

並列化シームカービングアルゴリズムの性能評価

木幡 康秀[†] 佐藤 健司[‡] 出崎 善久[‡][†]茨城大学大学院理工学研究科メディア通信工学専攻[‡]茨城大学工学部メディア通信工学科

1. はじめに

携帯端末等の普及に伴い、多様なアスペクト比のディスプレイが利用可能となっている。画像コンテンツを配信する場合、本来のアスペクト比と異なるディスプレイ上での表示に配慮する必要がある。典型的な例として、映画を家庭用のテレビに配信する場合は挙げられる。この場合、コンテンツの周辺にスペースを付加することにより、ディスプレイのアスペクト比にコンテンツを適合させることが多い(図1)。この手法では、単純なスケーリング(scaling)で生じるような画面の歪みは生じないが、画像はディスプレイに対して相対的に小さく表示されることになり、ディスプレイのサイズが小さい携帯端末では問題になる。



図1 アスペクト比の異なる画像

従来から知られているスケーリングやクロッピング(cropping)等のリサイズ手法では、人や物等の画像中の重要なオブジェクトに顕著な影響を与えるという問題がある。これらのアルゴリズムでは、画像中のオブジェクトを無視して画像に対する処理を一律に行うため、処理後の画像に歪みが生じる結果となっている。

近年盛んに研究されている画像の知的編集技術では、画像中のオブジェクトの視覚的重要度を意識して編集することを可能にしている。S. Avidan らによって提案されたシームカービング(seam carving)も、その一つである[1]。シームカービングにより、画像中のオブジェクトを保存したリサイズや特定のオブジェクトの削除が容易に行えるようになる(図2, 図3)。

シームカービングは汎用性の高いアルゴリズムであるが、実時間処理を要求するアプリケーションにおいては、処理時間が無視できないという問題が

ある。この問題を解決するために、文献[2]においてハードウェア指向のシームカービングアルゴリズムの枠組みを与え、設計した回路の一部をFPGA上に実装し評価した。本稿では、実画像に対してシームカービング処理可能な画像処理システムを完成させ、FPGA上に実装した場合の評価結果を示す。



(a) 元画像とシーム



(b) 縮小結果

図2 シームカービングによるリサイズの例

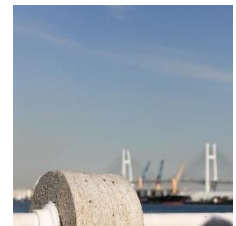


図3 オブジェクトの削除

2. シームカービングアルゴリズム

シームカービングアルゴリズムでは、シーム(seam)と呼ばれる画素集合を処理の単位として画像のリサイズを行う。画像の幅を変更する場合、シームを構成する画素は1行に1個のみ存在し、シーム全体は連続な離散曲線となっている。図2(a)にシームの例を示す。図2(a)中の赤い線がシームである。シームの削除、追加を繰り返すことにより、画像のリサイズが可能となる。

シームカービングアルゴリズムでは、シームを求める際に、適当なエネルギー関数 $E(x, y)$ により各画素 $I(x, y)$ に固有のエネルギー値を割り当てる。エネルギー関数は、エッジ周辺で高い値をとるものであればよく、文献[1]では式(1)で与えられる勾配強度を表す関数を使用している。本研究でも、式(1)をエネルギー関数として使用する。

$$E(x, y) = \left| \frac{\partial}{\partial x} I \right| + \left| \frac{\partial}{\partial y} I \right| \quad (1)$$

エネルギーの総和が低くなるような画素でシームを構成することにより、画像中のエッジ(オブジェクト)を避けてシームを選ぶことが可能になる。編集結果が目的に適合している限り、どのようなエネルギー関数を使用してもよい。また、一部の画素集合のエネルギーを直接指定して、特定のオブジェクトを削除することも可能である(図3)。

Performance evaluation of parallel processing for seam carving algorithm

[†] Yasuhide KOHATA, Major in Media and Telecommunications Engineering, Graduate School of Science and Engineering, Ibaraki University

[‡] Kenji SATO, Yoshihisa DESAKI, Department of Media and Telecommunications Engineering, Faculty of Engineering, Ibaraki University

3. シームカービング処理回路の設計

本研究で設計したシームカービング処理回路のブロック図を図4に示す。本設計では、入力画像から1ピクセル幅の縦方向のシームを探索し、シームを削除した（幅を1ピクセル縮小した）画像を出力するような仕様となっている。2ピクセル以上の幅で画像の縮小処理を行う場合は、図4の回路による処理を必要な回数繰り返せばよい。また、画像を縦方向に縮小する場合も、横方向に縮小する場合と同様の処理でリサイズすることが可能である。

