

# 高速特定物体認識手法を用いたモバイルデバイスでのリアルタイム AR

吉 田 悠 一<sup>†</sup>   安 倍 満 <sup>†</sup>

モバイルデバイス上でリアルタイムで動作する“Augmented Reality”を実現する、高速でメモリ効率の良い特定物体認識手法を提案する。特定物体認識とは、入力画像に映る対象と、同一の物体が映っている画像をデータベース内から探し出す処理のことであり、中でもデータベース中の参照画像と、入力画像との対応点を求めることで画像検索を行う手法がよく知られる。この対応点の情報から、参照画像に対する入力画像の相対位置関係が計算できるため、この手法はマーカーレス AR に広く応用されている。しかしながら、計算能力の低いモバイルデバイス上で実現するためには、次の2つの課題が残っている。一つ目の課題は、対応点の探索のために必要な局所特徴量の計算コストが大きい点である。二つ目の課題は、特定物体認識に要するデータベースのサイズや検索時間が大きくなり過ぎる点である。提案手法は、これらの2つの課題を解決する。提案手法を実装し、実行速度、メモリ効率について評価実験を行った。結果、モバイルデバイス上で実装した提案手法は、サーバ側で物体認識を行うタイプの既存のクライアント/サーバ型 AR システムと同等の性能を示した。

## Realtime mobile AR with fast object recognition framework

YUICHI YOSHIDA<sup>†</sup> and MITSURU AMBAI<sup>†</sup>

We propose a new fast object recognition method for mobile and realtime AR, which is highly efficient in computational cost and memory usage. Image-based object recognition is a problem of finding an image that contains the same object appeared in an input image, from image database. One of the well known methods of object recognition uses point correspondence between input and reference images. Recently, such method has been applied to markerless AR because the corresponding points could be also used to calculate relative position of the object between input and reference images. But, one has to overcome two problems for implementing such AR system on mobile devices. First, computational cost is too large for mobile devices to calculate local descriptors such as SIFT, which are used to find corresponding points. Second, a database is too large and searching the right object takes too much time. Our proposed method resolves both problems so as to make AR with image-based object recognition feasible. We evaluate execution time and efficiency of memory usage about our method. As a result, our method implemented on a mobile device is shown to have the same level of performance computing with the existing client/server AR system.

### 1. はじめに

近年、iPhone や Android 端末の登場により、カメラを備え、従来より計算能力が高いモバイルデバイスが普及してきた。このため、対象の物体を端末のカメラ越しに覗き込むことで、その解説が重畳表示されるような AR アプリケーションの実現性が高まってきた。

多くの AR 技術が提案されているが、中でも画像処理に基づくマーカーレス AR 技術は適用範囲が広く、AR アプリケーションを実現する上で有効な方向性であるといえる。特に近年では、特定物体認識の技術に応用することにより、マーカーレス AR を高精度で実

現する手法が提案されるようになった。

特定物体認識はマーカーレス AR を実現する上で非常に重要な技術といえるが、メモリ使用量や計算コストが非常に大きい。このため、計算資源の限られるモバイルデバイス上で AR アプリケーションを動作させるためには、より省メモリ、低計算コスト、低消費電力であることを特徴とする実現手法が重要となると考えられる。

そこで本研究では、モバイルデバイス上で AR を実現するための、高速でメモリ効率の良い特定物体認識手法を提案する。提案手法では、特定物体認識に必要な局所特徴量の計算コストを大幅に削減することで、高速化を実現した。また、局所特徴量のデータを単精度実数のベクトルからデータ量の小さいバイナリコード列に変換することで、特徴量同士の比較における計

<sup>†</sup> 株式会社デンソーアイティラボラトリ  
DENSO IT Laboratory, Inc.

算コストを削減するとともに、データベースサイズの削減を実現した。本文では、実行速度、メモリ効率、画像検索精度の3点についての評価結果を述べる。

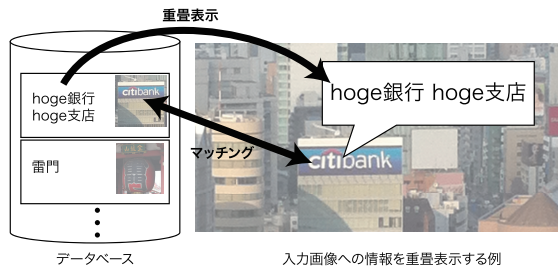


図 1 画像認識によるマーカーレス AR システム

Fig. 1 Markerless AR system based on computer vision



図 2 従来の AR システム

Fig. 2 Well-known AR systems

## 2. 関連研究

カメラで撮影した画像に情報を重畳するためには、何らかの方法で対象を識別し、その対象の画像中の位置を特定しなければならない。本節では、その既存手法と本研究の位置づけについて述べる。

