「シームカービングの高速化と動画像への応用」

2013 年月日

京都産業大学コンピュータ理工学部ネットワークメディア学科
弘中　渉

要約

画像内容に応じた拡大縮小技術として「シームカービング」が知られている。シームカービング技術は、視覚的に重要でないシームと呼ばれる領域を削除することで、画像を縮小する技術である。シームを検出するために、動的計画法が使われる。動的計画法は最適なシームを検出するが、計算コストは小さくない。本論文では、シームカービングの考え方を動画像に適応することを目的として、準最適なシームを、動的計画法よりも高速に計算する方法を研究した。研究した第１の方法は、画像の１列目から最も削除に適した画素を選択し、その画素から局所的に下方向に削除に適した画素を選択する方法である。第１の方法において、最初の画素選択を全ての列で行うと、列数のシームを得る。第２の方法は、この中で、最も削除に適したシームを選択する方法である。この２つの方法を動的計画法と比較したとき、処理速度において、第１の方法は２倍ほど早くなり、第２の方法はほとんど同じであった。削除したシームのエネルギで縮小効果を比較すると、第１の方法は２～５倍悪くなり、第２の方法は１．３～２倍悪くなるという結果を得た。第１の方法は高速ではあるが、削除に適したシームを選択することができなかった。第２の方法は動的計画法と同程度の処理速度で、削除されるシームは若干劣っていた。

目次

第１章　序論　・・・４

第２章　シームカービングについて　・・・５
２．１　シームとは何か　・・・５
２．２　画像のエネルギ　・・・５
２．３　動的計画法　・・・７
２．４　動的計画法によるシーム検出　・・・８
２．５　計算時間の増加のオーダー　・・・１０

第３章　局所探索法について　・・・１１
３．１　局所探索法　・・・１１
３．２　局所探索法によるシーム検出　・・・１１
３．３　計算時間の増加のオーダー　・・・１５

第４章　実験と考察　・・・１６
４．１　環境設定　・・・１６
４．２　シームカービング処理の流れ　・・・１６
４．３　動的計画法と局所探索法の比較　・・・１６
４．４　考察と結論　・・・２３

第５章　結論　・・・２５
５．１　成果　・・・２５
５．２　課題　・・・２５

参考文献
謝辞
付録

１.序論

１．１　本研究の目的と背景

画像を拡大・縮小することは最も基本的な画像処理である。例えば、640×480画素の小さな画像を2560×1920画素の高精細ディスプレイに表示するためには、画像を4倍に拡大する必要がある。逆に、4000×3000画素のデジタルカメラで撮影した画像を、800×600 画素のスマートフォンのディスプレイに表示するためには、画像を1/5に縮小する必要がある。

最近の傾向として、大画面のTV、中画面のPC、小画面のスマートフォンのように、表示に用いるデバイスが多様になっている。その結果、それぞれの表示デバイス用に制作された画像や映像を、それとは異なったデバイスで見る機会が増えている。地上波デジタル放送の映像を大画面のTVで視聴するとともに、同じ映像を小画面のスマートフォンで視聴することは普通に行われている。そのような時、画面が小さいために、映像中の重要な部分を見にくいと感じることがある。利用が想定される全ての表示デバイスに合わせて、映像や画像をあらかじめ複数準備しておけば、最適な視聴が可能になる。しかし、これは実現が困難である。シームカービングという画像内容に応じた画像の拡大・縮小技術はこのような場合に有効な技術である。

画像を単純に拡大・縮小すると、全体的に同じ割合で拡大・縮小される。シームカービング処理では、背景などのそれほど重要でない部分から優先的に除去することで画像のサイズを小さくする。

シームカービングは 1 枚の画像を、内容に応じて拡大・縮小する技術である。これを動画像に応用する研究がある。動画像に応用すると、あらかじめ制作した１つの動画像を、内容に応じて適応的に拡大・縮小することができる。これによって異なる表示デバイス間での動画の視聴を改善することができる。しかし、シームカービングのアルゴリズムは計算コストが高い。シームカービングを動画像に応用するための重要な課題は、この計算時間を短縮することである。したがって、本論文ではシームカービングの処理を高速化できるアルゴリズムを研究することにした。

２．シームカービングについて

２．１　画像のシーム

シームの意味として縫い目や継ぎ目、合わせ目がある。本論文におけるシームの定義は、画像の上端から下端、あるいは、右端から左端を結ぶ連結した画像列である。このとき、シーム上で連続する画素は 8 連結で、かつシームの方向順に並んでいる。上端と下端を結ぶシームを「縦シーム」、右端と左端を結ぶシームを「横シーム」と呼ぶ。図 1 では縦シームの例を表示している。

シームカービングは、画像から検出したシームを削除して画像を縮小する処理である。シームを削除すると、その部分の画素がなくなる。縦シームを削除した場合、削除した画素の右隣の画素から右端までの画素を、1 つ左にずらせる。この処理によって、画像の横幅が 1 画素分小さくなる。この処理を繰り返すことで、任意の横幅に縮小することが出来る。縦方向の縮小についても同様である。

なお、シームカービングの原論文[5]では、シームを削除することによる画像の縮小だけでなく、シームを挿入することによる画像の拡大など、シームに基づくさまざまなリサイズ技術を論じている。本論文では、画像の縮小に関することだけを論じる。

図１．縦シームの例

２．２　画像のエネルギー

シームカービングでは、削除に適したシームを検出することが重要である。これを行うために、画像のエネルギという考えを導入する。基本的なアイデア

は、人間の目に敏感に感じる画素はそのエネルギが大きいと、と考えることである。いくつかの考え方があるが、今回は濃淡画像にラプラシアンフィルタをかけて、絶対値をとったものを画素のエネルギとして使用した。

図２に、濃淡画像にラプラシアンフィルタの絶対値を処理した画像を示す。ラプラシアンフィルタは空間２次微分を計算し、輪郭を強調するフィルタである。輝度の差分の変化量が大きくなっている部分を抽出する。ラプラシアンフィルタは、図３に示すフィルタ係数で、注目画素と８近傍の画素の２次差分を取る。この値の絶対値が大きいほど注目する画素のエネルギ値が大きくなり、小さいほどエネルギ値が小さくなる。この計算を全画素に対して行うことで、重要な画素が大きい値をとるエネルギマップを生成する。

図２．濃淡画像(左)にラプラシアンフィルタをかけた画像(右)

1	1	1
1	-8	1
1	1	1

図３．８近傍のラプラシアンフィルタ演算における各ピクセルの係数(重み)

２．３　動的計画法

原論文では、削除するのに最適なシームを検出するために動的計画法を用いる。最適なシームとは、シームを構成する画素に対するエネルギの和が最小になるものである。まず、動的計画法について一般的な説明を行い、続いて、シーム検出に用いる場合の手順について説明する。動的計画法とは、１つ１つ段階を踏んで最も適した答えを導き出す場合に、繰り返し現れる部分的な最適解を表にして記憶させる方法である。ここでは、２変数関数の和を最大化する問題を例として説明する。

$$J = f_1(x_1) + h_1(x_1, x_2) + h_2(x_2, x_3) + \cdots\cdots + h_{n-1}(x_{n-1}, x_n) \rightarrow max \quad \cdots (1)$$

