



アルゴリズム×ハード×ソフト 最適化による超高速画像処理

名古屋工業大学 情報工学類
ネットワークプログラム
准教授 福嶋 慶繁

あらゆる画像処理を、
アルゴリズム ×
ハードウェア ×
コンパイラ
により最適化することで、最高の性能で計算
できる技術を研究しています。

社会背景と技術的課題

社会背景：映像配信，ロボット，自動運転，医療診断など，多様な場面で高精細な画像処理が不可欠。

リアルタイム性・省電力性・コスト効率も重要。

技術動向：多くの場合で、画像処理は既存ライブラリに依存しており，それらは「ハードウェアの性能」を全く引き出せていない。



映像配信



ロボット制御



自動運転



ドローン制御



電子顕微鏡



医用画像処理

本技術の特徴

ロアルゴリズム

ロハードウェア

ロコンパイラ・プログラミング言語

の3つ相互に考慮しながら最適化することで

「最速の画像処理」を「数行」で実現

- 既存画像処理ライブラリでは実現できない
並列性・高キャッシュ効率を達成

```

(a) Clean C++ : 9.94 ms per megapixel
void blur(const Image &in, Image &blurred) {
    Image tmp(in.width(), in.height());

    for (int y = 0; y < in.height(); y++)
        for (int x = 0; x < in.width(); x++)
            tmp(x, y) = (in(x-1, y) + in(x, y) + in(x+1, y))/3;

    for (int y = 0; y < in.height(); y++)
        for (int x = 0; x < in.width(); x++)
            blurred(x, y) = (tmp(x, y-1) + tmp(x, y) + tmp(x, y+1))/3;
}
    
```

tiling

multithreading

(b) Fast C++ (for x86) : 0.90 ms per megapixel

```

void fast_blur(const Image &in, Image &blurred) {
    _m128i one_third = _mm_setr_epi16(21846);
    #pragma omp parallel for
    for (int yTile = 0; yTile < in.height(); yTile += 32) {
        _m128i a, b, c, sum, avg;
        _m128i tmp[(256/8)*(32+2)];
        for (int xTile = 0; xTile < in.width(); xTile += 256) {
            const uint16_t *inPtr = &(in(xTile, yTile+y));
            for (int y = -1; y < 32+1; y++) {
                for (int x = 0; x < 256; x += 8) {
                    a = _mm_loadu_si128((__m128i*)(inPtr-1));
                    b = _mm_loadu_si128((__m128i*)(inPtr+1));
                    c = _mm_load_si128((__m128i*)(inPtr));
                    sum = _mm_add_epi16(_mm_add_epi16(a, b), c);
                    avg = _mm_mulhi_epi16(sum, one_third);
                    _mm_store_si128(tmpPtr++, avg);
                    inPtr += 8;
                }
            }
            tmpPtr = tmp;
            for (int y = 0; y < 32; y++) {
                _m128i *outPtr = (__m128i *)(&(blurred(xTile, yTile+y)));
                for (int x = 0; x < 256; x += 8) {
                    a = _mm_load_si128(tmpPtr+(2*256)/8);
                    b = _mm_load_si128(tmpPtr+256/8);
                    c = _mm_load_si128(tmpPtr++);
                    sum = _mm_add_epi16(_mm_add_epi16(a, b), c);
                    avg = _mm_mulhi_epi16(sum, one_third);
                    _mm_store_si128(outPtr++, avg);
                }
            }
        }
    }
}
    
```

vectorization

fusion

(c) Halide : 0.90 ms per megapixel

```

Func halide_blur(Func in) {
    Func tmp, blurred;
    Var x, y, xi, yi;

    // The algorithm
    tmp(x, y) = (in(x-1, y) + in(x, y) + in(x+1, y))/3;
    blurred(x, y) = (tmp(x, y-1) + tmp(x, y) + tmp(x, y+1))/3;

    // The schedule
    blurred.tile(x, y, xi, yi, 256, 32)
        .vectorize(xi, 8).parallel(y);
    tmp.chunk(x).vectorize(x, 8);

    return blurred;
}
    
```

従来技術との比較

| | OpenCV | scikit-image | 本技術 |
|---------|--------------|-----------------|-------------------------------|
| 原理・方法 | C++ ライブラリ | Python ライブラリ | 独自 ライブラリ・ プログラミング 言語 |
| 並列性 | ○ | △ | ◎ |
| キャッシュ効率 | △ | × | ◎ |
| 利便性 | ◎ | ◎ | ○ |

具体的な取り組み

□ CPUによるマルチコア並列化・SIMDベクトル化^[1]

□ 並列化・SIMD化による最速の画像変換

□ GPUによるCUDA最適化^[2]

□ 世界最速の畳み込みを実現

□ WebAssemblyによる最適化^[3]

□ ブラウザでネイティブ環境並みの画像処理

□ 量子計算機による画像処理表現^[4]

□ 量子コンピュータによる先進的な画像処理

□ 画像処理特化プログラミング言語^[5]

□ 言語レベルで統合的に最適化

[1] "x86/64CPU上における効率的な画像変換のための
ルックアップレジスタテーブル補間," 電子情報通信学会
画像工学研究会 (IE), Dec. 2024.

[2] "Tensor Coreによる二次元畳み込みの高速化," 電子
情報通信学会画像工学研究会 (IE), June 2025.

[3] "Performance evaluation of image convolution with
WebAssembly," International Workshop on Advanced
Image Technology (IWAIT), Jan. 2023.

[4] "量子画像表現の復元可能性に関する定量的評価,"
電子情報通信学会画像工学研究会 (IE), June 2025.

[5] "畳み込み乱択化のためのHalideによるドメイン固有
な記述," 画像符号化シンポジウム(PCSJ)/映像メディア
処理シンポジウム(IMPS), Nov. 2021.

求める連携先・メッセージ

- モバイル, クラウド, CPU, GPUといった, あらゆる環境でハードウェアの性能を最大限引き出す画像処理を実現する研究をしています.
- ロボットビジョン, 自動運転, 監視システム, Webシステムなどを構築するときに, 使える環境で最大限の性能を発揮する画像処理エンジンを構築することを得意とする研究室です.
- 「高速な画像処理」にご興味のある方はお問合せください.



本技術に関する情報

試作品の状況

提示可

※提供の際は諸手続が必要となるため、下記問合せ先までご連絡願います。

研究フェーズ



文献・特許の情報

- [1] "x86/64CPU上における効率的な画像変換のためのルックアップレジスタテーブル補間," 電子情報通信学会画像工学研究会 (IE), Dec. 2024.
- [2] "Tensor Coreによる二次元畳み込みの高速化," 電子情報通信学会画像工学研究会 (IE), June 2025.
- [3] "Performance evaluation of image convolution with WebAssembly," International Workshop on Advanced Image Technology (IWAIT), Jan. 2023.

【お問合せ】

名古屋工業大学 産学官金連携機構

〒466-8555 名古屋市昭和区御器所町字木市29番

TEL:052-735-5627

E-mail: nitfair@adm.nitech.ac.jp

URL: <https://technofair.web.nitech.ac.jp/>