社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
TECHNICAL REPORT OF IEICE.

スマートフォンで動作するリアルタイム文書画像検索

竹田　一貴†　　黄瀬　浩一†　　岩村　雅一†

† 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: †takeda@m.cs.osakafu-u.ac.jp, †{kise, masa}@cs.osakafu-u.ac.jp

あらまし　本稿では，スマートフォンで動作する新たな文書画像検索システムを提案する．本システムは，Locally Likely Arrangement Hashing と呼ばれるリアルタイム文書画像検索法に基づいており，印刷文書を通して現実世界と仮想世界をスムーズにリンクさせるものである．スマートフォンを印刷文書にかざすだけで，対応する電子文書を検索し，それに関連付けられている情報をユーザに提示することができる．本システムによって，ユーザは印刷文書を，関連情報を取得するための新たなメディアとして利用できるようになる．

キーワード　文書画像検索，リアルタイム処理，スマートフォン，ヒューマンコンピュータインターフェース，拡張現実

1. はじめに

近年，タブレット PC やスマートフォンのような高機能携帯端末が急速に普及している．これらは従来の携帯端末に比べ高解像度なディスプレイを備えており，電子書籍を読みやすいという利点を持つ．このような利点とスマートフォン等の持つ携帯性の高さから，ユーザがいつでもどこでも電子書籍を利用できる環境が整いつつある．これにより，電子書籍が世界的に普及していくことが見込まれており，電子書籍に対するさまざまなサービスが提供され始めている．そのようなサービスの例として，“Qlippy” [1] や “Layered Reading” [2] が挙げられる．Qlippy とは，読者が書籍の任意の部分についての意見やコメントを共有することができるサービスである．Layered Reading も同様のサービスである．このようなサービスを利用することで，読者同士が電子書籍を通じてコミュニケーションをとることができるようになる．また，意見・コメントは電子書籍上に表示されるため，ユーザビリティの高いものとなっている．

一方で，読みやすさや携帯性の面から，多くの人が従来の紙媒体の文書に対して高い利便性を感じており，その需要が滞ることはない．それにもかかわらず，印刷文書を対象とした Qlippy や Layered Reading のようなサービスは存在しない．印刷文書に対しても電子文書のようなサービスを適用することができれば，人と文書の関わり方を変えることができると考えられる．しかし，この要求を満たすためには解決しなければならない問題が 1 つある．それは，印刷文書とコンピュータの間には大きなギャップがあるという問題である．既に存在する印刷文書自体には，どんなディジタル情報をも付加することはできないことは明らかである．このような現実世界とディジタル世界のギャップを埋めるためには，印刷文書を撮影し，その撮影画像に対して働きかけることが考えられる．また，撮影するためのデバイスとしては，ユーザビリティの観点から，一般的に普及している携帯端末を用いることが望ましいと考えられる．

一般的な携帯端末を用いて印刷文書に情報を付加することができる手法として，スマートフォンを用いた印刷文書への拡張現実を考える．これは，各文書に関連付けられた情報を撮影画像に重畳表示するというものである．このタイプの拡張現実を実現するためには，撮影している印刷文書をリアルタイムで特定することが必要となる．撮影された印刷文書を特定する手法に，文書画像検索がある．携帯端末で動作する文書画像検索法はすでにいくつか提案されている [3] [4] [5]．しかし，これらの手法は拡張現実を実現できるほどの高速性を持ち合わせていない．処理速度の問題を解決しうる手法として，Locally Likely Arrangement Hashing (LLAH) [6] という文書画像検索法がある．LLAH は，リアルタイム処理が可能なほどの高速性を備えていることが知られている．また，撮影された印刷文書の位置と姿勢を推定することも可能である．これらの性質により，関連情報が文書のどの部分に関連付けられているかをリアルタイムで示すことが可能となる．したがって，LLAH は拡張現実との相性が良い手法であると考えられる．

本稿では，スマートフォン上で動作するリアルタイム文書画像検索を提案する．文書画像検索法として LLAH を利用することにより，印刷文書に対する拡張現実を実現する．しかし，スマートフォンは高度な処理能力を持たないため，処理速度の面で問題が生じる．これにより，関連情報がスムーズに描画されなくなる．この問題を解決するため，以下の 2 点の改良を加える．1 つは，撮影された印刷文書をトラッキングすることである．トラッキング処理は LLAH の処理よりも高速なため，関連情報の位置姿勢をより短い間隔で調整することが可能となる．もう 1 つは，マルチスレッド処理を用いて処理の分散を図るものである．本システムでは，印刷文書の撮影・表示と LLAH，トラッキング，拡張現実の描画という 4 つのスレッドと利用する．また，本システムを用いたさまざまなサービスの提案も行

