

携帯機器による高解像度画像観賞のための画像処理技術の開発（継続）

代表研究者	井 上 光 平	九州大学	大学院芸術工学研究院	助教
共同研究者	浦 浜 喜 一	九州大学	大学院芸術工学研究院	教授
〃	原 健 二	九州大学	大学院芸術工学研究院	准教授

1 はじめに

近年、携帯電話をはじめとする各種携帯機器の小型化・軽量化が進み、どこでも簡単に必要な情報を見ることができるようになり、携帯機器の画面で高解像度の画像や映像を十分な画質で視聴したいという要求も高まってきている。そこで本研究では、携帯機器などの小さな画面で画質を損なわずに高解像度の静止画や動画を観賞するための画像処理技術を開発することを主な目的として、いくつかの画像処理アルゴリズムを提案し、その有効性を実験により検証してきた。以下に、本研究の概要を報告する。

2 階層的クラスタリングに基づく画像からの行/列除去

画像の内容を保ちながら画像サイズやアスペクト比を変更する方法として、Shamir ら[1]のシームカービング法がよく知られているが、Cho ら[2]は、リターゲットングの処理の途中で削除される画素の重要性(importance)を隣接する画素に拡散する方法を提案し、それによってシームカービング法と行/列除去(row/column removal)法を改良し、改良された行/列除去法によって、従来のシームカービング法よりも良い結果が得られる場合があることを示した。この行/列除去法は、シームカービング法よりも計算量が少ないという利点もある。本章では、階層的クラスタリングに基づく行/列除去法を提案する。すなわち、画像の行の集合と列の集合とをそれぞれ個別にクラスタリングし、各クラスタを1つの行あるいは1つの列にまとめることによって、行数と列数を削減する。従来の行/列除去法では、重要性の低い行や列を1つずつ除去していき、最終的に残ったものだけを提示するというものであり、除去された行や列は出力画像には反映されないが、クラスタリングを用いる提案手法では、各クラスタを1つの行や列にまとめるときに、例えば、各クラスタの要素を平均したものを用いることによって、すべての行と列を出力画像に反映させることができ、従来手法よりも滑らかな出力画像が得られると期待される。Cho らの行/列除去法[2]との比較実験により、提案手法の有効性を示す。また、本行/列除去法を応用した画像の拡大法を提案し、対象物のアスペクト比を保存しながら画像を拡大することができることを示す[3]。

2-1 行と列の階層的クラスタリング

行数 m 、列数 n の画像を $f = [f_{ij}], i = 1, \dots, m; j = 1, \dots, n$ とし、その第 i 行を $r_i = [f_{i1}, \dots, f_{in}]$ とする。クラスタを $C_i = \{r_i\}, i = 1, \dots, m$ と初期化し、クラスタ間の距離が最も短い 2 つのクラスタを融合するのを、クラスタ数が $K < m$ になるまで繰り返す。ここでは、2 つのクラスタ $C_k, C_{k'}$ 間の距離を

$$D(C_k, C_{k'}) = \min_{r \in C_k, r' \in C_{k'}} d(r, r') \quad \dots \quad (1)$$

とする。 $d(r, r')$ は各クラスタの要素 r, r' の間の距離である。ここでは、画像上で離れた位置にある 2 つの行が融合するのを防ぐために、

$$d(r, r') = \begin{cases} \|r - r'\| & \text{if } |i_r - i_{r'}| \leq 1 \\ \infty & \text{otherwise} \end{cases} \quad \dots \quad (2)$$

とする。 $i_r, i_{r'}$ はそれぞれ r, r' の行番号である。1 回の融合でクラスタ数は 1 だけ減少するので、クラスタ数

を K にするためには、 $m-K$ 回の融合を行えばよい。式(1)、(2)より、融合は画像上で隣接する行あるいはクラスタ間でしか生じないので、実際には、隣接行間の距離(m 行に対して $m-1$ 個の距離)を計算し、距離が小さい順に融合していけばよい。

以上の方法は、 m 個の 1 要素クラスタを徐々に融合していく凝集的なボトムアップ手法であるが、別の方法として、1 個の m 要素クラスタを徐々に分割していくトップダウン手法も考えられる。その場合、要素間の距離を式(2)とすると、分割前の全要素の集合はグラフ理論における道(path)で表すことができ、 $d(x,y)$ の値が大きい順に道の辺を取り除いていくことによって、クラスタを細分化していくという手順になる。結局、どちらの方法も、隣接行間の距離をソートする処理に帰着する。画像の列についても、同様にクラスタリングする。

