

## SIFT 特徴量記述処理のヒストグラム並び替え近似及び小領域分割による省メモリかつ高速な SIFT アーキテクチャの提案

### Proposal for a Memory-Efficient and High-Speed SIFT Architecture through Approximation of Feature Descriptor Processing by Histogram Rearrangement and Subdivision of Regions

光元 歩<sup>†</sup>      弘中 哲夫<sup>†</sup>  
Ayumu MITSUMOTO    Tetsuo HIRONAKA

#### 1. はじめに

特徴量抽出は自動運転やロボットビジョンなど多岐に渡る分野で活用されている。特徴量抽出アルゴリズムのうち、SIFT[1]はその頑強性から広く利用されているが、計算量が多いため高速な SURF や ORB が代替として提案されている。しかし、SIFT は機械学習を除くほとんどの特徴量抽出手法と比べて優れた精度を示すことが判明している[2]。

本研究では、ストリーム処理を用いた SIFT アーキテクチャを提案する。本アーキテクチャでは、ガウシアン・DoG ピラミッドの生成、特徴点検出とローカライズ、勾配強度・方向の算出を Huang ら[3]のストリーム処理アーキテクチャと同様に行い、その後の SIFT 特徴量記述処理をヒストグラムの並び替えて近似することで処理速度を向上させた Jiang ら[4]のアルゴリズムを改良した手法で行う。これは、Jiang らの研究では処理速度を優先する余り精度が悪くなっているという問題があるからである。そこで、本研究では Jiang らのアルゴリズムの並列性を向上させることで精度を犠牲にすることなく、性能向上が実現できる SIFT アーキテクチャを提案する。

#### 2. 提案する SIFT アーキテクチャ

##### 2.1 ストリーム処理

図 1 は提案アーキテクチャの全体構成を示す。このアーキテクチャでのストリーム処理は 3 ステージで行われる。

ステージ 1 では、入力画像のピクセルを 25 行のバッファに毎サイクル書き込む。さらに、ガウシアン・DoG ピラミッド生成では、バッファから読み込んだピクセルからぼかし値と DoG 値を計算し、それぞれ GF バッファと DoG バッファへ書き込む。

続くステージ 2 では、特徴点検出とローカライズで DoG バッファからデータを読み込み、KP バッファに特徴点の有無に関するフラグを書き込む。さらに、勾配強度・方向算出で GF バッファのデータを読み込み、Grad バッファに勾配情報を書き込む。

最後のステージ 3 では、KP バッファから読み出したフラグが True の場合、Grad バッファの勾配情報を用いて特徴量記述処理を開始する。

ステージ 1, 2, 3 はパイプライン的に並列に動作するが、ステージ 3 で特徴量記述処理が開始された場合、ステージ 1, 2 の処理を停止してステージ 3 の処理が優先される。

提案アーキテクチャでは特徴量記述に関して改善を行った、次節でその説明を行う。

<sup>†</sup> 広島市立大学大学院情報科学研究科 Graduate School of Information Sciences, Hiroshima City University

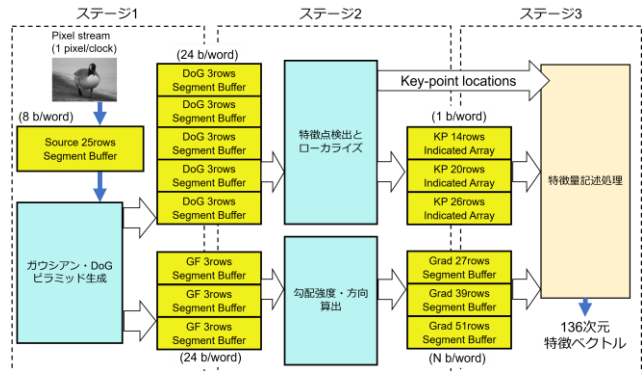


図 1 提案アーキテクチャの全体構成

##### 2.2 特徴量記述処理

図 2 は、提案アーキテクチャにおける特徴量記述処理のブロック図を示す。

提案アーキテクチャでは、まず check keypoint で KP バッファを確認し、特徴点が存在する場合に Grad バッファの勾配情報を用いて find keypoint orientation で主要方向を求め、generate subregion histogram で 17 個の勾配方向ヒストグラムを算出する。提案アーキテクチャは、2.3 節で説明する記述子窓の小領域への分割により Jiang らを上回る並列性を実現し、処理性能を向上した。generate subregion histogram は静的に決定された領域からヒストグラムを算出するため find keypoint orientation と並列に動作する。