### 2.1 既存手法

画像を用いない単純な実現方法としては、GPS と電子コンパスを併用する手法が挙げられる (図 2-(a))<sup>2)</sup>。この手法は、あらかじめデータベース上に対象物体の位置を記憶しておき、ユーザの位置とカメラの向きに基づいて、現在のカメラの画角内にある対象物の位置を推定する。センサーを搭載していれば実装は比較的容易であるが、ユーザの位置とカメラの向きの推定精度は、GPS と電子コンパスの精度に依存するため、例えば図 1 のように、撮影対象に正確に情報を重畳することができないという欠点がある。

画像処理を用いた成功例として、2次元コードを用いた手法が挙げられる (図 2-(b))<sup>3)</sup>。この手法では、2次元コードが印刷されている平面に対する自分のカメラの姿勢を、高速かつ高精度に推定することが可能である。また、2次元コードは大量のコードパターンを容易に作成できるため、多様な対象を識別できる。し

かしながら、情報を重畳したい対象、もしくはその付近に2次元コードを付与する必要があるため、適用範囲が限られる。例えば、屋外のランドマークや大きな看板等に情報を提示するような AR アプリケーションには不向きである。

さらに、画像処理を用いた手法の中には、前述の2次元コードを使わず、撮影対象を画像を用いて識別し、AR を実現する手法が提案されている。Taketa らは、テンプレートマッチングで対象を識別し、その後、特徴点をトラッキングすることで、絵本の上に3次元仮想物体を重畳表示する手法を提案した<sup>4)</sup>。この手法は、対象物に2次元コードを付与する必要性がなく応用範囲が広い。

### 2.2 特定物体認識に基づく AR

マーカーレス AR の究極的な目標を、『カメラをかざすだけで、対象の情報が取得できる』こととするならば、より多くの対象を識別できる対象識別能力と、画像中の対象の位置を同定する能力が必要となる。そこで近年では、局所特徴量を用いた特定物体認識手法に基づく AR システムが提案されている。特定物体認識とは、データベース中の参照画像と、入力画像に共通して含まれる対応点を求めることで画像検索を行う手法であり、大規模な画像検索に成功した事例が多く報告されている<sup>15)</sup>。また、対応点の情報から、参照画像に対する入力画像の相対位置関係が計算できるため、マーカーレス AR に広く応用されている。

Lowe により提案された SIFT 特徴量は画像の局所特徴量として最も成功した事例であり、あらかじめ登録された複数の対象物を、高精度で検出および識別することに成功している<sup>8)</sup>。多くの特定物体認識は、この SIFT や、SIFT を高速化した SURF 特徴量<sup>7)</sup>に基づいている。Chen らは、モバイル端末で撮影した画像から SURF 特徴量を抽出し、ネットワークを通じてサーバ側に特徴量を送信することで、物体認識をする AR アプリケーションを構築した<sup>9)</sup>。この研究では、20000 種類の本と CD を識別する AR アプリケーションを実現している。このように、SIFT、SURF などの局所特徴量を用いた物体認識手法は、色ヒストグラム等を利用するテンプレートマッチングと比較し、画像の特徴を記述する能力が高く、多くの対象を識別するのに有効である。

しかしながら、局所特徴量をベースにしたマーカーレス AR をモバイルデバイス上で実現するには以下の3つの問題点がある。

#### (1) 局所特徴量抽出の計算コストが高い

SIFT は計算コストが高く、モバイルデバイス

上での実装に不向きである。SURF は SIFT よりも高速であるが、それでもなお AR のためには不十分である。

## (2) 局所特徴量のサイズが大きい

局所特徴量は、単精度実数で表現された数百次元のベクトルで表現される。一枚の画像から数百～数千の局所特徴量が抽出されるため、そのデータ量は数十キロ～数メガバイトになる。モバイルデバイス上での主記憶領域にデータベースを保持することは非常に困難である。

## (3) 局所特徴量のマッチングの計算コストが大きい

局所特徴量を用いた対応点探索は、局所特徴量同士の  $L_2$  距離を計算する。この高次元のベクトル同士の距離計算は計算負荷が極めて高い。k-d tree などの近似最近傍探索アルゴリズムを用いて高速化することもできるが、それでもなお、その計算コストは大きい。

### 2.3 本研究の位置づけ

そこで本論文では、モバイル端末上でも高速に計算できる FAST GLOH 特徴量と、局所特徴量のデータサイズを削減するバイナリコード化処理を備えた、AR のための特定物体認識手法を提案する。提案手法は計算負荷が低くメモリ効率が良いため、モバイル端末単体で動作する。従って、Chen らのシステムのように、サーバーサイドの識別処理が不要である。表 1 に関連研究との比較をまとめる。