2-2 行と列の除去

上記の方法で、画像 f の m 行が $K(< m)$ 個のクラスタにまとめられ、 n 列が $L(< n)$ 個のクラスタにまとめられたとする。1 つのクラスタを 1 行あるいは 1 列で表すことによって画像の行と列の数を減らし、行数 K 、列数 L の縮小画像を作成する。

各クラスタをその要素の平均で表すとすると、行の第 k クラスタ C_k は、1 つの行

$$\bar{r}_k = \frac{1}{|C_k|} \sum_{r \in C_k} r \quad \cdots \quad (3)$$

で表される。これを K 個のクラスタについて行くと、 f の行数を K に減らした画像

$$\tilde{f} = \begin{bmatrix} \bar{r}_1 \\ \vdots \\ \bar{r}_K \end{bmatrix} = [c_1, \dots, c_n] \quad \cdots \quad (4)$$

が得られる。 \tilde{f} の第 j 列を c_j とすると(式(4))、列の第 l クラスタ C_l は、1 つの列

$$\bar{c}_l = \frac{1}{|C_l|} \sum_{j \in J_l} c_j \quad \cdots \quad (5)$$

で表される。ここで J_l は C_l に属す列の番号の集合である。これを L 個のクラスタについて行くと、 \tilde{f} の列数を L に減らした K 行 L 列の画像

$$g = [\bar{c}_1, \dots, \bar{c}_L] \quad \cdots \quad (6)$$

が得られる。 g の k 行 l 列の値 g_{kl} は

$$g_{kl} = \bar{c}_l(k) = \frac{1}{|C_l|} \sum_{j \in J_l} c_j(k) = \frac{1}{|C_l|} \sum_{j \in J_l} \frac{1}{|C_k|} \sum_{r \in C_k} r(j) = \frac{1}{|C_k| \cdot |C_l|} \sum_{j \in J_l} \sum_{i \in I_k} r_i(j) = \frac{1}{|C_k| \cdot |C_l|} \sum_{i \in I_k} \sum_{j \in J_l} f_{ij} \quad \cdots \quad (7)$$

と書ける。ここで $\bar{c}_l(k), c_j(k)$ はそれぞれ \bar{c}_l, c_j の第 k 要素であり、 $r(j), r_i(j)$ はそれぞれ r, r_i の第 j 要素である。

また、 I_k は C_k に属す行の番号の集合である。すなわち、 g_{kl} は C_k に属す行と C_l に属す列の共通部分の画素の平均である。従って、行と列の除去の順序を入れ替えても、同様の結果が得られる。

2-3 行(列)を除去した後に列(行)を除去する方法について

上記の方法は、画像の行の集合と列の集合とをそれぞれ個別にクラスタリングし、各クラスタを 1 つの行あるいは 1 つの列にまとめることによって、行数と列数を削減するものであり、式(7)に示したように、結果

は行と列の除去の順序にはよらない．これとは別の方法として，行(列)をクラスタリングした後に，直ちに行(列)を除去し，その後，行(列)を除去した画像の列(行)をクラスタリングして列(行)を除去するという方法も考えられる．先に行を除去する場合，列のクラスタリングは画像 \tilde{f} (式(4))に対して行うことになる． f の第 j 列と第 $(j+1)$ 列の間の距離の 2 乗は，

$$\begin{aligned} \sum_{i=1}^m (f_{ij} - f_{i,j+1})^2 &= \sum_{k=1}^K \sum_{i \in I_k} (f_{ij} - f_{i,j+1})^2 = \sum_{k=1}^K |C_k| \frac{1}{|C_k|} \sum_{i \in I_k} (f_{ij} - f_{i,j+1})^2 \geq \sum_{k=1}^K |C_k| \left[\frac{1}{|C_k|} \sum_{i \in I_k} (f_{ij} - f_{i,j+1}) \right]^2 \quad \dots \quad (8) \\ &= \sum_{k=1}^K |C_k| [c_j(k) - c_{j+1}(k)]^2 \geq \sum_{k=1}^K [c_j(k) - c_{j+1}(k)]^2 = \|c_j - c_{j+1}\|^2 \end{aligned}$$

となり， \tilde{f} のそれとは一般に異なる．式(8)の 2 つめの不等号は $|C_k| \geq 1$ による．従って，行-列の順に除去する方法と列-行の順に除去する方法とは，一般に異なる結果が得られる．