式(１)で、x_1 に注目する。これは、$f_1(x_1)$ と $h_1(x_1, x_2)$ にしか関係しない。したがって、$f_1(x_1) + h_1(x_1, x_2)$ を最大にするように x_1 を選べばよい。しかし、そのためには x_2 の値が分かっている必要がある。そこで、x_2 の可能な全ての値に対して、最適な x_1 をそれぞれ計算する。これを x_2 の関数とみなして $x_1(x_2)$ と書く。その $f_1(x_1) + h_1(x_1, x_2)$ の最大値も x_2 に依存するので、それを x_2 の関数とみなして、

$$f_2(x_2) = max\,[f_1(x_1) + h_1(x_1, x_2)]$$

と定義する。これを用いると、式 J は次のように書ける。

$$J = f_2(x_2) + h_2(x_2, x_3) + \cdots\cdots + h_{n-1}(x_{n-1}, x_n) \rightarrow max$$

これは元の式と同じ形であり、かつ変数の数が１つ減っている。この操作を繰り返して行うと最終的に関数 $f_n(x_n)$ を

$$f_n(x_n) = \max_{x_{n-1}}\,[f_{n-1}(x_{n-1}) + h_{n-1}(x_{n-1}, x_n)]$$

と定義して、最大値を与える x_{n-1} を $x_{n-1}(x_n)$ と置く。そうすると元の式は、次のように書ける。

$$J = f_n(x_n) \rightarrow max$$

この１変数関数 $f_n(x_n)$ が最大になる x_n の値を x_n^* とする。これに対する最適な x_{n-1} の値は x_{n-1}^* は $x_{n-1}^* = \hat{x_{n-1}}(x_n^*)$ である。以下、逆をたどって x_{n-2} の最適な値 $x_{n-2}^* = \hat{x_{n-2}}(x_{n-3}^*)$ ，… ， x_1 の最適な値 $x_1^* = \hat{x_1}(x_2^*)$ が決定できる。上の手順を一般的に書くと、次のように書ける。

$$J = f(x_1, x_2, \ldots, x_n) \rightarrow max$$

２．４　動的計画法によるシーム検出

動的計画法を用いたシームの検出は次のようになる。図４を参照しながら説明する。

Step1：入力画像のラプラシアンの絶対値を計算する。これをエネルギマップと呼ぶ。

Step2：エネルギマップを、動的計画法のアルゴリズムが計算に用いる表(S 表(シームのエネルギを計算する表という意味)とよぶことにする）にコピーする。

Step3：第２行目の左端の画素から右端の画素まで順に Step4 の操作を行う。

Step4：注目画素の１行上の３つの近傍画素について、対応するＳ表の値が最も小さいものと注目画素のＳ表の値を足し合わせ、注目画素のＳ表の値を更新する。

Step5：Step3、Step4 と同様の処理を第３行目から最下端の行まで繰り返す。

Step6：最下端のＳ表で値が最小となるものを検出する。その値は、全シームの中で、シームを構成する画素のエネルギの和の最小である。

Step7：Step6 で検出した S 表の位置から、S 表を上方向に値が小さい位置を繋ぐことでシームを検出する。

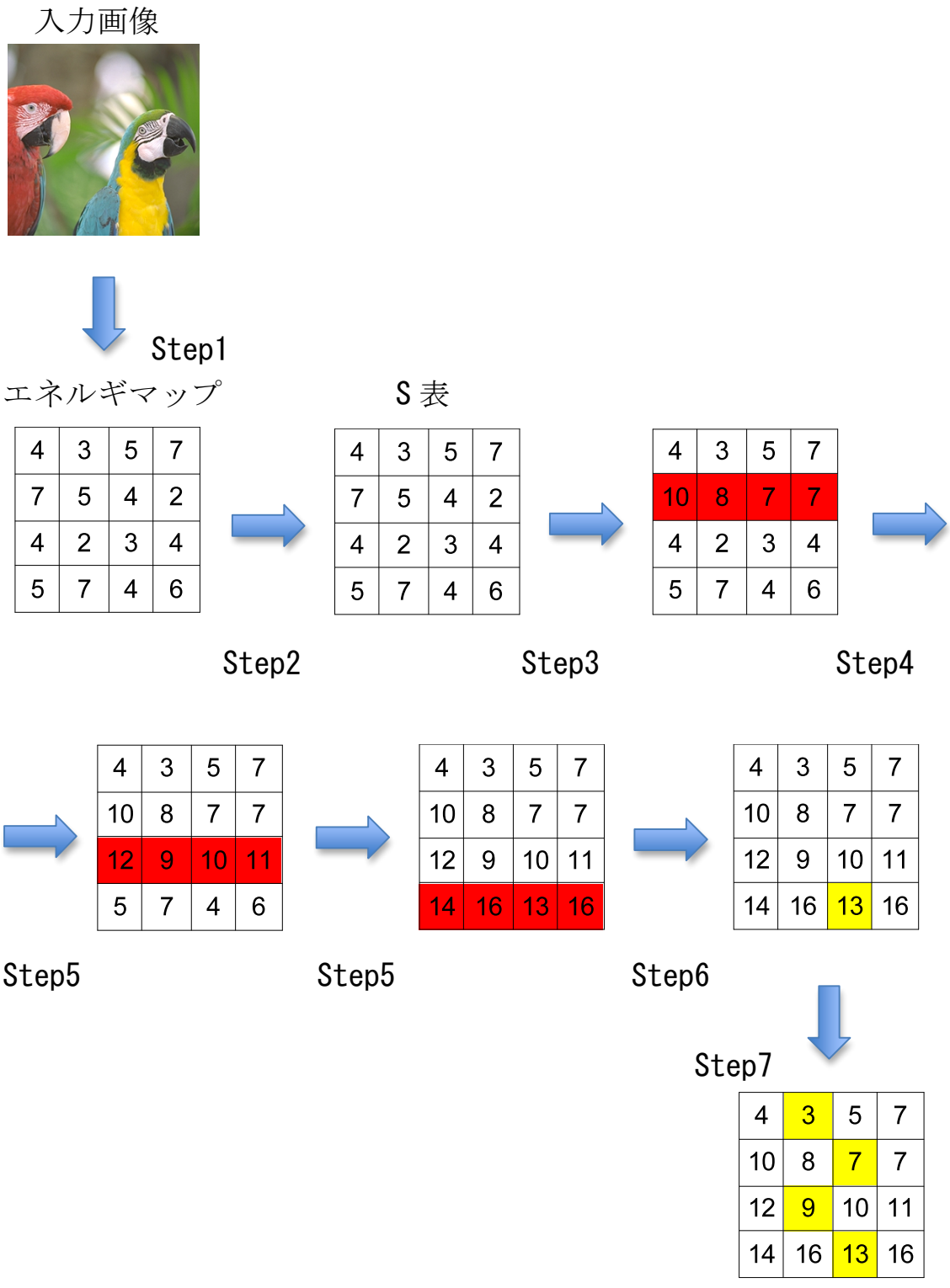


図４．動的計画法によるシーム検出の手順

検出したシームを削除すると、その箇所が空くので、詰めて１画素分縮小された画像にする。その処理を削除するシームの本数繰り返して行うことにより、サイズを縮小する。動的計画法は重複計算を省いた高速なアルゴリズムである。シームカービングは多くのシームを削除するため、シームを高速に検出する必要がある。動的計画法は、エネルギが最小のシームを計算する最も早いアルゴリズムである。