表 1 既存手法と提案手法の比較  
Table 1 Comparison with related works

手法	重畳の精度	識別数	計算量/メモリ量	マーカー
GPS	×	○	○	○(不要)
2 次元コード	○	○	○	×(要)
テンプレート	○	△	△	○
局所特徴量	○	○	×	○
提案手法	○	○	○	○

本論文の特徴をまとめる。

## (1) FAST GLOH 特徴量

方向ヒストグラムに基づく局所特徴量抽出において、ボトルネックであった特徴点周辺のパッチの回転演算および微分の演算を大幅に削減することにより、高速化を実現した。

## (2) Sparse projection によるバイナリコード化

単精度実数で表現された数百次元のベクトルからなる局所特徴量を 64～128bit 程度のバイナリコードに変換し、局所特徴量のデータサイズを圧縮することで、モバイルデバイスのような

小さなメモリやストレージ上でも動作可能とした。また、バイナリコード化に必要な変換行列を疎行列とすることで、高速なバイナリコード化を実現した。

### 3. 提案手法

最初に、提案手法の処理の概要を説明し (図 3)、次に本手法を具体的に処理の段階に分けて説明する。

#### 3.1 概要

まず始めに、入力画像から局所特徴量を抽出する。スケール不変な局所特徴量を得るために入力画像を  $1/\sqrt{2}$  ずつ縮小し、4 層のピラミッド画像を構築する。そして、各層の画像に対して特徴点抽出手法<sup>10)</sup> を適用する。各層の特徴点周辺 (半径 25 ピクセル) 領域について回転に不変な FAST GLOH 特徴量を計算し、Sparse Projection によってバイナリコードに変換する。

次に、入力画像に含まれるバイナリコードに最も近いバイナリコードをデータベースの中から検索し、対応付ける。バイナリコード間の距離としては、ハミング距離を用いる。データベースには、あらかじめ識別対象を撮影した参照画像から抽出したバイナリコードを登録しておく。データベース中において、最も多く類似した対応点を持つ画像を検索結果とする。最後に、この対応点の情報から、検索結果画像に対する入力画像の相対位置関係を記述するホモグラフィ行列を計算する。ホモグラフィ行列に基づき、入力画像に対して情報の重畳表示を行う。

#### 3.2 FAST GLOH 特徴量

FAST GLOH 特徴量は、GLOH (Gradient location-orientation histogram) 特徴量<sup>12)</sup> を元にした高速に計算可能な局所特徴量である。局所特徴量の中でも、GLOH は patch を円形にし、さらに subpatch の形状を扇型にすることによって回転不変性を高めたものであり、SIFT よりも画像特徴の記述能力が高いことが知られている。

従来の SIFT<sup>11)</sup> や SURF<sup>7)</sup>、GLOH などのエッジ方向ヒストグラムに基づく局所特徴量は、次のような手順で求められている。

- (1) 特徴点の周辺領域 (patch) を微分する。
- (2) patch 内のエッジ方向ヒストグラムのピークを求め、主軸とする (図 4-(a))。
- (3) 主軸が上を向くように patch を回転させる (図 4-(b))。
- (4) 再び patch 内を微分する。
- (5) patch 内を複数の subpatch に分割し、sub-

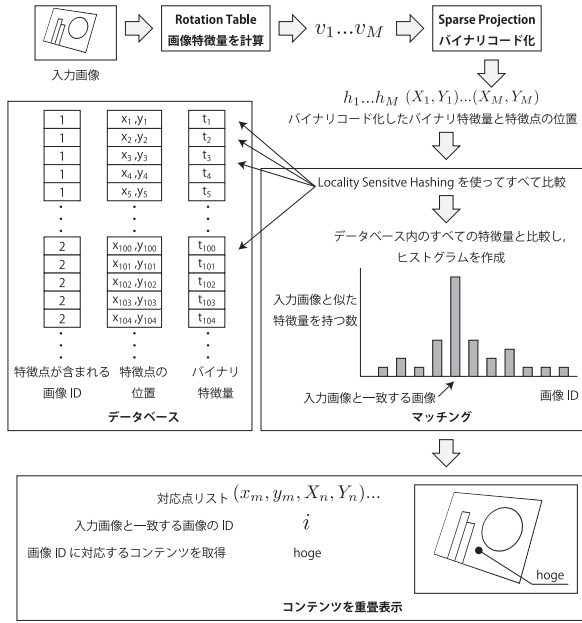


図 3 提案手法の概要  
Fig. 3 proposed system

patch ごとにエッジ方向ヒストグラムを求める。  
以上の処理は、patch 内で微分が 2 回行われており、冗長である。また主軸が上を向くように patch を回転させるとき、サブピクセル単位での画像変換に伴い膨大な浮動小数点演算が行われるため、非常に計算コストが高い。