2-4 画像の拡大

本節では，上記の行/列除去法を画像の拡大に応用する方法を提案する．ここでは，対象物のアスペクト比を保存しながら画像を拡大することを考える． m 行 n 列の画像 f を M 行 N 列に拡大する(すなわち， $M > m, N > n$) 場合，次の 2 つの方法が考えられる．1 つは， $am \geq M$ かつ $an \geq N$ となるように f の縦横を α 倍し，はみ出した行や列の分を行/列除去法で取り除くというものであり，もう 1 つは， $\beta m < M$ かつ $\beta n = N$ ，あるいは， $\beta m = M$ かつ $\beta n < N$ となるように f の縦横を β 倍し，足りない行あるいは列を追加するというものである．ここでは，後者の行/列追加法の 1 つを提案する．

f に Δ 行を追加するとする．すなわち， $\Delta = M - m$ である．隣接行間の距離 $d(r_i, r_{i+1}) = \|r_i - r_{i+1}\| (i=1, \dots, m-1)$ を計算し，その値が小さい順に K 個を選択する．選択された $d(r_i, r_{i+1})$ に対応する i の集合を S_K とする． $i \in S_K$ に対して値 $\phi(d(r_i, r_{i+1}))$ をとり，その他では 0 となる関数

$$\psi(i) = \begin{cases} \phi(d(r_i, r_{i+1})) & \text{if } i \in S_K \\ 0 & \text{otherwise} \end{cases} \quad \dots \quad (9)$$

を定め，それを累積したものを

$$\Psi(i) = \sum_{i'=1}^i \psi(i') \quad \dots \quad (10)$$

とする．更に，

$$\tilde{\Psi}(i) = \frac{\Delta}{\Psi(m)} \Psi(i) \quad \dots \quad (11)$$

とし， $[\tilde{\Psi}(i-1), \tilde{\Psi}(i)]$ の範囲内にある整数の個数を Z_i とする．ただし， $\tilde{\Psi}(0) = 0$ とする．式(9)内の $\phi(d(r_i, r_{i+1}))$ は $d(r_i, r_{i+1})$ の関数であり，いくつかの具体例を次節で示す．第 i 行 ($i \in S_K$) と第 $(i+1)$ 行の間に Z_i 行を追加する．

第 z_i 行 ($z_i \in \{1, \dots, Z_i\}$) は， r_i と r_{i+1} の線形補間

$$r_{z_i} = \frac{(Z_i + 1 - z_i)r_i + z_i r_{i+1}}{Z_i + 1} \quad \dots \quad (12)$$

とする．列の追加も同様にできる．

2-5 実験例

最初に，提案手法によって画像の行と列をクラスタリングした例を示す．図 1 (a) の画像の行と列をクラスタリングして得られたクラスタの境界線を図 2 に白線で示す．

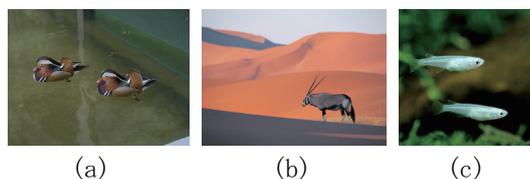


図 1: 入力画像: (a) 351×468 画素, (b) 768×1024 画素, (c) 500×500 画素

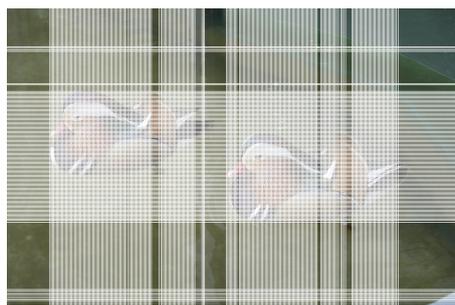


図 2: クラスタの境界線

白線で囲まれた各矩形が，出力画像では 1 つの画素で表される．その画素値は，式 (7) に示したように，矩形内の画素値の平均値となる．このように本手法は，対象物の領域を密にサンプリングし，その他の領域を疎にサンプリングするような画像縮小法であるといえる．

図 1 に示す 3 枚の画像から行と列を除去して画像を縮小した結果を図 3 に示す．図 3 (a), (b), (c) は，Cho らの手法 [2] による結果であり，図 3 (d), (e), (f) は提案手法による結果である．