２．５　動的計画法の計算時間のオーダー

アルゴリズムの解析では、問題のサイズが大きくなるにしたがって、計算時間がどのように増加するかを解析する必要がある。また、それの解析結果を漸近記号を使って表現する。その記号には、θ 記号、O 記号、Ω 記号、o 記号などの種類がある。これらの記号を用いて、時間計算量の評価を簡潔に記述することをオーダー表記と言う。

N×N の画像に対して、各行ごとに左端の画素から右端の画素まで、注目画素の１行上の３つの近傍画素のエネルギを比較し、最小のものと注目画素のエネルギを足し合わせている。その計算量は３N である。この処理を第２行目から最下端の行まで行うので、繰り返し回数は N-１回である。これらを掛け合わせることで、１本のシームを検出するための計算量を得る。１本のシームを計算するのに要する計算時間は、$3N^2-3N$ であり、オーダーは $\theta(N^2)$ になる。

３．準最適で高速なシームの検出アルゴリズム

３．１　準最適なアルゴリズム

本研究の目標は動画像にシームカービング処理を適用することである。動画像に対するシームカービング処理は、画像の場合と比べて、計算時間が多くかかる。そのため、少しでも計算時間を早くすることが望ましい。動的計画法は、シームの最適解を最も早く計算するアルゴリズムであるが、最適解を求めているため、計算コストが高い。そこで、シームの最適性をある程度犠牲にして、計算コストが小さい準最適なアルゴリズムの開発に取り組んだ。

動的計画法は、シームの計算において、全ての場合を網羅的に計算する最も効率的なアルゴリズムである。したがって、本研究で取り組むアルゴリズムは網羅的に探索することを犠牲にせざるを得ない。準最適なアルゴリズムは局所的な探索を取り入れたものになる。局所探索法は、近似アルゴリズムの中で最も単純なアルゴリズムである。またその中でグリーディなアルゴリズムでかつ、ヒューリスティックな手法である。動的計画法では最適解を求めることができるのに対して、局所探索法では近似解を求める。そこで少しでも最適解に近い近似解を求めるために、シームカービングの処理に合わせていくつかの方法を考案した。

３．２　局所探索法によるシーム検出

今回、局所探索のアルゴリズムを２つ考案し、実験した。図５を用いて方法１(以後「単純局所探索法」とよぶことにする)によるシーム検出を説明する。また、図６を用いて方法２(以後「繰り返し局所探索法」とよぶことにする)によるシーム検出を説明する。