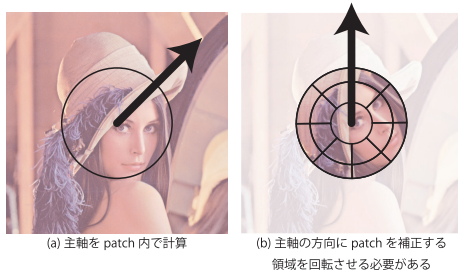


図 4 GLOH 特徴量  
Fig. 4 GLOH

一方で、FAST GLOH 特徴量は、主軸が上を向くように patch を回転させるのではなく、subpatch のパターン自体を回転させる (図 5)。このとき、subpatch 内のエッジ方向は、主軸の傾き分だけずれていることになるため、それぞれ微分方向から主軸の傾きを引くことで補正する。この方法は微分の計算が一度だけで済み、patch のサブピクセル単位での回転変換も不要であるため、非常に高速である。

以上の手順は、ルックアップテーブルを用いることで、さらなる高速化が可能である。例えば、あらかじめ

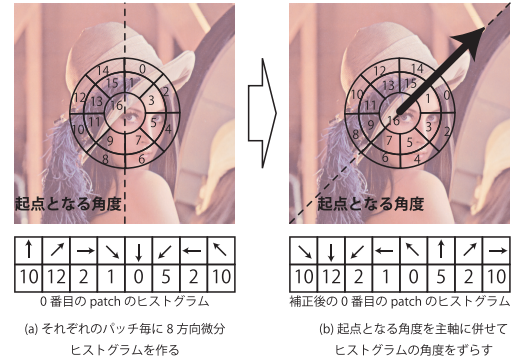


図 5 FAST GLOH 特徴量  
Fig. 5 FAST GLOH

め 40 通りの subpatch の回転パターンを計算しておき、メモリ上に保存しておけば、周辺ピクセルがどの subpatch に属するかを一度のメモリアクセスだけで決定できる。また、微分方向の補正量についても、あらかじめ補正パターンを用意しておけば、こちらも一度のメモリアクセスだけで決定できる。

### 3.3 バイナリコード変換：Sparse Projection

次に、FAST GLOH 特徴量をバイナリコード化する。バイナリコード化することにより、実数からなるベクトルデータよりも小さなデータ量で特徴量を表現できるようになる。また、局所特徴量同士の距離計算を単純なハミング距離計算で置き換えられるようになるため、後段のマッチングの速度を高速化できる。このバイナリコード化の最も古典的な方法として、次の random projection が広く用いられている。

入力ベクトル  $v \in \mathbb{R}^D$  に対して、 $d$  次元 (ビット) のバイナリコードは、次式 1 で与えられる。

$$h_k(v) = \frac{\text{sgn}[w_k^T v] + 1}{2}, k = 1, \dots, d \quad (1)$$

$h_k(v)$  は、バイナリコードの  $k$  番目のビット値であり、 $D$  は局所特徴量の次元数である。入力ベクトル  $v$  は、事前にデータの平均が原点になるようにオフセットを加え、さらに原点からの  $L_2$  距離が 1 になるように正規化しておく。 $w_k$  は、 $D$  次元空間において、中心が原点の半径 1 の超球上の点から、一様分布に従ってランダムにサンプリングしたベクトルである。

バイナリコードのハミング距離は、その極限において、元の入力ベクトルの空間におけるコサイン距離と一致する。任意の二つのベクトル  $v_1, v_2$  を  $h_k(v)$  によって変換したとき、それぞれのビットが異なる値になる確率が、 $v_1$  と  $v_2$  のベクトルが成す角度  $\theta$  に比例することが知られている<sup>13)</sup>。

$$\Pr(h_k(v_1) \neq h_k(v_2)) = \frac{\arccos(v_1^T v_2)}{\pi} \quad (2)$$



左辺の確率の値は、バイナリコードのハミング距離とみなせる。このため、十分長いビット列を求めれば、元の空間におけるベクトルのコサイン距離とそのハミング距離が一致するのである。FAST GLOH 特徴量の場合、128 ビット程度で十分であることを実験的に確かめた。これについては次節で詳しく述べる。

また、 $w$  を一様分布からサンプリングするのではなく、次のようにサンプリングしても、式 2 が近似的に成立することが知られている<sup>14)</sup>。

$$w_i = \begin{cases} +1 & \text{with prob. } \frac{1}{2\sqrt{D}} \\ 0 & \text{with prob. } 1 - \frac{1}{\sqrt{D}} \\ -1 & \text{with prob. } \frac{1}{2\sqrt{D}} \end{cases} \quad (3)$$