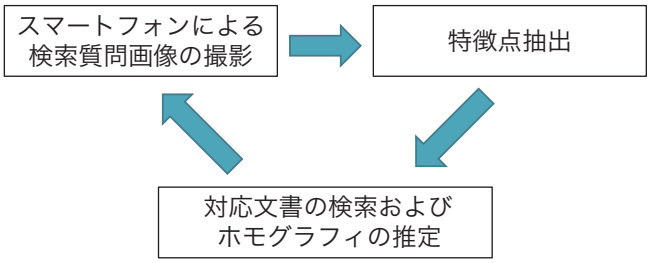

図 1　リアルタイム検索処理

う．実験結果から，本システムは検索精度 95%，処理時間 10 fps で動作することが確認された．

2. 関 連 手 法

ここでは，スマートフォンで動作する文書画像検索の従来手法について紹介する．

“Mobile Retriever” [3] は，“token pair” と “token triplet” に基づく検索手法である．token pair は複数の単語の shape code で定義されるものである．token triplet は 3 つの単語とそのオリエンテーションで構成される．これらに用いられる単語は，OCR によって認識されている．この手法では，スマートフォンで撮影した印刷文書の画像をサーバへ送信することで，対応する文書を検索する．Mobile Retriever は大規模なデータベースから高い精度で検索できることが確認されている．しかし，1 クエリ当たり約 4 秒という非常に長い処理時間を必要とする．つまり，この手法はリアルタイム性が欠落しており，拡張現実を滑らかに表示させることはできない．

“HotPaper” [4] もまた，文書のテキストに基づく特徴量を用いる手法の 1 つである．この特徴量は，Brick Wall Coding Features (BWC) と呼ばれる．BWC は，複数の単語のバウンディングボックスを表す局所特徴量である．これは，スケールに変化に不変であり，軽微な射影変換に対応できる．この手法は処理が高速であるという特徴を持ち，1 クエリあたり約 300[ms] で検索可能である．一方で，データベースのサイズは非常に小さく，5000 ページ以下である．また，検索精度も 60[%] 以下である．このようなスケーラビリティの悪さは，サービスの幅を制限することになる．

上記の問題に加えて，Mobile Retriever と HotPaper は，ある共通の問題を抱えている．これらの手法は単語のレイアウトに基づいているため，日本語や中国語のような，分かち書きのなされていない言語で記述された文書に対して有効ではない．言語による有効性の違いもまた，サービスを制限することにつながる．

“PaperUI” [5] は，この問題を解決しうる文書画像検索法である．PaperUI では文書を特定するために 7 つのアプローチを採用しているが，その中の 1 つに SIFT [7] や FIT などの局所特徴量を用いるものがある．これらはテキストベースの特徴ではないため，文書の違いによらず検索可能であると考えられる．しかし，SIFT は大量のメモリを必要とするため，PaperUI はスケーラビリティに関して問題がある．多くのページに対して

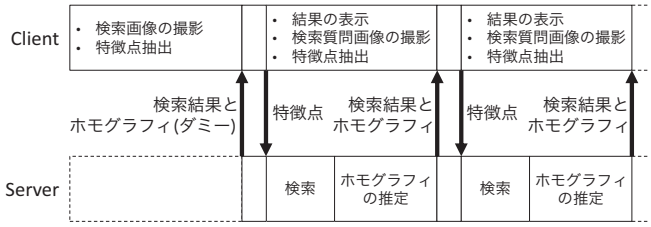

図 2　クライアントサーバシステム

サービスを提供するためには，大規模データベースを扱える能力が必要となる．

本稿では，上記の問題をすべて解決することのできる手法として，Locally Likely Arrangement Hashing (LLAH) [6] を利用する．LLAH は高いスケーラビリティを持ち，リアルタイム処理を実現できることが知られている．具体的には，2,000 万ページのデータベースにおいて，検索精度 99[%]，処理時間 50[ms] で検索可能であることが確認されている [8]．また，LLAH はすでに多言語で記述された文書への対応もされている [9]．さらに，検索処理を通じて撮影画像の位置や姿勢を推定することも可能である．これにより，関連情報を表示する位置や姿勢を簡単に決定することができる．上記の性質から，LLAH はスマートフォン上に関連情報を自然な形で表示するという要求に適していると考えられる．しかし，スマートフォンは高い処理能力を持っていないため，拡張現実を実現するためには LLAH の処理による負荷を削減する必要がある．

