6C-02

Development of face detection application using 1000FPS high-speed camera

Joji Kuwabara† Toshiyuki Umeda† Masanori Ogawa† Jun Kubota† Michito Uekusa† Yasushi Mishima‡
Photron Ltd.† Hemibola Inc.‡

1. 緒言

我々は、最大 1000Hz の高速度でリアルタイム顔検出するアプリケーションを開発しました。

顔検出は、モニタリングシステムや自動制御システムなど様々な画像アプリケーションで使用される非常に一般的な画像処理ですが、その処理を 1000Hz の速度でリアルタイムに実行することは非常に困難です。

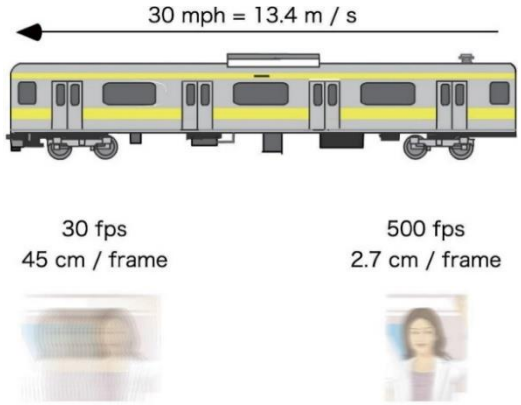

Fig.1 Passengers on high-speed trains.

例として、高速走行する列車に乗った乗客をホームなど側面から顔検出できるかどうかを考えてみましょう。

ある列車が時速 30 マイル（秒速 13.4 メートル）で駅を通過しています。通過する列車を 30FPS で撮影すると、1 フレームの間に顔が約 45cm 動きます。

30FPS 程度の低速のカメラでは対象がぼやけてしまい、顔検出に適した画像が得られないことは明らかです。もし対象物がそれほど速く動かないのであれば 30FPS でも問題ありません。しかし、高速で動くものを扱う場合は、十分ではありません。（Fig.1）

2. 高速画像処理システム

この問題を解決するためには、高 FPS で撮影可能なカメラデバイスと、高速な画像処理機構を組み合わせる必要があります。

今回、カメラデバイスとして、INFINICAM UC-1（株式会社フォトロン製）を使用しました。INFINICAM UC-1 は、1.2Mpixel で 1000FPS の撮影が可能なハイスピードカメラであり、カメラハードウェアでリアルタイムに画像データを圧縮して、制御 PC に転送する機能を持っています。

また、画像中の顔の位置を検出するために OpenCV を使用しています。OpenCV には様々な特徴検出技術がありますが、ここでは「Haar-like 特徴によるカスケード分類器」を使用します。

Fig.2 Data workflow using CPU

仮に、カメラデバイスによる撮影、PC へのデータ転送、画像処理、結果表示に至るまでの一連の工程を PC の CPU のみで実施したとします。

CPU の処理性能にも依存しますが、現時点での比較的入手しやすい PC の CPU では負荷が高すぎ、高速度でリアルタイムに顔検出することはできません。（Fig.2）

そこで今回開発したアプリケーションでは、カメラデバイスからの画像転送とその後の画像処理に工夫をしています。

まず、カメラデバイスから送られてくる圧縮データのデコード処理を GPU 上にて実行するようにしています。これにより、圧縮画像のデコード処理を CPU ではなく GPU に任せることで CPU の負荷低減ができるようになります。（Fig.3）

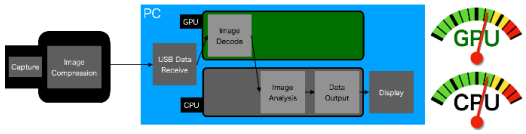

Fig.3 Data workflow using CPU and GPU

次にデコード処理した後の画像データの取り扱いについてです。

CPU にて画像処理を行う場合、OpenCV の処理を

行うために、画像を GPU メモリから CPU メモリに移動させる必要があります。高速かつ高解像度な動画像を扱う場合、データレートが非常に高いので、GPU から CPU へのデータ転送がパフォーマンス上の重要なボトルネックになる可能性があります。

すなわち、GPU でデコードすることで、ある程度の CPU パワーを解放することができますが、転送のオーバーヘッドを慎重に考慮する必要があります。 (Fig. 4)