これは、very sparse random projection と呼ばれる。このときの  $w$  は疎行列であり、かつ非ゼロの要素が  $-1$  または  $+1$  のみで構成されているため、 $w^T v$  の計算において乗算が不要であり、また加減算の回数が非常に少なくてすむという利点がある。そこで、Fast GLOH 特徴量を式 1,3 に基づいてバイナリコード化する。  $D = 136, d = 128$  のときにおける、計算コスト削減の効果を表 2 にまとめる。特徴ベクトルの次元数  $D$  が大きいほど、計算コスト削減に有効であることがわかる。

表 2 バイナリコード化の計算回数の比較

Table 2 Comparison of computation steps for binary encoding

計算方法	乗算	加算	乗算と加算 回数の総和
random proj.	$d \times D$	$d \times (D - 1)$	34688
sparse proj.	0	$\frac{d \times (D-1)}{\sqrt{D}}$	1482

### 3.4 データベースの構築

あらかじめデータベースに登録する検出対象を撮影した参照画像も同様にバイナリコードに変換された特徴量で表現する。このため、参照画像  $I_i$  に含まれる特徴点の数を  $M_i$  と定義すると、データベースのレコード数は、 $\sum_i M_i$  となる。それぞれの特徴点ごとに、その特徴点が属する参照画像の ID と、その特徴点の参照画像中での位置をデータとして付加しておく。

### 3.5 特定物体認識

データベースに含まれる局所特徴量の中から最も類似する局所特徴量を検索し、投票による原理で近い参照画像をデータベースから探し出す<sup>15)</sup>。ここでは、ハミング距離用の近似最近傍探索アルゴリズムとして、Locality Sensitive Hashing(LSH)を用いた。LSH によって、類似する局所特徴量が見つかった場合、その

特徴点が含まれる画像 ID をデータベースから調べ、参照画像のヒストグラムに 1 票投票する。入力画像に含まれる全ての局所特徴量に対して投票処理を適用し、類似する特徴点を最も多く持つ参照画像を検索結果画像とする。

### 3.6 重畳表示

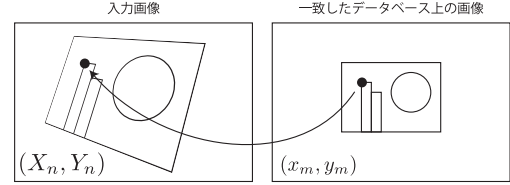


図 6 入力画像と検索結果画像間における対応点

Fig. 6 Corresponding points between query image and retrieved image

特定物体認識が完了すると、入力画像と検索結果画像間における、対応点のリストが得られる(図 6)。このリストを用いて、検索結果画像  $(x, y)$  から入力画像座標  $(X, Y)$  へ写像するホモグラフィ行列  $H$  を計算する。この行列  $H$  を用いることで、データベースの参照画像上の座標を、入力画像における座標に変換できる。そこで、あらかじめ重畳したい情報を座標位置とともにデータベースに登録しておくことで、入力画像にアノテーションを加えることができる。

## 4. 評価実験

提案手法の有効性を検証するため、その特徴である FAST GLOH 特徴量、Sparse Projection およびシステム全体の評価を行った。

### 4.1 FAST GLOH 特徴量および Sparse Projection の計算速度

SIFT, SURF, FAST GLOH の計算時間と、FAST GLOH を Sparse Projection によってバイナリコードに変換したときの計算時間を比較した結果を、図 7 に示す。横軸は画像サイズ、縦軸はフレームレートである。1024 × 768 ピクセルの画像を 5 枚用意し、計算時間の平均をプロットした。低解像度の画像は、画像を縮小して生成した。実験に用いた計算機は Intel Core 2 6400 2.13GHz、メモリ 3GB のデスクトップ PC であり、開発環境は Matlab および Visual C++ 2008 を用いた。

提案手法は、SIFT, SURF よりも 2 倍以上高速に特徴量を計算できることが示された。また、Sparse Projection によってバイナリコード変換を行っても、処理速度はほとんど変化しなかった。このことから、Sparse Projection 手法は、特徴量の計算時間に影響