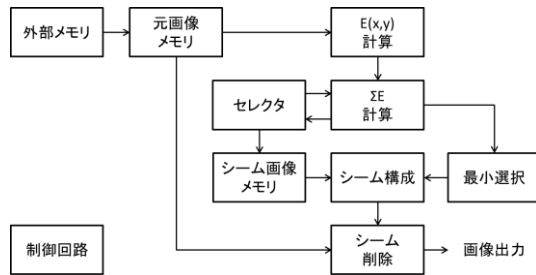


図4 シームカービング処理回路

- 外部メモリ上の処理対象画像を元画像メモリに読み込む。
 - $E(x,y)$ 計算回路で画素値の差分（勾配）をとり、画素ごとのエネルギー $E(x,y)$ を求める（図5(a)）。
 - $\Sigma E(x,y)$ 計算回路で、エネルギーの和 $\Sigma E(x,y)$ を求める（図5(b)）。
 - セレクトラでは、 $\Sigma E(x,y)$ の候補の中から最小のものを $\Sigma E(x,y)$ 計算回路に渡し、選択した $\Sigma E(x,y)$ の相対的な位置をシーム画像メモリに記憶する（図6）。
 - エネルギーの計算が全て終了した後、最小選択回路によって最終的な $\Sigma E(x,y)$ の候補の中からエネルギーが最小のものを選ぶ（図6）。
 - シーム構成回路では、最小選択回路から渡された $\Sigma E(x,y)$ の座標を始点としてシーム画像メモリ内のシーム画像をたどってゆき、シームを決定する（図6）。
 - 元画像からシームに対応する画素を削除し、画像を出力する
- また、制御回路により、回路全体の動作を制御する。

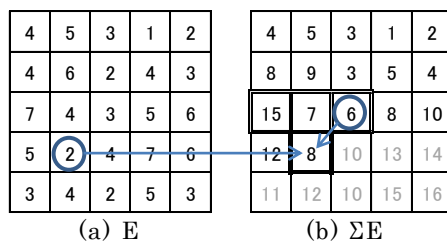


図5 エネルギーマップ

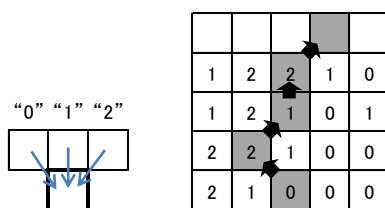


図6 シームの決定

アルゴリズム中のエネルギー（ E および ΣE ）計算は各画素に対して必要になるが、これらの処理に専用の回路を割り当てて並列化することが可能である。本設計では、入力画像の横1行分の画素に対するエネルギーの計算を並列処理することにより、処理時間の短縮を図っている。

4. 性能評価

シームカービングアルゴリズムを、今回設計した専用回路、及びソフトウェアで実装した場合の性能評価を行う。エネルギー計算の処理に要する時間を評価指標とし、それぞれの実装方式の処理性能を比較する。各実装方式の開発環境を以下に示す。

・開発環境

[専用回路]

HDL : Verilog HDL

開発環境 : ISE Design Suite 14.5 (Xilinx 社)

FPGA : Spartan-6, XC6SLX45 (Xilinx 社)

[ソフトウェア]

CPU : Intel Xeon W3565 (3.2GHz)

メモリ : 8GB

OS : Windows7 (64bit)

開発環境 : Microsoft Visual C++ 2010 Express

画像処理ライブラリ : OpenCV 2.4.2

回路の実装には Digilent 社製の FPGA ボード ATLYS を使用した。同ボードは、Xilinx 社製の Spartan-6 (XC6SLX45) を搭載し、100MHz のクロックで動作する。また、計 64MB のフラッシュメモリ（図4の外部メモリに対応）や FPGA 内の 2,088Kb のブロック RAM（図4の元画像メモリ、シーム画像メモリに対応）を利用することができる。640×480 ピクセルの画像を処理対象とした場合のエネルギーマップの処理時間を表1に示す。ソフトウェアによる処理時間は実測値、専用回路の処理時間は、シミュレーションにより求めたアルゴリズムの処理に要する消費クロック数と、ターゲットデバイスとなる FPGA のグローバルクロック周波数（100MHz）から求めた値である。

各実装方式の処理時間を比較すると、専用回路による処理はソフトウェアによる処理に比べて約 3,750 倍高速になっている。ソフトウェアによる処理では、エネルギー計算を1ピクセルごとに行う必要があるのに対して、専用回路では、画像の1行分（640ピクセル）の処理を並列に行っているため、処理時間を大幅に短縮することができた。

表1 エネルギーマップの処理時間

	処理時間[μs]
専用回路	約 4.8
ソフトウェア実装	約 18,000

参考文献

- [1] S. Avidan and A. Shamir, "Seam Carving for Content-Aware Image Resizing," SIGGRAPH, 2007.
- [2] 木幡康秀, 風間萌美, 八木澤良文, 出崎善久, "シームカービングアルゴリズムの FPGA 実装," 情報処理学会第75回全国大会, 2013.