続いて、reordering histogram と reordering subregion で主要方向に基づいてヒストグラム内の要素と、記述子窓の中心を除いた 16 個の勾配方向ヒストグラム同士を並べ替える。

最後に、histogram to vector で並べ替えたヒストグラムを正規化して 136 次元の特徴ベクトルを算出する。なお、Jiang らは正規化の処理は行っていない。

次節では、generate subregion histogram について説明する。

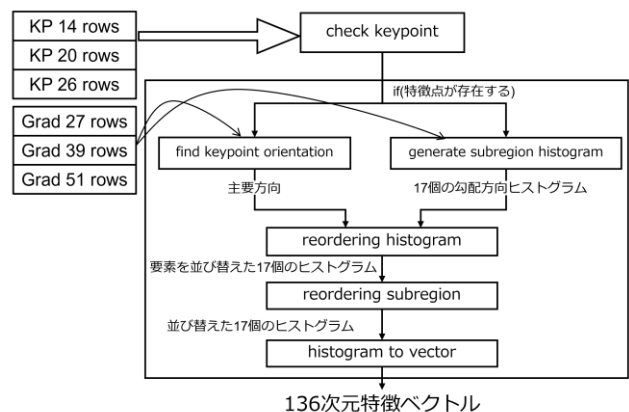


図 2 提案アーキテクチャの特徴量記述ブロック図

## 2.3 特徴記述子の生成処理

ここでは、generate subregion histogram で行う特徴記述子の生成処理の詳細について述べる。

提案アーキテクチャでは、図 3 に示すように  $M \times M$  の記述子領域を 36 個の小領域へ分割する。  $M$  の値は計算の簡略化のために 3 の倍数かつ奇数としており、提案アーキテクチャでは回転操作の近似を考慮して 27, 39, 51 とした。また、黒塗りされた部分は複数小領域が重複した箇所を示す。提案アーキテクチャでは、各小領域において勾配方向ヒストグラムを算出し、それらを足し合わせることで 17 個のヒストグラムを計算している。具体的には、図 3 における①, ②, ⑦, ⑧の各小領域で算出された 8 方向の勾配方向ヒストグラムの要素を足し合わせることで 1 領域分の勾配方向ヒストグラムの算出が可能であり、これを 17 個の領域分計算を行っている。

小領域の計算は依存関係が無いため最大 36 並列で計算を行うことが可能であり、この並列性により特徴量記述処理を高速化することが出来る。

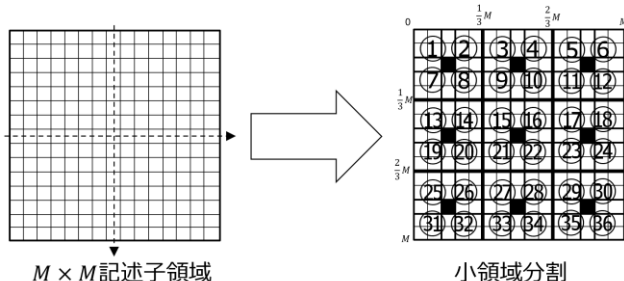


図 3 特徴記述子領域の小領域分割

## 3. 評価

本章では、提案アーキテクチャのメモリ消費量、精度、並列性について先行研究と比較し、評価を行う。

### 3.1 メモリ消費量

表 1 は各アーキテクチャのメモリ消費量の比較を示す。提案アーキテクチャの特徴点検出部は Huang らのアーキテクチャを基としており、そのメモリ消費量は 0.87Mb である。特徴量記述部のメモリ消費量は勾配情報のビット数  $N$  に依存して変動する。提案アーキテクチャの合計メモリ消費量は  $0.87+0.131*N$  Mb であり、Huang らと条件を揃えて 4.28Mb ( $N=26$ )となる。これは、他の研究と比較し省メモリなアーキテクチャであることが分かる。