3. ソフトウェアデザイン

3.1 クライアントサーバシステム

リアルタイム文書画像検索は，図 1 に示すような処理を繰り返すことによって実現することができる．まず，検索質問画像として，スマートフォンを用いて印刷文書を撮影する．次に，検索質問画像から特徴点を抽出する．そして，抽出された特徴点に基づいて検索質問画像に対応する文書を検索する．LLAH では，それぞれの特徴点に対して特徴量が定義される．特徴量は，周囲の特徴点の配置に基づいて計算される．特徴量に基づいて，データベースから対応する特徴点を探索することにより，検索は実現される．さらに，特徴点の対応関係から，ホモグラフィを推定することができる．このホモグラフィを利用することにより，オリジナルの文書上における撮影画像の位置や姿勢を特定することが可能となる．

これらのステップは独立して実行することができるため，クライアントサーバシステムを利用することによって処理を並列化することができる．図 2 に，クライアントおよびサーバが担う役割を示す．本システムにおいては，スマートフォンがクライアントの役割を担うことになる．クライアントにおいて，検索質問画像が撮影され，そこから特徴点が抽出される．これらの処理は並行して実行されることに注意されたい．この並列化については，次節で詳しく説明する．抽出された特徴点は，TCP ソケットを用いてサーバへ送信される．サーバでは，受信した特徴点に基づいて，対応文書を検索し，ホモグラフィが

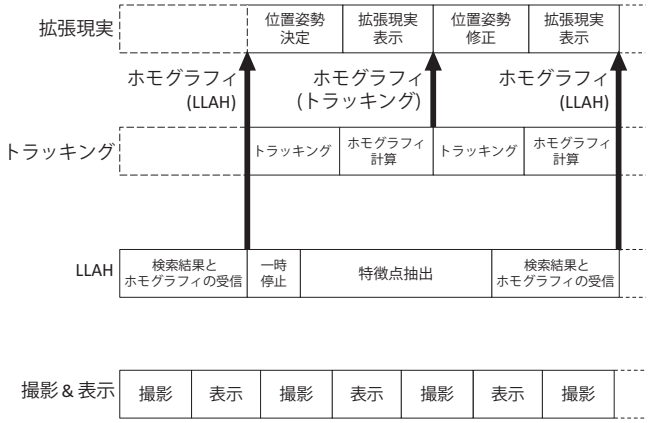

図 3　マルチスレッド処理とトラッキングによる位置姿勢の修正

推定される．それと同時に，クライアントでは再び検索質問画像を撮影し，特徴点を抽出する．クライアントとサーバの両方の処理が終了すると，検索結果とホモグラフィがクライアントに送り返され，特徴点がサーバへ送信される．繰り返し処理の並列化により，本システムは高いフレームレートを実現することができる．

3.2 改良手法

次に，処理速度の問題を解決するための手法ついて説明する．

拡張現実を実現する上で，仮想物体の位置や姿勢を現実世界に適合させることは非常に重要である．また，関連情報を滑らかに表示するために，リアルタイム処理も重要になる．仮想物体の位置姿勢に関しては，検索処理を通じて計算されたホモグラフィを用いることによって正確に求めることができる．しかし，LLAH の処理が既存手法よりも高速であるとはいえ，スマートフォン上で実行すると処理速度の面で依然問題が残る．これは，スマートフォンが高度な処理能力を持たないためである．したがって，単純に LLAH のみを適用するだけでは，拡張現実のスムーズな描画は実現できない．この問題を解決するために，撮影された印刷文書の追跡と，マルチスレッド処理を利用する．これらの改良により，自然な拡張現実を実現する．具体的な処理については，以下で説明する

