecture Notes in Computer Science*, Springer-Verlag Heidelberg, September 30 - October 2 2001, p. 247-255.

[LIM 04a] LIM J., JIN J., « Semantics discovery for image indexing », (EDS.) T. P. . J. M., Ed., *Proc. of European Conference on Computer Vision*, Springer-Verlag, Germany, LNCS 3021, May 2004, p. 270–281.

[LIM 04b] LIM J.-H., CHEVALLET J.-P., MERAH S. N., « SnapToTell : Ubiquitous Information Access from Camera. A Picture-Driven Tourist Information Directory Service », *Mobile and Ubiquitous Information Access (MUIA'04) Workshop as part of the conference Mobile Human Computer Interaction with Mobile Devices and Services (Mobile HCI 04), Glasgow University of Strathclyde, Scotland*, 13 September 2004, p. 21–27.

[MAI 03] MAI W., DODDS G., TWEED C., « A PDA-Based System for Recognizing Buildings from User-Supplied Images », *Mobile and Ubiquitous Information Access : Mobile HCI 2003 International Workshop*, vol. 2954 / 2004, Lecture Notes in Computer Science, Springer-Verlag Heidelberg, Sept 2003, p. 143–157.

[RAM 05] RAMNATH V., LIM J.-H., CHEVALLET J.-P., ZHANG D., « Harnessing Location-Context for Content-based Services in Vehicular Systems », *IEEE 61st Semiannual Vehicular Technology Conference VTC2005-Spring, Stockholm, Sweden*, 2005.

[SCH 87] SCHWARTZ M. W., COWAN W. B., BEATTY J. C., « An Experimental Comparaison of RGB, YIQ, L*a*b*, HSV, and Opponent Color Models », *ACM Transactions on Graphics*, vol. 6, avril 1987, p. 123–158.

[SWA 91] SWAIN M., BALLARD D., « Color indexing », *Intl. J. Computer Vision*, vol. 7(1), 1991.

[TOL 04] TOLLMAR K., YEH T., DARRELL T., « IDeixis - Searching the Web with Mobile Images for Location-Based Information », BREWSTER S., DUNLOP M., Eds., *Mobile Human-Computer Interaction - MobileHCI 2004 : 6th International Symposium*, vol. 3160/2004 de *Lecture Notes in Computer Science*, Springer-Verlag Heidelberg, September 13 - 16 2004, p. 288 - 299.