表 1 各アーキテクチャのメモリ消費量比較

	Huang	Jiang	提案
画像サイズ	640×480	512×512	640×480
メモリ消費量	特徴点検出	7.8 Mb	0.87 Mb
	特徴量記述	4.86 Mb	4.5 Kb
合計メモリ消費量	5.73 Mb	7.8 Mb	$0.87+0.131*N$ Mb

### 3.2 精度

図 4 は、各研究における精度評価の比較である。図の上には対象画像名と加えた変換が示されている。図の縦軸はマッチング精度、横軸「1vsX」は元画像(1)と変換後の画像(X)間のマッチングを示し、Xの値は変換の大きさを示す。

図下部の Original は従来の SIFT を 32 ビット浮動小数点型で実装したものである。また、Proposed は Original の特徴量記述部を 2.3 節で述べた提案手法を実装したものである。特徴量の類似度はユークリッド距離で計算し、閾値  $k = 0.8$  で対応点を算出した。マッチング精度は Mikolajczyk のデータセットで評価した[5]。図 4 より、Proposed の精度は Jiang らのアーキテクチャと比較し大幅に精度が向上されていることが分かる。また、Original の精度が 80% に近づいた場合は Proposed の精度が大幅に低下しやすいという傾向はあるが、全体的に近しい精度が達成できていることも分かる。

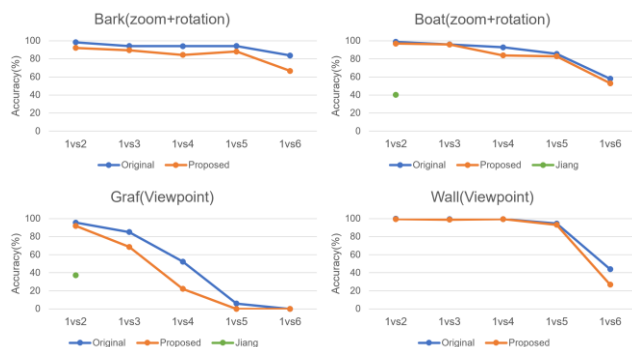


図 4 精度の評価結果

### 3.3 小領域分割の並列性

ヒストグラムに値を加算する操作が 1 サイクルで完了するとした時、17 個の勾配方向ヒストグラムの算出に掛かるサイクル数は Jiang らのアルゴリズムでは  $M \times M$  サイクルとなる。それに対し、提案アーキテクチャでは小領域の計算に  $(\lceil M/6 \rceil)^2$  サイクル、36 個の小領域が持つ 8 方向のヒストグラムを足し合わせるのに  $8 \times 36 = 288$  サイクル、合計で  $(\lceil M/6 \rceil)^2 + 288$  サイクルとなる。

提案アーキテクチャの  $M (= 27, 39, 51)$  を用いる場合、スケールに応じて約 2.3~7.0 倍の高速化が達成できることが分かる。

## 4. まとめ

本研究では、SIFT 特徴量記述処理にヒストグラム並び替え近似と小領域分割を導入することで省メモリかつ高速な SIFT アーキテクチャを提案した。メモリ消費量は Huang らと Jiang らのアーキテクチャと比較した場合、それぞれ 25%、45% と削減できた。また、提案手法の精度は Original に対し条件付きで近しい精度を達成した。今後の課題として実装に向けてハードウェアの更なる詳細化が課題である。

### 参考文献

- [1] D. G. Lowe, "Distinctive image features from scale-invariant key points," *Int. J. Comput. Vis.*, vol. 60, no. 2 (2004).
- [2] I. Suárez, J. M. Buenaposada and L. Baumela, "Revisiting Binary Local Image Description for Resource Limited Devices," *arXiv:2108.08380* (2021).
- [3] F.-C. Huang, S.-Y. Huang, J.-W. Ker and Y.-C. Chen, "High-Performance SIFT Hardware Accelerator for Real-Time Image Feature Extraction," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 3 (2012).
- [4] J. Jiang, X. Li and G. Zhang, "SIFT Hardware Implementation for Real-Time Image Feature Extraction," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 7 (2014).
- [5] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 10 (2005).