3.2.1 マルチスレッド

まず，処理速度の改善を図るために，マルチスレッド処理を適用する．本システムでは，図 3 に示すように，検索質問画像の撮影・表示と LLAH，トラッキング，拡張現実の描画という 4 つのスレッドを利用する．検察質問画像の撮影と LLAH の処理を並列化することによるメリットは，撮影画像をスムーズに表示することができるという点にある．また，1 フレームあたりの処理時間を削減するために，撮影画像の表示と拡張現実の描画を並列化させている．ここで，他のスレッドの処理時間を確保するため，LLAH を 1 回実行するごとに 100[ms] だけ処理を一時停止させている．

3.3 トラッキング

トラッキングの処理は，LLAH の処理よりも高速に実行できる．従って，LLAH の処理と平行して撮影された印刷文書の追跡を行うことで，より短い間隔で関連情報の表示位置や姿勢を

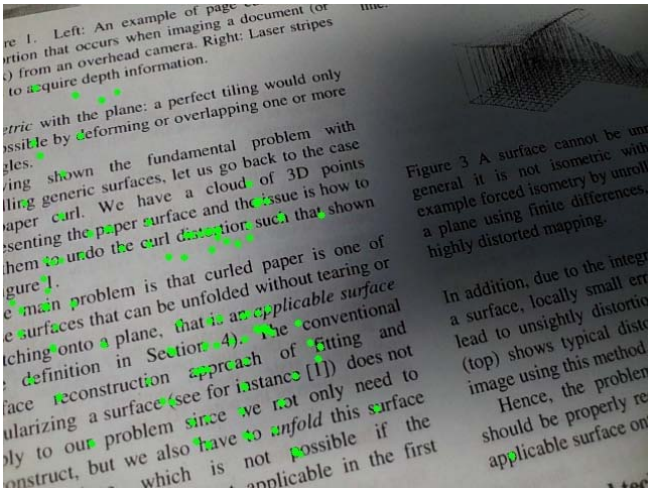

図 4　追跡特徴点の例

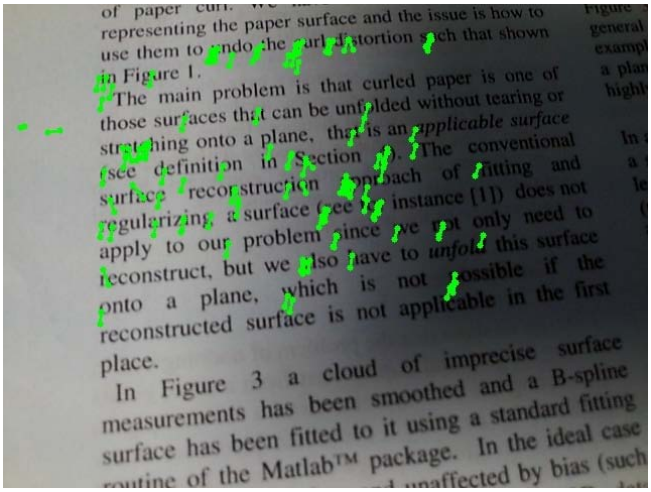

図 5　トラッキングの様子

修正することが可能となる．

印刷文書を追跡する方法を以下で示す．まず，初期フレームから印刷文書を追跡するための特徴点を抽出する．この特徴点は，ハリスオペレータ [10] を用いて抽出される．図 4 に特徴点の例を示す．次に，Lucus-Kaneda 法 [11] を用いて特徴点のオプティカルフローを計算し，追跡を行う．図 5 に特徴点を追跡している様子を示す．追跡できなかった特徴点に対しては処理を打ち切る．また，前フレームと次フレームの間で追跡できた特徴点数について閾値を設定する．追跡点数が閾値を下回る場合，そこまで追跡してきた特徴点を削除し，新たに初期特徴点を抽出する．トラッキングによって，連続フレームの対応点を得ることができる．対応点の集合から RANSAC [12] を適用して，前フレームから次フレームへのホモグラフィを計算することができる．このホモグラフィを利用することで，関連情報を重畳するための位置姿勢を修正する．

図 3 にトラッキングによる位置姿勢修正の方法を示す．まず，LLAH で計算されたホモグラフィを用いて関連情報の位置姿勢を決定する．次に，トラッキングの結果からホモグラフィを計算し，位置姿勢を修正する．トラッキングは高速だが失敗しがちなため，この修正を多数連続して適用することはできな