単純局所探索法

Step1：入力画像のラプラシアンの絶対値を計算し、それをエネルギマップとする。

Step2：エネルギマップを単純局所探索法のアルゴリズムが計算に用いるＳ表にコピーする。

Step3：第１行目で対応するＳ表の値が最も小さいものを検出する。

Step4：Step3 で検出した S 表の位置から、S 表の下方向の３つの近傍画素で値が小さい位置を繋ぐことでシームを検出する。

Step5：Step4 の処理を最下端の行まで繰り返す。

繰り返し局所探索法

Step1：入力画像のラプラシアンフィルタの絶対値を計算し、それをエネルギマップとする。

Step2：エネルギマップを繰り返し局所探索法のアルゴリズムが計算に用いる S 表にコピーする。

Step3：第１行目から最下端の行まで Step4～Step6 を繰り返す。

Step4：注目する行で対応する S 表の値が小さいものを検出する。

Step5：Step4 で検出した S 表の位置から、S 表の下方向と上方向の３つの近傍画素で値が小さい位置をつなぐことでシームを検出する。

Step6：Step5 の処理を、シームが両端に達するまで繰り返す。

Step7：このようにして検出した全シームの中で、エネルギが最も小さいものを選択する。

入力画像

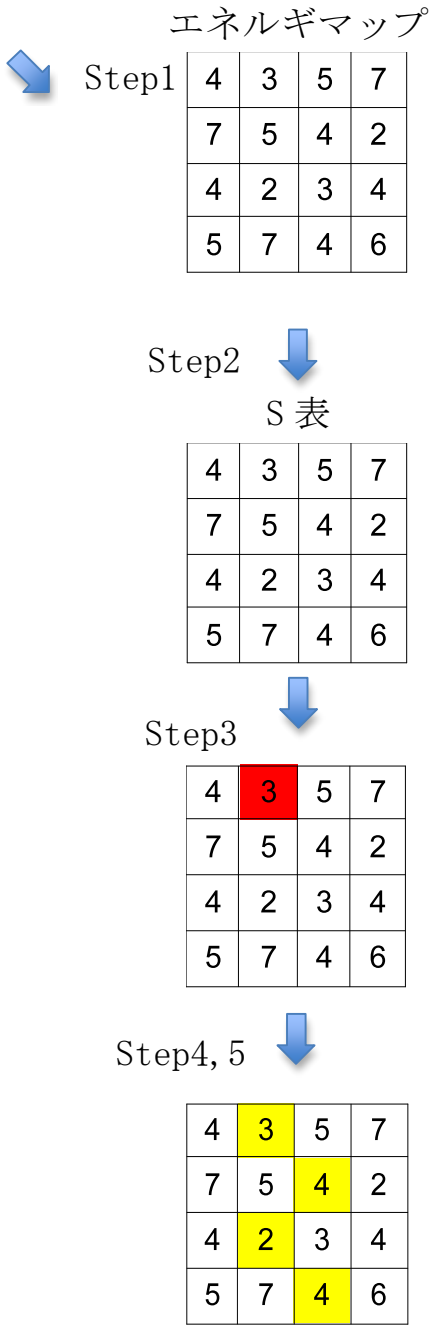


図５. 単純局所探索法によるシーム検出

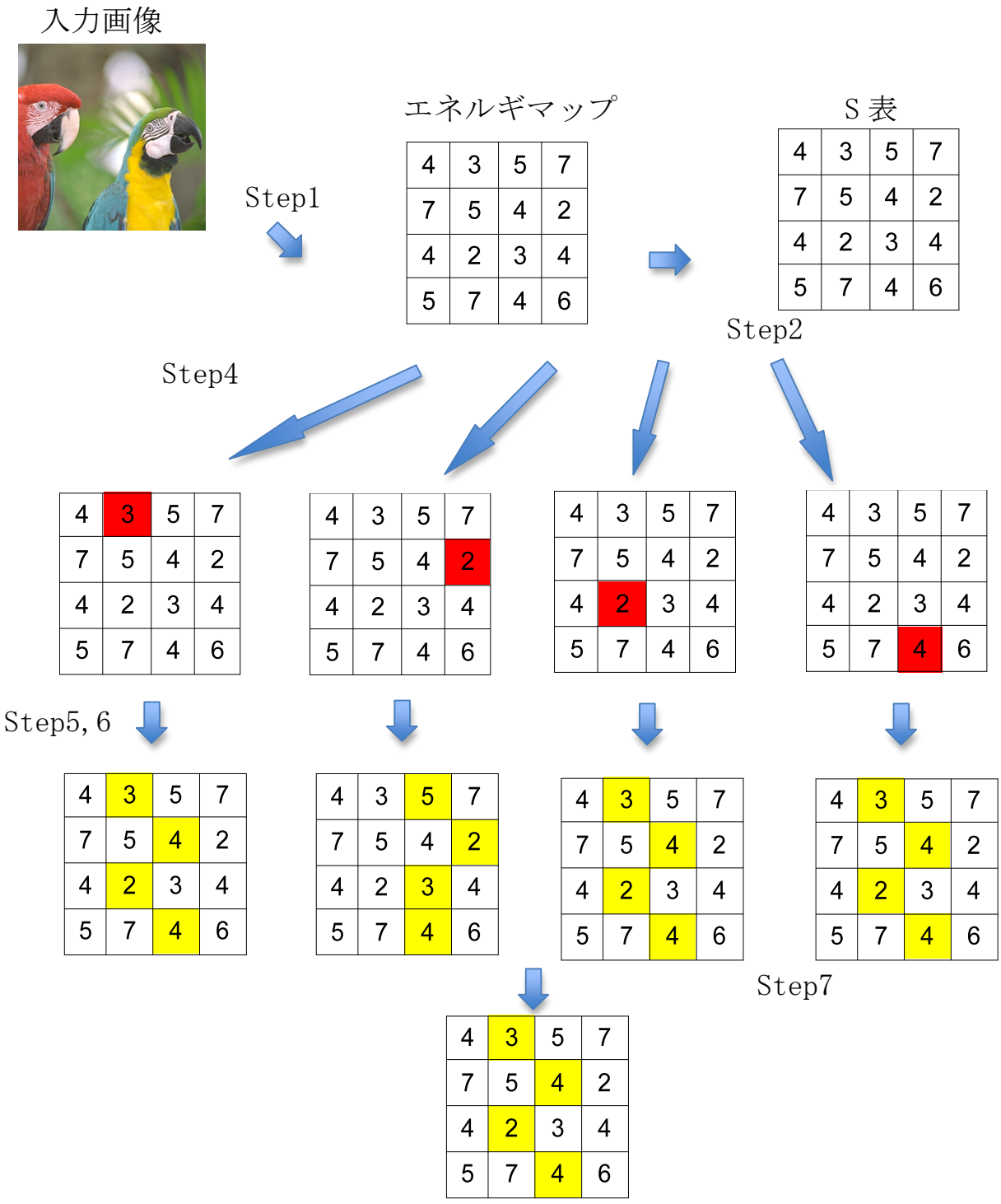


図６．方法２によるシーム検出手順

３．３　提案手法の計算時間の増加のオーダー

先ほど動的計画法によるシームの計算時間のオーダーは、$\theta(N^2)$ であった。提案手法の計算時間のオーダーを計算する。N×N の画像に対して、単純局所探索法の場合、第１行目で対応する S 表の値が最も小さいものを検出するのに要する計算量は N である。検出した位置から３つの近傍画素で値が小さい位置を最下端の行までつなぐのに要する計算量は 3(N-1)である。これらを足し合わせることにより１本のシームを検出することができる。１本のシームを計算するのに要する計算時間は 4N-3 であり、オーダーは $\theta(N)$ になる。

繰り返し局所探索法の場合、単純局所探索法の計算時間に N を掛けることにより N 本のシームが検出されて、その中でシームを構成する画素のエネルギ和が最小のシームだけを検出する。１本のシームを計算するのに要する計算時間は $4N^2-3N$ であり、オーダーは $\theta(N^2)$ になる。

したがって、単純局所探索法は動的計画法よりも高速である。繰り返し局所探索法は動的計画法と同じオーダーである。

４．実験と考察

４．１　環境設定

研究を行うためのプログラム開発環境に Visual Studio 2010 を用いた。プログラム言語に C++を用いた。画像入出力や画像処理を行うためのライブラリとして OpenCV2.2 を用いた。パソコンの OS は Windows 7、CPU は Intel(R) Xeon(R) で、クロック周波数 2.40GHz、主メモリ 4.0GB のものを用いた。

４．２　シームカービング処理の流れ

以下で行う実験では、カラー画像を入力し、指定した本数のシームを削除することで、横方向に縮小した画像を生成する。全体の処理の流れは、次のようになる。

Step1：カラー画像をグレー画像に変換する。
Step2：グレー画像のラプラシアンの絶対値を計算し、エネルギマップを作成する。
Step3：エネルギマップから動的計画法あるいは提案手法を用いて、１本のシームを計算する。
Step4：もとのカラー画像の R 成分画像、G 成分画像、B 成分画像から、シームを削除し、横方向を１画素分縮小する。
Step5：指定した本数のシームを削除するまで、Step1～Step4 を繰り返す。

４．３　動的計画法と提案手法の比較

動的計画法と提案手法の違いを比較する。11 枚の 256×256 の画像に対して 30 シームと 60 シームを削除するのに要した時間を表 1、表 2 に示す。11 枚の画像の始めの 10 シームのエネルギ値の比較を表 3、表 4、表 5 に示す。１シーム目から 120 シーム目までのシームを構成する画素のエネルギ和の比較を図 8 に示す。画像サイズを変更しての処理速度の比較を表 6、図 9 に示す。動画像に対して、3 つの方法で 200 シームを削除するのに要した時間を表 7 に示す。11 枚の元画像と、それらを 3 つの方法で 30 シーム、60 シーム削除した画像と、元画像を単純に 30 シーム、60 シーム縮小した画像を付録 a に示す。なおシームの計算時間は、4.2 における Step1～Step4 の計算時間の和である。

11 枚の画像を動的計画法、提案手法の単純局所探索法、繰り返し局所探索法により 30 シームと 60 シームを削除するのに要した時間を表 1 に示す。処理速度の単位は[ms]である。