を与えない程度の計算量でバイナリコードに特徴量を変換できることが示された。

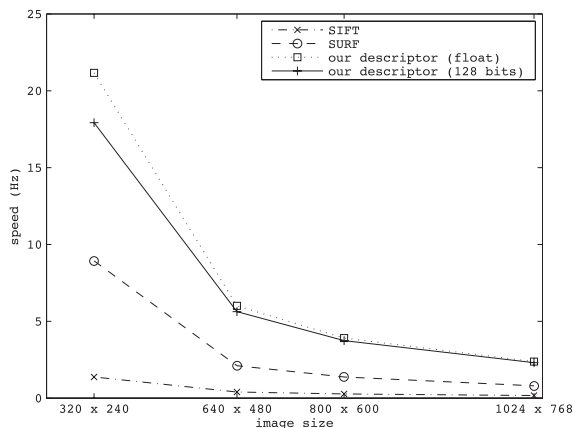


図 7 局所特徴量の計算速度

Fig. 7 Computational performances of obtaining local descriptors

## 4.2 Sparse Projection のハミング距離精度

Sparse Projection および Random Projection によって得られたバイナリコードのハミング距離は、元の特徴空間におけるベクトル間角度を近似したものである。Sparse Projection は、射影ベクトルが疎であるという制約を加えているため、Random Projection よりもハミング距離の近似性能が劣化する可能性がある。そこで、Sparse Projection によって作成されるバイナリコードの性能を評価した。

関連文献<sup>17)</sup>で採用されている評価方法に従い、近傍探索性能を評価した結果を図 8 に示す。横軸はビットコードの長さであり、縦軸は近傍探索性能を MAP(mean average precision) で評価した結果である。以下、評価の方法を簡潔に記載する。評価用に準備した複数枚の画像から 20000 点の局所特徴量をランダムに選択し、1000 点をテストデータ、19000 点を訓練データとして抽出した。局所特徴量をバイナリコード化する前に、訓練データの局所特徴量の平均を求めておき、テストデータの重心を補正した。テストデータに対して、もともとの特徴空間 ( $\mathbb{R}^D$ ) で、上位 2% の近傍にある局所特徴量を正しい近傍点であると定義し、ハミング距離空間内においても正しい近傍が得られるかを評価した。なお、実験環境は、前述の評価実験と同条件とした。

結果、図 8 に見られるように Sparse Projection と Random Projection に顕著な性能差は見られず、両方ともビットコードを長くすればするほど、近似性能が向上することが分かった。このことから、提案手法の Sparse Projection は、画像検索の精度と速度を両

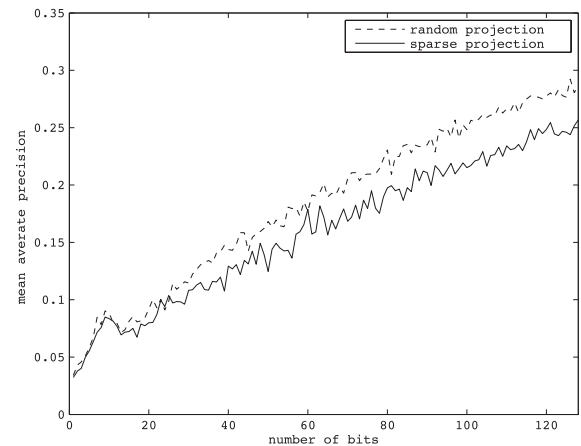


図 8 Sparse Projection と Random Projection の性能比較  
Fig. 8 Computational performances between S.P. and R.P.

立させる優れた手法であるといえる。

## 4.3 特定物体認識の性能評価

次に、提案手法の特定物体認識の性能評価について述べる。まずはじめに、一般的な雑誌の誌面やポスターなど 2 次元コンテンツを 50 枚程度データベースに登録し、認識精度を評価したところ、SIFT を用いた従来手法、提案手法いずれの場合においても 100% の True Positive が達成された。この予備実験の結果から、提案する局所特徴量の記述能力の高さが伺えるが、従来手法との性能比較を精査するために、3 次元の物体画像データベースである Recognition Benchmark Images<sup>16)</sup>を用いて性能を追試した。このデータセットは、3 次元物体がいくつかの方向から撮影されているため、2 次元コンテンツよりもはるかに認識が困難であるという特徴を持つ。

Recognition Benchmark Images の中から、同一対象の 4 枚の画像を 1 セットとし、合計 100 セットを評価対象とした (図 9)。1 セットの中から 1 枚をデータベース用の画像とし、残りを 3 枚をクエリ画像として、それぞれデータベース画像合計 100 枚、クエリ画像合計 300 枚を準備した。評価は、SIFT, FAST GLOH, 128bit にバイナリコード化した FAST GLOH, 64bit にバイナリコード化した FAST GLOH の 4 つの特徴量を用いて行った。実験環境は、前述の二つの評価実験と同条件とした。4 つの特徴量を用いた認識結果の ROC 曲線を図 10 に示す。