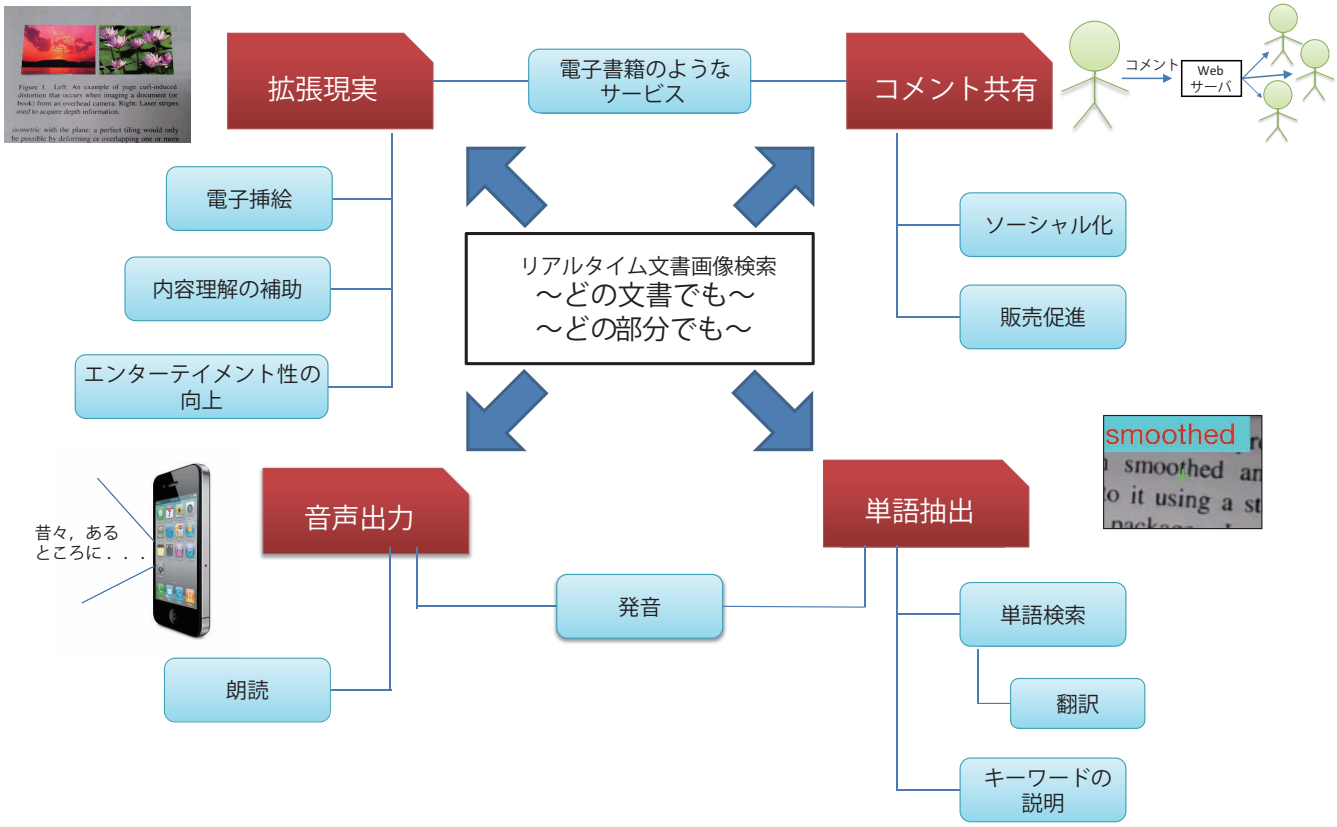

図 6　アプリケーションの概念図

い．この問題は，トラッキングによる修正を 2〜3 回行うごとに，LLAH で計算されるより信頼性の高いホモグラフィを適用することで解決する．図 3 で示すような処理を繰り返すことにより，印刷文書へのスムーズな拡張現実を実現する．

3.4　アプリケーション

本システムを用いたアプリケーションについて考える．図 6 に概念図を示す．本システムの特徴は，どの文書のどの部分を撮影しても検索できるというものである．この特徴を生かしたサービスとして，拡張現実や単語抽出，音声出力が考えられる．拡張現実の具体的な例としては，小説等で挿絵のないページを撮影すると，その内容にあった絵を重畳表示するような電子挿絵が挙げられる．また，動画や 3 次元物体を表示させることにより，文書の枠を超えたエンターテイメント性を実現できると考えられる．さらに，コメント共有機能と組み合わせることにより，Qlippy のようなサービスを印刷文書に対して提供できるようになる．音声出力に関しては，撮影した部分から朗読を開始するようなトリガーの役割を果たす．また，単語抽出と組み合わせることで，単語の発音を瞬時に聞くことのできるアプリケーションを実現できる．