表１．動的計画法と提案手法で 30 シーム削除したときの処理速度の比較

	動的計画法 [ms]	単旬局所探索法 [ms]	繰り返し局所探索法 [ms]
Aerial.bmp	1010	520	1074
Airplane.bmp	1019	535	1067
Balloon.bmp	1010	519	1056
couple.bmp	999	522	1052
Earth.bmp	1003	519	1040
Girl.bmp	1015	514	1060
Lenna.bmp	1015	519	1070
Mandrill.bmp	1021	515	1060
milkdrop.bmp	1024	519	1080
Parrots.bmp	1024	515	1035
Pepper.bmp	1025	522	1052

表２．動的計画法と提案手法で 60 シーム削除したときの処理速度の比較

	動的計画法 [ms]	単純局所探索法 [ms]	繰り返し局所探索法 [ms]
Aerial.bmp	1919	988	2035
Airplane.bmp	1912	972	1997
Balloon.bmp	1907	987	1982
couple.bmp	1894	972	2002
Earth.bmp	1894	972	1962
Girl.bmp	1897	972	1987
Lenna.bmp	1903	987	1980
Mandrill.bmp	1904	982	2015
milkdrop.bmp	1961	987	1992

Parrots.bmp	1897	987	1989
Pepper.bmp	1887	972	2012

次に、動的計画法と提案手法で削除した始めの 10 シームのエネルギ値を表 3～表 5 に示す。

表 3．動的計画法で削除した始めの 10 シームのエネルギ値表

動的計画法	1	2	3	4	5	6	7	8	9	10	平均
Aerial.bmp	4806	5074	5282	5374	5560	5694	5762	5830	6078	6322	5578.2
Airplane.bmp	1534	1694	2008	2160	2278	2292	2364	2464	2692	2695	2218.2
Balloon.bmp	436	466	472	530	554	568	570	628	646	652	552.2
couple.bmp	350	404	432	484	512	560	602	710	778	856	568.8
Earth.bmp	1564	1632	1798	1894	1956	2078	2166	2182	2282	2394	1994.6
Girl.bmp	792	88	922	958	978	1012	1028	1086	1140	1206	1000.2
Lenna.bmp	780	1020	1204	1286	1318	1378	1480	1526	1632	1690	1331.4
Mandrill.bmp	4088	4868	5482	5998	6180	6235	6534	6672	6850	7274	6018.1
milkdrop.bmp	774	848	1103	1223	1326	1374	1380	1392	1442	1568	1243
Parrots.bmp	498	552	618	626	658	698	712	776	776	782	669.6
Pepper.bmp	1754	1838	1964	2136	2196	2306	2354	2446	2450	2476	2192

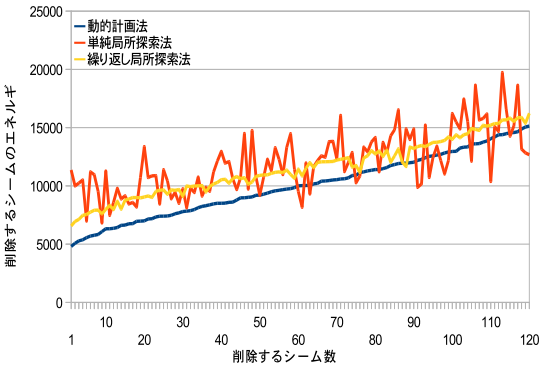
表４．単純局所探索法で削除した始めの 10 シームのエネルギ値表

単純局所探索法	1	2	3	4	5	6	7	8	9	10	平均
Aerial.bmp	1136	1000	1023	1050	6986	1120	1096	9538	6823	1127	9889.9
Airplane.bmp	7984	7444	3292	5445	5423	7543	8159	8880	5384	3748	6330.2
Balloon.bmp	910	1038	1376	1192	1160	2110	2094	2038	2150	2558	1662.6
couple.bmp	1656	1802	2378	2312	4711	1374	1496	1702	1206	7176	2581.3
Earth.bmp	2964	7750	4568	3966	4448	4486	4966	3466	4268	3938	4482
Girl.bmp	1244	1184	1408	1190	2412	5693	5400	7557	5664	5494	3724.6
Lenna.bmp	2678	2430	3422	1850	2074	7312	8213	1441	5027	5168	5259.1
Mandrill.bmp	1694	9232	1018	1244	1204	1090	1220	8835	5680	1288	11134.
milkdrop.bmp	2046	3742	4624	2836	5019	3296	2018	1070	1612	1898	2816.1
Parrots.bmp	3704	3064	5083	8712	7320	4227	802	756	968	858	3091.4
Pepper.bmp	4658	7779	6460	5852	5784	4714	6544	5359	4124	2516	5379

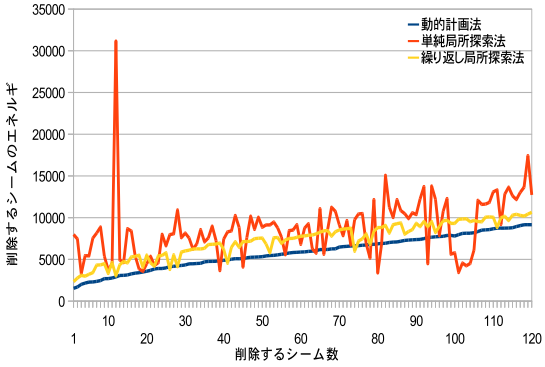
表５．繰り返し局所探索法で削除した始めの 10 シームのエネルギ値表

繰り返し局所探索法	1	2	3	4	5	6	7	8	9	10	平均
Aerial.bmp	6563	6944	7123	7466	7550	7748	7910	7924	7598	7984	7484
Airplane.bmp	2330	2768	3090	2982	3216	3409	4320	4339	4483	3314	3425.1
Balloon.bmp	814	852	1070	1234	992	1092	1434	1386	1342	1394	1161
couple.bmp	988	1090	1156	1218	948	1170	1442	1580	1566	1616	1277.4
Earth.bmp	2742	2920	2810	3380	3248	3598	3464	3780	3486	3796	3322.4
Girl.bmp	1038	1152	1152	1188	1194	1248	1272	1356	1392	1418	1241
Lenna.bmp	988	1204	1448	1682	1926	2086	2118	2366	2394	2660	1887.2
Mandrill.bmp	5248	6612	6960	7104	8017	8849	8863	9301	9072	9452	7951.1
milkdrop.bmp	930	1580	1782	1812	1862	1986	2042	2054	2100	2292	1844
Parrots.bmp	788	822	822	834	970	944	1022	1014	1190	1140	954.6
Pepper.bmp	2973	2898	3202	3118	3379	3614	3765	3291	2919	3774	3293.3

