近似最近傍探索の多段階化によるリアルタイム物体認識システム

野口　和人†　　本道　貴行†　　岩村　雅一†　　黄瀬　浩一†

† 大阪府立大学大学院工学研究科　〒 599–8531 大阪府堺市中区学園町 1–1
E-mail: {noguchi,hondo}@m.cs.osakafu-u.ac.jp, {masa,kise}@cs.osakafu-u.ac.jp

あらまし　本稿では，局所記述子として PCA-SIFT を用いたリアルタイム物体認識システムを提案する．従来，PCA-SIFT を用いて物体を認識する場合，局所記述子の数が膨大となるため，局所記述子の抽出，ならびに局所記述子の照合にかかる処理時間が問題であった．抽出の問題を解決するため，提案システムでは，GPU(Graphical Processing Unit) を用いる．照合の問題に対しては，多段階化された近似最近傍探索器を用いる解決策を提案する．これら 2 つの要因によって，例えば，十万物体のデータベースを用いた場合でも，リアルタイム認識が可能となる．

キーワード　物体認識, PCA-SIFT, 近似最近傍探索, 多段階化, GPU, CUDA

Real-Time Recognition of Objects by Cascading Approximate Nearest Neighbor Searchers

Kazuto NOGUCHI†, Takayuki HONDO†, Masakazu IWAMURA†, and Koichi KISE†

† Graduate School of Engineering, Osaka Prefecture University
1–1 Gakuencho, Naka, Sakai, Osaka, 599–8531 Japan
E-mail: {noguchi,hondo}@m.cs.osakafu-u.ac.jp, {masa,kise}@cs.osakafu-u.ac.jp

Abstract　We demonstrate a system for real-time object recognition which employs PCA-SIFT as local descriptors of images. Object recognition based on PCA-SIFT has posed problems of processing time for both extraction and matching of local descriptors because of their large number. To solve the problem about extraction, the proposed system is equipped with a GPU (Graphical Processing Unit). For the problem about matching, we utilize a cascaded approximate nearest neighbor searchers. These two factors enable us real-time object recognition even with a database of 100,000 objects.

Key words　Object recognition, PCA-SIFT, Approximate nearest neighbor search, Cascade, GPU, CUDA

1. はじめに

カメラ付き携帯電話やウェアラブルコンピュータを用いて，リアルタイムで物体を認識したいという要求が高まっている．リアルタイム認識が可能となると，ロボットの視覚に用いるなどさまざまな利用法が考えられる．

すでに我々は，局所記述子として PCA-SIFT(PCA Scale-Invariant Feature Transform) [1] によって得られる特徴量を用いて，高速かつ高精度に認識を行う手法 [2] を提案している．この手法は，デジタルカメラ等で撮影したポスターなどの平面物体を，データベースに登録されている大量の画像の中から高速に探し出すことができる．例えば，10 万画像のデータベースを用いた場合に，認識率 97.75%，処理時間 8.6ms を実現している．しかし，PCA-SIFT 特徴量は抽出に数秒かかってしまうため，そのままではリアルタイムに処理することは難しい．

この問題を解決するため，近年，GPU(Graphics Processing Unit) を局所記述子の抽出に利用する研究 [3] が行われている．GPU とは，3D グラフィックスの計算に用いるプロセッサである．近年の GPU の進化は著しく，高速な並列計算能力を持つようになった [4]．例えば，本デモシステムで用いる GeForce8800GTX は，最新の CPU(Core2 Duo 3GHz) と比較して，7〜8 倍の浮動小数点演算性能がある．そこで，この GPU の演算性能をグラフィックス以外の汎用目的に活用しようとする動き (GPGPU(General-Purpose computations on GPUs) [5]) が活発になってきている．しかし，GPU は本来 3D グラフィックス描画用であるため，汎用目的の利用には，種々の制約がある．また，プログラミングには NVIDIA Cg(C for Graphics) や Microsoft HLSL(High Level ShadingLanguage), GLSL (OpenGL Shading Language) といったシェーダ言語を利用しなければならない．シェーダ言語は本来 3D グラフィッ