以下では，本システムを用いて実装したアプリケーションを紹介する．

3.4.1　拡 張 現 実

上述のように，本システムは印刷文書への拡張現実を実現できる．その一例を図 7 に示す．拡張現実とは，撮影画像に関連情報を重畳表示するものである．関連情報としては，テキストや画像，ハイライト，アンダーライン，手書き文字などを考えている．また，三次元オブジェクトを表示することも可能である．この技術によって，追加情報を得るための新たなメディアとして印刷文書を利用することができる．

3.4.2　単語・文章抽出

他のアプリケーションとして，撮影範囲にある単語や文章を抽出することを考える．検索された文書のオリジナルの PDF から，単語の位置を得ることができる．本システムでは撮影範囲の推定が可能なため，ユーザは撮影範囲内の単語を得ることができる．図 8 に単語抽出の例を示す．このように，画像の中心にある単語を正確に抽出できていることがわかる．この技術のメリットは，文字認識を必要としないことにある．このアプリケーションを利用することにより，単語の意味を検索したり，キーワードの説明を取得することができる．

3.4.3　電子文書の検索

LLAH の検索結果として，文書名を取得することができる．この文書名に基づいて，オリジナルの PDF を取得し，表示することができる．このサービスにより，大量のファイルの中から望みの文書ファイルを探索する時間を節約することができる．また，ユーザはこの PDF 上に意見やコメントを書き込むことができる．これらの注記は共有され，拡張現実に反映することができる．

4.　実　　験

本稿では，提案手法を用いて上記のサービスを実装した．これらのサービスを有用なものにするためには，本システムが高精度かつ高速な検索処理を実現しなければならない．そこで，

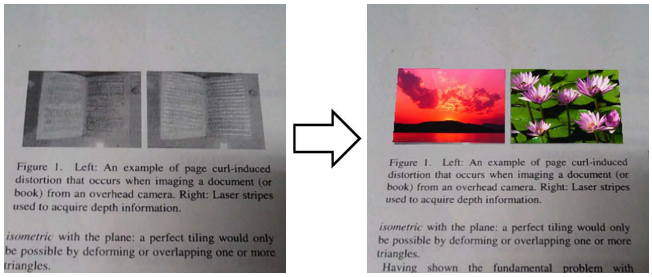

図 7　文書への拡張現実

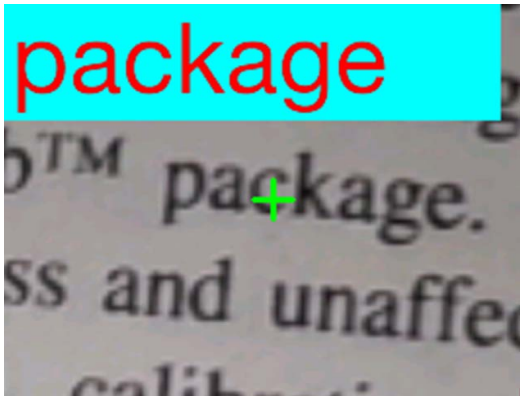

図 8　単語抽出の例

本システムの検索精度と処理時間について調査した．

実験のために，100 万ページの文書を登録したデータベースを作成した．これらの文書は主にインターネットから収集されたものである．実験で用いたスマートフォンは，1.2GHz Dual Core CPU と 1GB メモリを備えたものである．

4.1 検索精度

はじめに，検索精度について測定した．本実験のため，データベースに含まれるページの中から 100 ページを抜き出し，それぞれを 5 カ所から撮影した．データベースに登録されている画像の例を図 9 に，検索質問画像の例を図 10 に示す．従って，検索質問画像数は 500 枚である．これらの検索質問画像は，文書全体ではなく，一部分を撮影したものである．