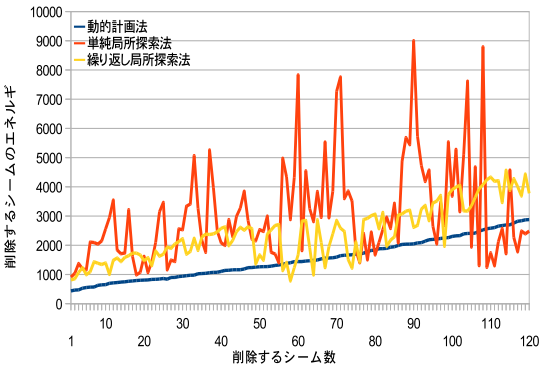
次に11枚の画像の120シームまでのエネルギ値を折れ線グラフを用いて比較する。折れ線グラフの縦軸は削除されるシームのエネルギで、横軸は削除するシーム数である。折れ線グラフの青色が動的計画法を用いて、計算したシーム。折れ線グラフの赤色が単純局所探索法を用いて、計算したシーム。折れ線グラフの黄色が繰り返し局所探索法を用いて、計算したシームである。図８にて比較を行う。



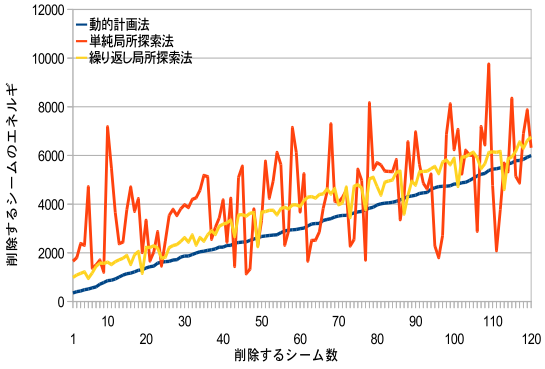
(a)Aerial.bmp



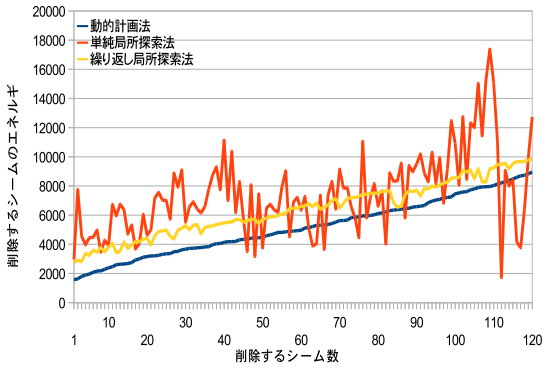
(b)Airplane.bmp



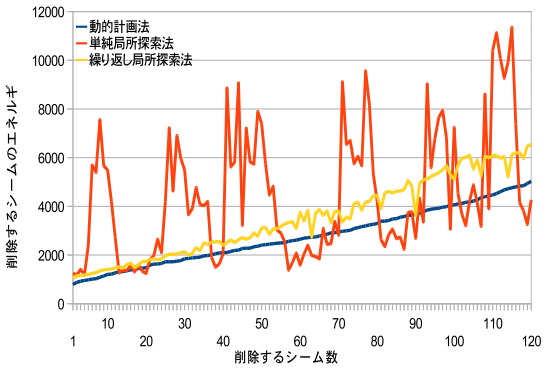
(c)Balloon.bmp



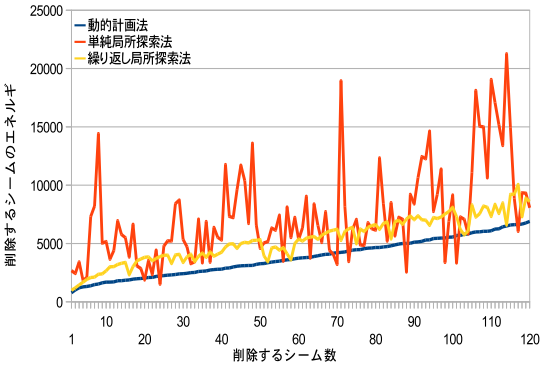
(d)couple.bmp



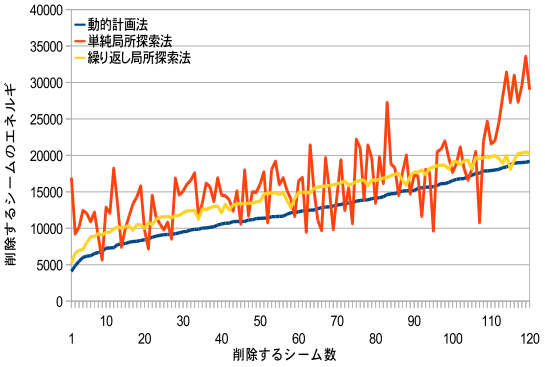
(e)Earth.bmp



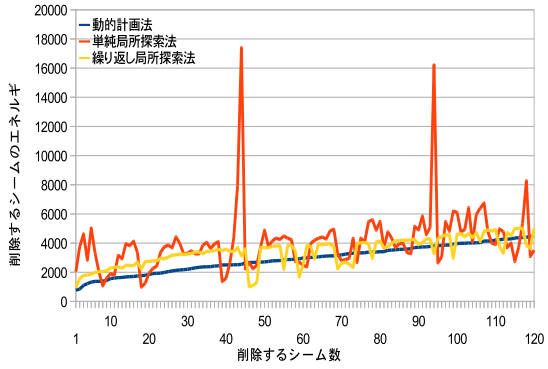
(f)Girl.bmp



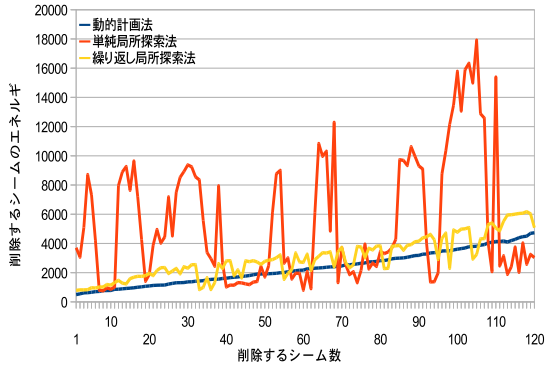
(g)Lenna.bmp



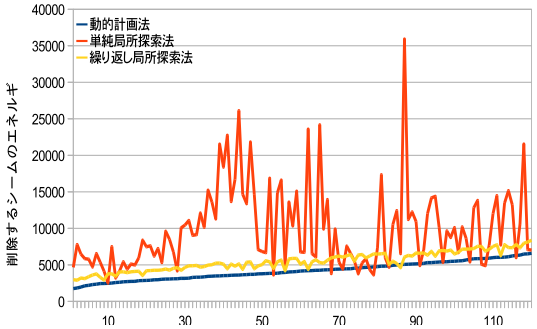
(h)Mandrill.bmp



(i)milkdrop.bmp



(j)Parrots.bmp



(k)Pepper.bmp

図８．11 枚の画像の 1 シーム目から 120 シーム目までのシームを構成する画素のエネルギ値を表した折れ線グラフ

ここまでの実験の画像サイズは全て、256×256 であった。次いで、画像のサイズを 16、32、64、128、256、512、1024 の７種類の同じ画像で１０シーム削除したときに要した時間の比較を行い、表６と図９に表記する。今回は先ほどの１１枚の画像の中の Girl.bmp の画像で行った。時間の単位は[ms]である。