図 10 から分かるように、SIFT 特徴量には及ばぬものの、128bit 表現による Fast GLOH 特徴量は False Positive が 1% のときに True Positive が 40% 以上という高い性能を達成することができた。なお、SIFT 特徴量よりも性能が劣った主要な原因として、主軸の



図 9 特定物体認識の精度評価のためのデータセット

Fig.9 Image dataset for evaluation

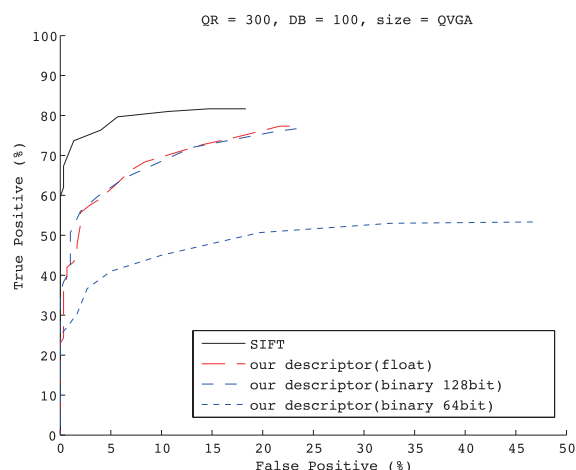


図 10 ROC 曲線

Fig.10 ROC curve

回転パターンの粒度を 40 通りに制限したことから生じた、量子化誤差の影響があると考えられる。従って、主軸の回転パターンの粒度を 360 通りにするなど、量子化誤差の影響を極力回避することで、更なる性能改善が達成できると考える。

128bit にバイナリコード化した FAST GLOH 特徴量は、バイナリコード化を適用しない FAST GLOH 特徴量と同等の性能であった。一方、64bit の場合は顕著な性能劣化が見られた。従って、FAST GLOH 特徴量のバイナリコード長としては 128bit 程度が適切であると考えられる。

次にメモリ使用量の観点から考察する。この実験と同条件で 100 枚の画像を登録したデータベース中の特徴量のサイズ、物体認識時に必要なメモリ量を表 3 に示す。提案手法の特徴量が 128bit のバイナリコードであるのに対し、SIFT 特徴は 128 次元の実数ベクトルである。SIFT 特徴を単精度の実数で表現した場合、データベースサイズは、提案手法の 32 倍の 54MB にも達し、モバイルデバイス上での実装は困難である。一方、表 3 からわかるように提案手法は、実行時のメモリ使用量も少なく、モバイルデバイスの主記憶領域上に十分展開可能なサイズである。従って、提案手法はメモリ使用量という観点から非常に優れているといえる。

表 3 提案手法におけるメモリ使用量

Table 3 Memory usage of our proposing method

内訳	提案手法	SIFT
データベース中の特徴量のサイズの総和	1.7MB	54MB
LSH に必要なメモリサイズ	2.2MB	-

#### 4.4 AR アプリケーションとしての評価

従来環境 Windows PC とモバイルデバイス (iPhone4) 上で、提案手法による AR アプリケーションを実装し、その応答速度を評価した。データベースに約 50 ページ分程度の本や表紙を登録し、各ページ上にコンテンツを重ね表示する AR アプリケーションを実装した。図 11 に、iPhone4 で動作するアプリケーションの例を示す。図 11 から、データベース上に登録されている画像上の適切な位置に複数のコンテンツを重ね表示できていることがわかる。表 4 にアプリケーションの実行時の計算速度を示す。Windows PC の実験環境は前述の 3 つの実験と同条件で、モバイルデバイスは iPhone4(iOS4.1) を利用した。

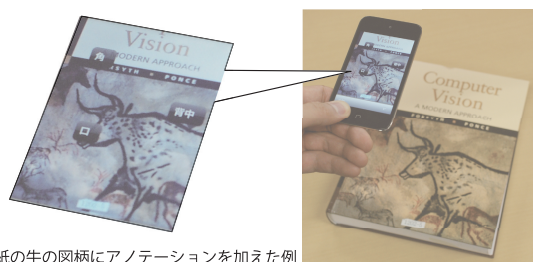


図 11 iPhone で動作するテスト AR アプリケーション

Fig.11 Prototype of AR application on iPhone

表 4 応答速度

Table 4 Speed of response

環境	1 フレームの画像の認識にかかる時間
Windows PC	84 msec(11.9Hz)
iPhone4	500msec(2Hz)

本システムは、表 4 に示した程度の性能で、認識結果をオンラインで重ね表示できることがわかった。特に、モバイルデバイスである iPhone4 では、デバイス単体で 2Hz の速度で物体認識が可能であった。Chen らの実験結果では<sup>9)</sup>、リモートサーバー上でほとんどの計算を実行しているにも関わらず、特徴量抽出と認識に合計で 650msec の計算時間を要している。この関連研究と本研究の結果を比較すると、計算リソースがかなり限定された iPhone4 でこの応答速度を実現できる提案手法は、非常に有用であるといえる。