実験結果から，95.2[%] という高い精度で検索可能であることが分かった．図 11 に正しく検索できた検索質問画像の例を示す．このように，文章を多く含む検索質問画像が正しく検索しやすいものだった．図 12 に，検索に失敗した検索質問画像の例を示す．左上の画像を除いて，ほとんど文章が含まれていないことがわかる．LLAH では安定した検索を実現するために多くの文字が撮影されていることが望ましいため，これらの画像に対して正しく検索することは困難であると考えられる．一方，左上の画像は文章を多く含んでいるにもかかわらず，検索に失敗した．これは，ドット [.] やクオーテーション [“] のような細かな記号が多く含まれているためである．このような小さな連結成分が含まれていると，安定して特徴点抽出が行えないことがわかっている [13]．そのため，より安定した特徴点が抽出できるような処理を加える必要があると考えられる．

4.2 処理時間

処理時間についても同様に測定した．本実験では，さまざまな処理にかかる時間を計測した．

図 9　登録画像

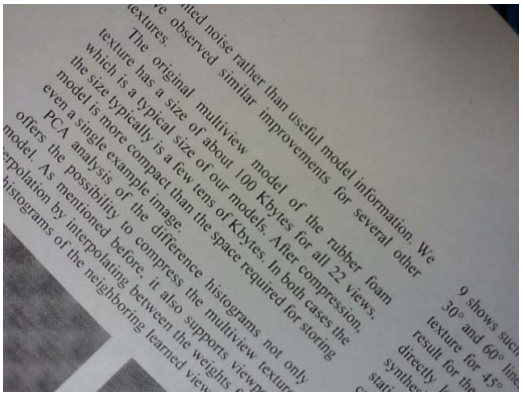

図 10　検索質問画像

検索質問画像の撮影と表示にかかる時間にかかる時間は約 40[ms] であった．これは，ほかの処理と比較して無視できるほど高速である．

次に，LLAH の処理にかかる時間について示す．クライアントであるスマートフォンにおいて，LLAH の処理は 2 つある．1 つは特徴点抽出処理であり，もう 1 つはサーバとの通信処理である．ここで，撮影画像の撮影・表示は別スレッドで実行されているので，処理時間には含まない．特徴点抽出にかかった時間は 100[ms] であり，通信処理には 2[ms] の時間を要した．これに加えて，ほかのスレッドの処理時間を確保するために，1 回の実行ごとに 100[ms] だけ一時停止する．結果として，LLAH の処理にかかる処理時間は 202[ms] となった．

トラッキング処理には，初期特徴点の抽出処理と特徴点の追跡処理がある．特徴点抽出には 278[ms] 必要であり，特徴点追跡には約 70[ms] の時間がかかった．特徴点抽出処理は，トラッキングの初期化時にのみ実行されることから，めったに適用されない．従って，処理時間は追跡処理にかかる時間が支配的となる．つまり，トラッキング処理は 1000/70 fps，つまり 10 fps 以上の処理速度で実行可能である．

最後に，拡張現実にかかる時間について示す．これには，関連情報の位置姿勢を決定する処理と，それらを描画する処理が含まれる．位置姿勢決定にはほとんど時間がかからず，描画についてのみ約 70[ms] 必要とした．結果として，拡張現実を実