表６．画像サイズを変更したときの処理速度の比較

Girl.bmp	動的計画法［ms］	単純局所探索法［ms］	繰り返し局所探索法［ms］
16×16	20	6	10
32×32	20	10	10
64×64	30	18	30
128×128	100	55	90
256×256	350	190	360
512×512	1386	732	1432
1024×1024	5488	2916	5506

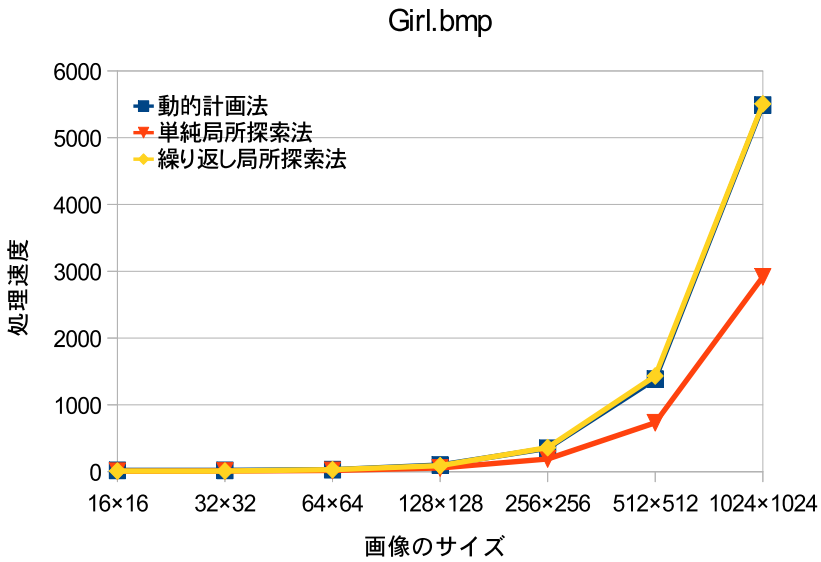


図９．画像サイズを変更したときの処理速度の比較の折れ線グラフ

次に動画像に対して、動的計画法、単純局所探索法、繰り返し局所探索法の 3 つの方法で 200 シーム削除するのに要した時間を表 7 に示す。動画像のサイズは 720×480、長さは 10 秒、データ速度は 987kbps、総ビットレートは 1115kbps、フレームレートは 29.97fps である。拡張子は.avi でシームカービング処理を行う。

表 7．720×480 の 10 秒の動画に対して、200 シーム削除した時の処理速度の比較

basketball.avi	動的計画法 ［秒］	単純局所探索法 ［秒］	繰り返し局所探索法 ［秒］
200		4820	

４．４　考察と結論

表 1 と表 2 から、処理速度は単純局所探索法が動的計画法と比べて 2 倍ほど良い結果となった。繰り返し局所探索法はあまり変わらない結果となり、また画像による処理速度の違いはあまり見られない。

表 3～表 5 から、始めの 10 シームのエネルギ値の平均を、単純局所探索法と動的計画法とを比べると、2～5 倍悪い結果となった。単純局所探索法では最上端の行しか注目していないのであまり良い結果が出なかった。繰り返し局所探索法は動的計画法と比べると、1.2～2.2 倍悪い結果となった。始めの 10 シーム

だけを見ると動的計画法に比べて、やや悪い結果となった。

図 8 の折れ線グラフを見ると、動的計画法のグラフは綺麗な右肩上がりとなっている。単純局所探索法のグラフは最上端の行にしか注目していないので、削除するシーム毎のエネルギ値は不規則となっている。繰り返し局所探索法のグラフは動的計画法と似たグラフとなっている。しかし、少しずつであるが、動的計画法より大きい値のシームを抽出していて、順最適解を求めていると分かる。

表 6 と図 9 を見ると、単純局所探索法でのオーダーは $\theta(N)$ なので、計算時間は直線的に増えるはずが、カーブ描いてオーダーが $\theta(N^2)$ と同じグラフをしている。単純局所探索法のプログラムで何か余計な処理が含まれている可能性がある。

５．結論

５．１　成果

動的計画法と単純局所探索法、繰り返し局所探索法によるシームカービング処理での違いは単純局所探索法では、元画像と少し違う印象を持ったが、繰り返し局所探索法では画像を見る限りあまり感じない。しかし、実験結果としてシームのエネルギ値最適解を導くことが出来る動的計画法に比べて、単純局所探索法は2～5倍ほど悪くなり、繰り返し局所探索法は1.2～2.2倍ほど悪い結果となった。処理速度は動的計画法と比べて、単純局所探索法は 2 倍ほど良くなり、繰り返し局所探索法はあまり変わらない結果となった。研究では今回、解の正確性よりも処理速度に重点をおいていたので、あまり良い結果とはならなかった。

今回、動画像へのシームカービング処理を行った。シームカービング処理後の動画像を見比べると、動的計画法での処理後の動画は不自然なずれをあまり検出されなかったのに対して、単純局所探索法では大きなずれを、繰り返し局所探索法では小さなずれを確認できた。

５．２　課題

課題として、処理速度が動的計画法より処理速度が早い単純局所探索法の解を処理速度を維持したまま、最適解に近づける。もしくは、準最適解を導き出せる繰り返し局所探索法の処理速度を準最適解のままで処理速度を早める必要がある

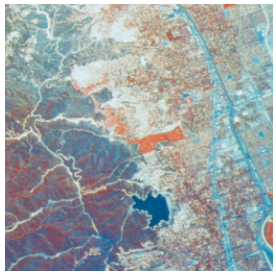
今回の研究では、1 本のシームを毎回除去していたが、それでは毎回エネルギーマップを作り直す必要があり、そのぶん時間がかかってしまう。そこで初めに作成したエネルギーマップから全てのシームを除去できるような新しいプログラムの作成が今後必要になってくると考えられる。また、今回はラプラシアンフィルタをかけたエネルギーマップからシームの検出を行ったが、違うフィルタをかけてシームを検出すると、結果はどう変わるのかなどを比較する必要がある。

参考文献

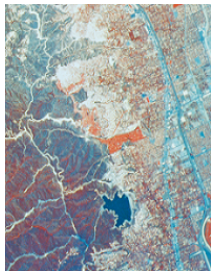
[1] 山崎俊彦，「100 行で書く画像処理最先端　勾配ベースの画像編集」PoissonImage Editing，映像情報メディア学会誌, Vol.64, No.5, pp.729-737, 2010 年.

[2] Shai Avidan, Ariel Shamir, “Seam Carving for Content-Aware Image Resing, “ ACM Trans.on Graphics, Vol. 26, Issue 3, Article No. 10, July 2007.