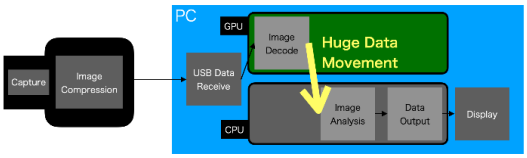

Fig. 4 Data movement from GPU to CPU

画像処理の高速化のため、OpenCV ではカスケード分類器の CUDA アクセラレーションバージョンを利用することが可能です。

実際、多くの OpenCV の関数が CUDA で高速化されたバージョンを持っています。デコードしたデータを GPU 上に保持し、GPU メモリ上のデータに対して GPU 上で OpenCV の処理を適用することで、理想的な性能を実現することができます。

GPU 上での OpenCV による処理結果は検出した顔の位置と大きさです。画像データに比較すると無視できるぐらい小さいサイズであるため、解析や表示のために CPU に送る時にボトルネックにはなりません。 (Fig. 5)

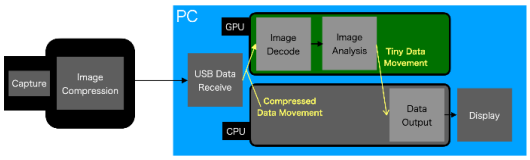

Fig. 5 Optimized workflows

3. 結果

Fig. 6 ではこの実験装置にて実際にリアルタイム画像処理をしたスクリーンショットを示しています。

30FPS 撮影では対象がぼやけてしまい正確に顔検出ができていないのに対して、600FPS 撮影では正確に顔検出ができています。

30FPS のカメラを使っていても、速いシャッタースピードがあれば、ブレの少ない鮮明な画像を得ることは可能です。しかし、高 FPS カメラは、30FPS のカメラよりも時間方向により多くのデータを取得することができます。Fig. 7 は、時間的な整合性を利用して、オプティカルフローを取得するためのポイントを特定するものです。

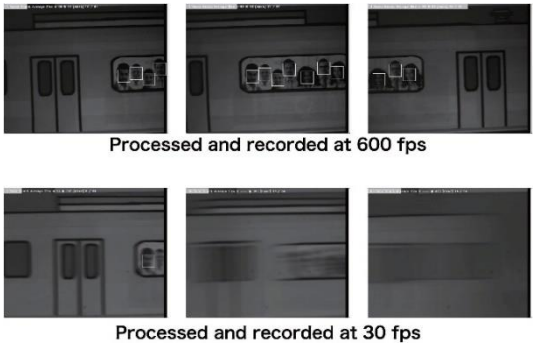

Fig. 6 Face detection result

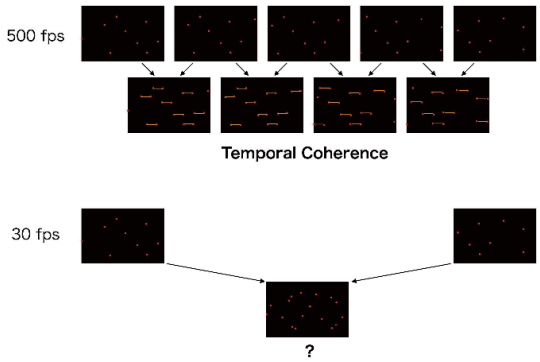

Fig. 7 Optical flow at different FPS

30FPS のカメラを速いシャッタースピードで使った場合、隣接するフレームからオプティカルフローを取得することは困難ですが、高 FPS カメラを使用する事でそれを可能にしています。

4. 結論

本論文では最大 1000Hz の高速度でリアルタイム顔検出するワークフローを示し、実際にそのアプリケーションを開発しました。

このアルゴリズムを単純に実装すると、デコード処理や CPU・GPU メモリ間のデータ転送に関するボトルネックが発生し、高速に移動する対象に対する画像処理ができません。

しかし、最初からパフォーマンスを意識してワークフローを設計すると、驚くようなパフォーマンスを発揮することが可能です。

画像処理が比較的軽いもの（例えば、閾値、cvtColor、integral などの単純な画像変換）であれば、CPU のみでも高速リアルタイム画像処理は可能ですが、「Haar-like 特徴によるカスケード分類器」のような処理を高速リアルタイムで行う場合、INFINICAM UC-1 のような高 FPS カメラと GPU を利用したワークフローが必要となることが確認できました。