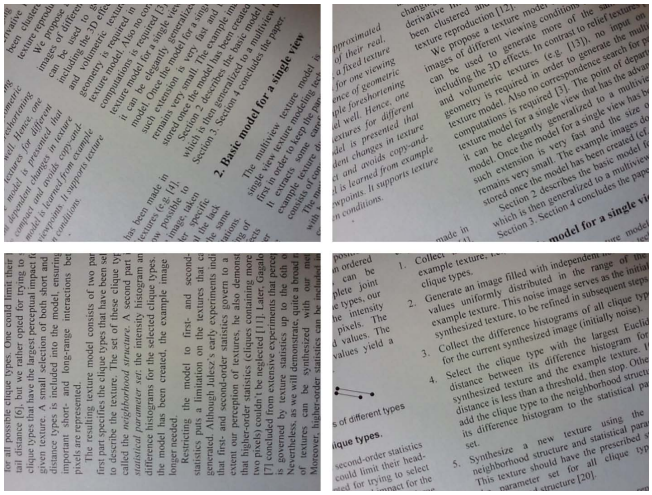

図 11　検索成功例

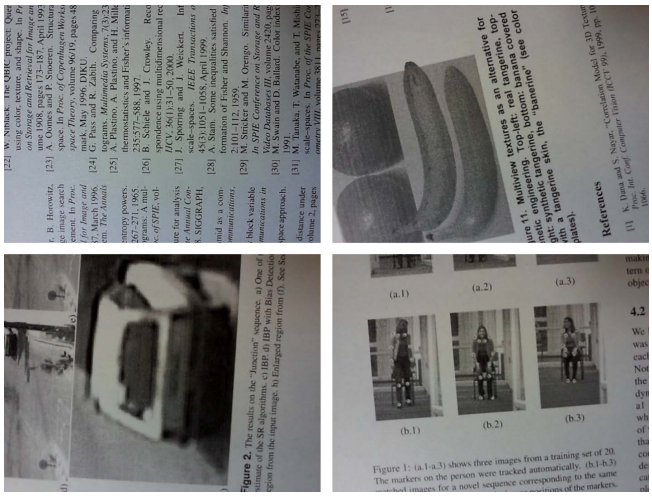

図 12　検索失敗例

行するには 70[ms] の時間がかかることになる．また，位置姿勢の更新頻度はトラッキングの速度に依存するため，本システムは 10 fps 以上の処理時間で関連情報の描画が可能であることが分かった．

ユーザの感知できる処理時間は，拡張現実の描画時間である．この時間はほかの処理よりも速いわけではないため，これがユーザの知覚を支配する．従って，本システムを実行するための処理時間は 10 fps 以上となる．

5. まとめと考察

本稿では，LLAH を適用することによってスマートフォン上で動作するリアルタイム文書画像検索を提案した．処理時間に関する問題を解決するため，撮影文書の追跡とマルチスレッド処理を適用した．実験結果から，95.2[%] の検索精度を実現し，10 fps 以上の処理速度で拡張現実を描画することかできることを確認した．また，本システムを用いることによって実現できるサービスの提案も行った．

今後の課題は，処理時間をより高速にするための改良を加えることや，より魅力的なサービスを提案することである．

謝　　辞

本研究の一部は，JST CREST および日本学術振興会科学研究費補助金基盤研究（B）(22300062)，挑戦的萌芽研究 (21650026) の補助による．

文　　献

[1] http://qlippy.com/.
[2] http://kaleyedoscope.net/.
[3] X. Liu and D. Doermann, “Mobile retriever: access to digital documents from their physical source,” Int. J. Doc. Anal. Recognit., vol.11, pp.19–27, Sept. 2008.
[4] B. Erol, E. Antúnez, and J.J. Hull, “Hotpaper: multimedia interaction with paper using mobile phones,” Proceeding of the 16th ACM international conference on Multimedia, pp.399–408, 2008.
[5] Q. Liu and Chunyuan Liao., “Paperui,” Proceeding of the 4th International Workshop on Camera-Based Document Analysis and Recognition, pp.3–10, sep 2011.
[6] T. Nakai, K. Kise, and M. Iwamura, “Use of affine invariants in locally likely arrangement hashing for camera-based document image retrieval,” Lecture Notes in Computer Science (7th International Workshop DAS2006), vol.3872, pp.541–552, feb 2006.
[7] D.G. Lowe, “Distinctive image features from scale-invariant keypoints,” Int. J. Comput. Vision, vol.60, pp.91–110, Nov. 2004.
[8] K. Takeda, K. Kise, and M. Iwamura, “Memory reduction for real-time document image retrieval with a 20 million pages database,” Proceedings of 19th International Conference on the Pattern Recognition, 2011.
[9] T. Nakai, K. Kise, and M. Iwamura, “Real-time retrieval for images of documents in various languages using a web camera,” Proceedings of the 10th International Conference on Document Analysis and Recognition (ICDAR2009), pp.146–150, jul 2009.
[10] C. Harris and M. Stephens, “A combined corner and edge detector,” Proc. Alvey Vision Conf, pp.147–151, 1988.
[11] B. Lucas and T. Kanade, “An iterative image registratio technique with an application to stereo vision,” Proceedings of 7th International Joint Conference on Artificial Intelligence, pp.674–679, 1981.
[12] M.A. Fischler and R.C. Bolles, “Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography,” Commun. ACM, pp.381–395, 1981.
[13] K. Takeda, K. Kise, and M. Iwamura, “Real-time document image retrieval for a 10 million pages database with a memory efficient and stability improved llah,” Proceedings of the 11th International Conference on Document Analysis and Recognition (ICDAR2011), 2011.