今回の実験では、50 ページの画像をデータベース

に登録した。このデータベースへの登録画像の数が増えると LSH によるマッチングの処理時間が長くなる。提案手法によって大規模な画像検索を実現するために、LSH の近傍探索性能を向上させることが今後の課題である。LSH の高速化に関する研究は文書検索や Web サービスの分野を中心に多く行われており、これらの研究成果を取り入れてゆく予定である。

## 5. ま と め

計算資源が限られたモバイルデバイス上で AR アプリケーションを実現する手法を提案した。提案手法は、高速に計算できる局所特徴量 FAST GLOH, 特徴量を高速にバイナリコードに変換する Sparse Projection の二つの仕組みから成り立っており、iPhone に代表されるようなモバイルデバイスでも動作が可能であることを示した。提案手法を用いて iPhone4 の実機上で AR アプリケーションを実装し、認識速度を評価した結果、従来研究のサーバーサイドでの物体認識速度と同レベルの速度で動作することを示した。今後は、上述の大規模な画像検索の実現とともに、提案手法を用いて実装した AR アプリケーションのユーザビリティ評価等も行っていく予定である。

## 参 考 文 献

- 1) Aristides Gionis, Piotr Indyk, Rajeev Motwani, Similarity Search in High Dimensions via Hashing, Proceedings of the 25th International Conference on Very Large Data Bases table of contents, pp.518-529(1999)
- 2) セカイカメラ, <http://sekaicamera.com/>
- 3) 暦本純一, 2次元マトリックスコードを利用した拡張現実感の構成手法, Workshop on Interactive Systems and Software (WISS'96)
- 4) Nobuko Taketa, Kenichi Hayashi, Hirokazu Kato and Shogo Noshida, Virtual Pop-Up Book Based on Augmented Reality, HUMAN INTERFACE AND THE MANAGEMENT OF INFORMATION. INTERACTING IN INFORMATION ENVIRONMENTS, Lecture Notes in Computer Science, 2007, Volume 4558, pp.475-484(2007)
- 5) Sivic, J. And Zisserman, A., Video Google: A Text retrieval Approach to Object Matching in Videos, Proceedings, IEEE International Conference on Computer Vision, pp.1470-1477(2003).
- 6) David G. Lowe, "Object recognition from local scale-invariant features," International Conference on Computer Vision, Corfu, Greece (September 1999), pp. 1150-1157.
- 7) Herbert Bay, Andreas Ess, Tinne Tuytelaars, Luc Van Gool, Speeded-Up Robust Features (SURF), Computer Vision and Image Understanding archive Vol.110, Issue 3, pp.346-359(2008)
- 8) D. G. Lowe, Distinctive image features from scale-invariant keypoints, International Journal of Computer Vision, Vol.60 No.2 pp.91-110(2004)
- 9) D. Chen, S. Tsai, R. Vedantham, R. Grzeszczuk, B. Girod, Streaming Mobile Augmented Reality on Mobile Phones, 8th International Symposium on Mixed and Augmented Reality, Orlando, Florida, October 19-22, 2009 (ISMAR 2009)
- 10) Jianbo Shi and C. Tomasi, Good features to track, Computer Vision and Pattern Recognition, Proceedings, IEEE Computer Society Conference, pp.593-600(1994)
- 11) Edward Rosten, Tom Drummond, Fusing points and lines for high performance tracking, IEEE International Conference on Computer Vision, Vol.2, pp.1508-1511(2005)
- 12) Krystian Mikolajczyk and Cordelia Schmid, "A Performance Evaluation of Local Descriptors", IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, Vol.27, pp.1615-1630, No.10,(2005)
- 13) Goemans, Michel X.; Williamson, David P. "Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming", Journal of the ACM 42 (6): 1115-1145(1995)
- 14) Ping Li, Trevor J. Hastie, Kenneth W. Church, Very sparse random projections, Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, pp.287-296(2006)
- 15) Herve Jegou, Matthijs Douze, Cordelia Schmid, Improving Bag-of-Features for Large Scale Image Search, International Journal of Computer Vision, Vol.87, Issue 3, pp.316-336(2010)
- 16) D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), volume 2, pages 2161-2168, June 2006.
- 17) Sequential Projection Learning for Hashing with Compact Codes, Jun Wang, Sanjiv Kumar, Shih-Fu Chang. In Proc. of the 27th International Conference on Machine Learning (ICML), 2010.