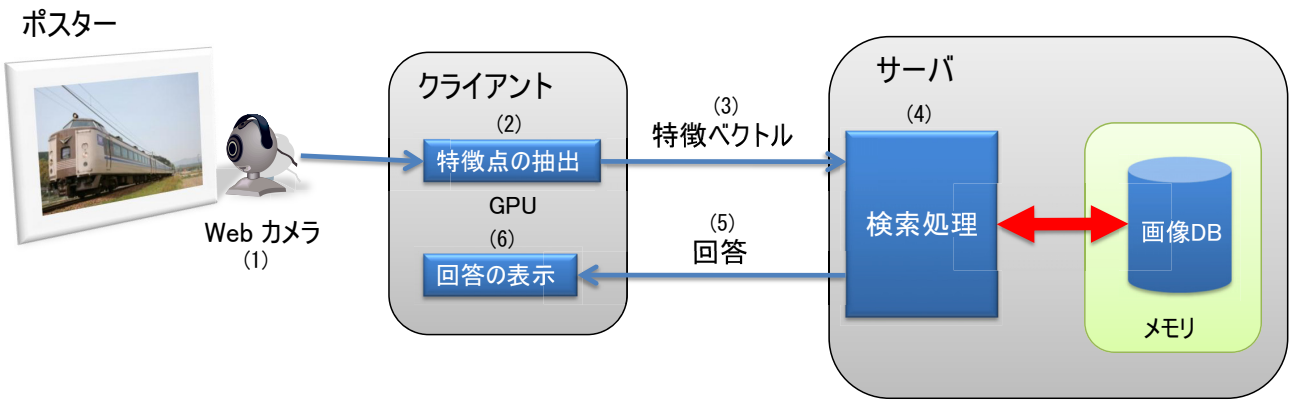

図 1　システムの構成

クス用の言語であるため，汎用計算を行う上での拡張性や保守性に問題がある．

この問題は，CUDA(Compute Unified Device Architecture) (注1)を利用することで解決できる．CUDA は GPU をより汎用的に利用するためのハードウェアとソフトウェアの統合環境である．現状では利用できる GPU は NVIDIA GeForce8 シリーズに限られるものの，プログラミング言語には，標準的な C 言語を用いることができ，シェーダ言語と比べると開発も容易である．

そこで本稿では，多段階化された近似最近傍識別器と CUDA を用いる，リアルタイム認識システムを提案する．

2. デモシステム

提案するシステムの構成を図 1 に示す．全体の処理の流れに沿って図中の (1) から順に説明する．

（1） ポスターや写真といった平面物体を撮影する．撮影には，安価な Web カメラを利用する．撮影画像のサイズは VGA(640 × 480) から QVGA(320 × 240) 程度である．

（2） Web カメラから取得した画像に対して，PCA-SIFT 特徴量を抽出する．この処理は，CPU を利用すると数秒かかってしまう．そのため，本システムでは GPU を利用し高速に処理を行う．この処理で得られる PCA-SIFT 特徴量は，36 次元の特徴ベクトルであり，画像の内容に依存するが，その数は数百から数千程度である．

（3） 得られた PCA-SIFT 特徴量をサーバへ転送する．本システムでは，クライアントサーバ間は LAN を利用して接続しているが，インターネットを使用しての接続も可能であると思われる．

（4） サーバでは，データベースに登録された 10 万枚の画像中から，対応する画像を検索する．データベースには，画像そのものではなく，あらかじめ抽出しておいた特徴ベクトルを登録している．10 万画像のデータベース場合には，およそ 2 億の特徴ベクトルが登録されている．画像データベースはメインメモリ上に構築されているため，非常に高速な処理が可能となっている．なお，サーバのメモリ使用量は，10 万画像の場合で約 20GB である．

（5） 得られた回答をクライアントに転送する．

（6） 回答を表示する．撮影した物体にさまざまな情報を関連付けしておくことで，Web のアクセスやアプリケーションの起動などが考えられるが，本システムでは，画像データベースに登録されている画像を表示する．

なお，デモに用いる計算機は以下のものである．

- クライアント
  CPU　Athlon64 X2 6000+ (3.0GHz)
  メモリ　4GB
  GPU　GeForce8800GTX
  OS　WindowsXP
- サーバ
  CPU　Opteron 2.8GHz
  メモリ　32GB
  OS　Linux Debian

謝辞　本研究の一部は日本学術振興会科学研究費補助金（基盤研究 (B)19300062）の補助による．

文　献

[1] Y. Ke and R. Sukthankar: “PCA-SIFT: A more distinctive representation for local image descriptors”, Proc. of CVPR2004, Vol. 2, pp. 506–513 (2004).
[2] 野口, 黄瀬, 岩村：“近似最近傍探索の多段階化による物体の高速認識”, 画像の認識・理解シンポジウム (MIRU2007) 論文集 (2007).
[3] S. Sinha, J. Frahm and M. Pollefeys: “GPU-based Video Feature Tracking and Matching”, Technical report, Tech. Rep. TR06-012, University of North Carolina at Chapel Hill, May 2006.
[4] CUDA Programming Guide Version 0.8.2, http://developer.nvidia.com/object/cuda.html.
[5] GPGPU, http://www.gpgpu.org/.

(注1)：http://developer.nvidia.com/object/cuda.html よりダウンロードできる．