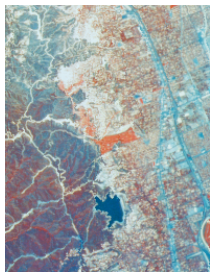
付録 a



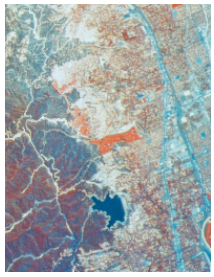
(a)元画像

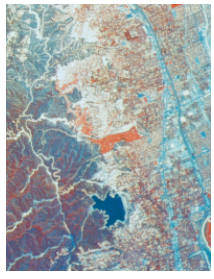
(b)単純縮小画像(左：30 シーム、右：60 シーム)

(c)単純局所探索法（左：30 シーム、右：60 シーム)

(d)繰り返し局所探索法(左：30 シーム、右：60 シーム)

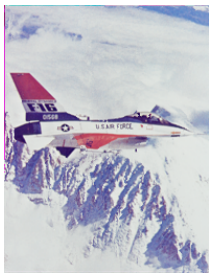



(e)動的計画法(左：30 シーム、右：60 シーム)

図 a.1 Aerial.bmp の処理結果

(a) 元画像


(b)単純縮小画像(左：30 シーム、右：60 シーム)

(c)単純局所探索法(左：30 シーム、右：60 シーム)


(d)繰り返し局所探索法(左：30 シーム、右：60 シーム)


(e)動的計画法(左：30 シーム、右：60 シーム)

図 a.2 Airplane.bmp の処理結果

(a)元画像


(b)単純縮小画像(左：30 シーム、右：60 シーム)


(c)単純局所探索法(左：30 シーム、右：60 シーム)


(d)繰り返し局所探索法(左：30 シーム、右：60 シーム)


(e)動的計画法(左：30 シーム、右：60 シーム)

図 a.3 Balloon.bmp の処理結果

(a)元画像

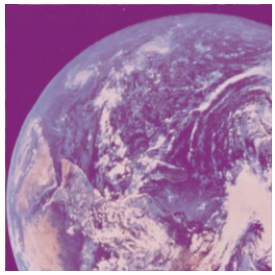

(b)単純縮小画像(左：30 シーム、右：60 シーム)


(c)単純局所探索法(左：30 シーム、右：60 シーム)

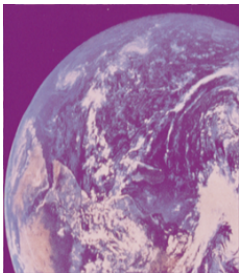
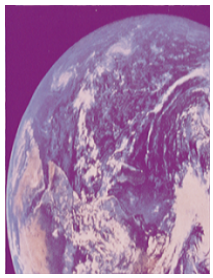

(d)繰り返し局所探索法(左：30 シーム、右：60 シーム)


(e)動的計画法(左：30 シーム、右：60 シーム)

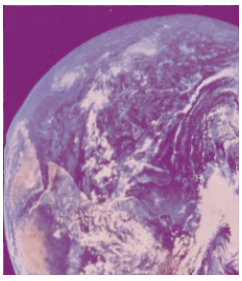
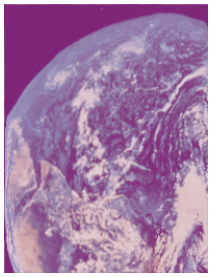
図 a.4 couple.bmp の処理結果



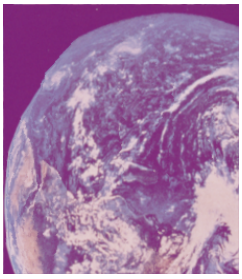
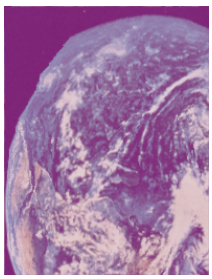
(a)元画像

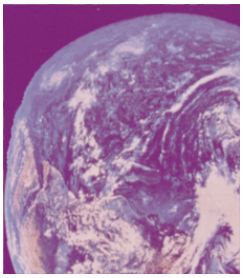
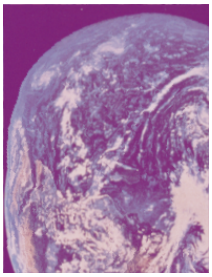
(b)単純縮小画像(左：30 シーム、右：60 シーム)

(c)単純局所探索法(左：30 シーム、右：60 シーム)

(d)繰り返し局所探索法(左：30 シーム、右：60 シーム)

(e)動的計画法(左：30 シーム、右：60 シーム)

図 a.5 Earth.bmp の処理結果

(a)元画像


(b)単純縮小画像(左：30 シーム、右：60 シーム)


(c)単純局所探索法(左：30 シーム、右：60 シーム)

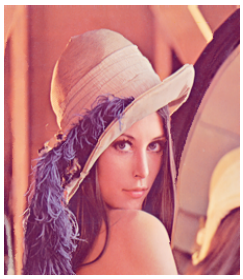

(d)繰り返し局所探索法(左：30 シーム、右：60 シーム)


(e)動的計画法(左：30 シーム、右：60 シーム)

図 a.6 Girl.bmp の処理結果

(a)元画像


(b)単純縮小画像(左：30 シーム、右：60 シーム)




(c)単純局所探索法(左：30 シーム、右：60 シーム)

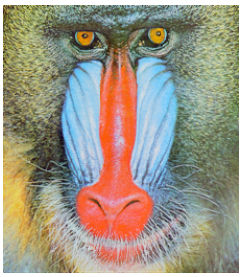

(d)繰り返し局所探索法(左：30 シーム、右：60 シーム)


(e)動的計画法(左：30 シーム、右：60 シーム)

図 a.7 Lenna.bmp の処理結果

(a)元画像


(b)単純縮小画像(左：30 シーム、右：60 シーム)




(c)単純局所探索法(左：30 シーム、右：60 シーム)


(d)繰り返し局所探索法(左：30 シーム、右：60 シーム)


(e)動的計画法(左：30 シーム、右：60 シーム)

図 a.8 Mandrill.bmp の処理結果

(a)元画像


(b)単純縮小画像(左：30 シーム、右：60 シーム)


(c)単純局所探索法(左：30 シーム、右：60 シーム)


(d)繰り返し局所探索法(左：30 シーム、右：60 シーム)


(e)動的計画法(左：30 シーム、右：60 シーム)

図 a.9 milkdrop.bmp の処理結果

(a)元画像


(b)単純縮小画像(左：30 シーム、右：60 シーム)


(c)単純局所探索法(左：30 シーム、右：60 シーム)


(d)繰り返し局所探索法(左：30 シーム、右：60 シーム)


(e)動的計画法(左：30 シーム、右：60 シーム)

図 a.10 Parrots.bmp の処理結果

(a)元画像


(b)単純縮小画像(左：30 シーム、右：60 シーム)


(c)単純局所探索法(左：30 シーム、右：60 シーム)


(d)繰り返し局所探索法(左：30 シーム、右：60 シーム)


(e)動的計画法(左：30 シーム、右：60 シーム)

図 a.11 Pepper.bmp の処理結果