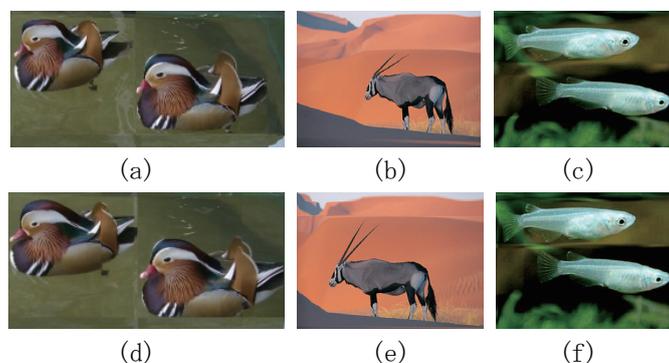


図 3: 行/列除去の結果: (a)-(c) Cho らの手法 [2], (d)-(f) 提案手法, (a, d) $K = \frac{m}{3}, L = \frac{n}{2}$,

$$(b, e) K = \frac{m}{2}, L = \frac{n}{2}, (c, f) K = \frac{m}{2}, L = \frac{2}{3}n$$

図 3 (a) では，左側の鳥の頭部がやや平らになっているが，図 3 (d) では，頭部の丸みが保たれている．図 3 (b) では，レイヨウの角が曲がっているが，図 3 (e) では，角の直線的な形状が保たれている．図 3 (c) では，下のメダカの口先が丸くなっているが，図 3 (f) では，元のやや尖った形状が保たれている．このように，提案手法では，対象の主要な形状を保ちながら行と列の除去ができる．また提案手法では，各クラスタの要素の平均を行と列に並べているので，行/列の除去によって生じる行間及び列間の不連続がある程度抑えられている．各画像を生成するのに要した計算時間を表 1 に示す．

表 1: 計算時間 (sec.)

	Cho らの手法[2]	提案手法
図 1(a)	0.187	0.156
図 1(b)	1.218	0.765
図 1(c)	0.312	0.265

どの画像においても、提案手法のほうが計算時間が短い。計算環境は Pentium 4(R) CPU 3.40GHz, 2.00GB RAM であり、プログラミング言語は Microsoft Visual C++ である。

次に、2-3 節に示した行(列)の除去後に列(行)を除去する方法による結果を図 4 に示す。

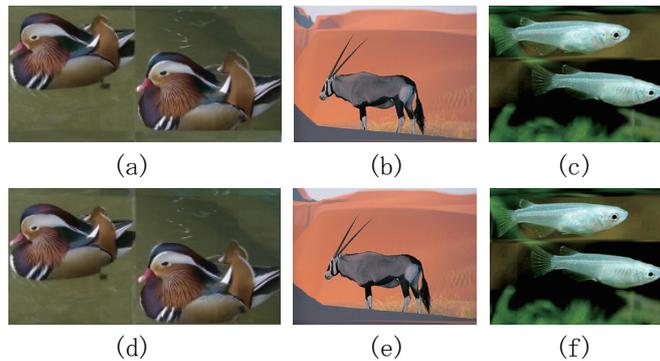


図 4: 行(列)-列(行)の順に除去した結果: (a)-(c) 行-列の順, (d)-(f) 列-行の順

図 1(a)-(c)の画像から行-列の順に除去した結果が図 4(a)-(c)であり、列-行の順に除去した結果が図 4(d)-(f)である。図 4(a)と(d)を比較すると、画像の上と下の背景領域の幅が異なる。図 4(b)と(e)では、左上の背景に違いが見られる。図 4(c)と(f)では、大きな違いは見られない。このように、2-3 節に示した 2 つの方法では、一般に異なる結果が得られる。

他の画像での結果を図 5 に示す。1 行目は入力画像であり、Liu ら[4]の画像データベースから選んだものである。2 行目は Cho らの手法[2]による結果であり、3 行目は提案手法による結果である。どの画像でも、Cho らの手法[2]による結果より提案手法による結果のほうが、対象物が大きく表示されており、行や列が除去された背景の不連続も少ない。例えば、1 列目の画像では、煉瓦の塔の側面の陰になっている部分が、図 5(f)では細くなっているが、図 5(k)では保存されている。また、1, 4, 5 列目の画像では、画像上部の背景部分に、図 5(f)では行を除去したことによって不連続が生じているが、図 5(k)ではそれが緩和されている。Cho らの手法[2]では、行や列を単純に削除するだけであるが、提案手法では、同一クラスに属する行あるいは列の平均を出力するので、このような違いが生じたと考えられる。

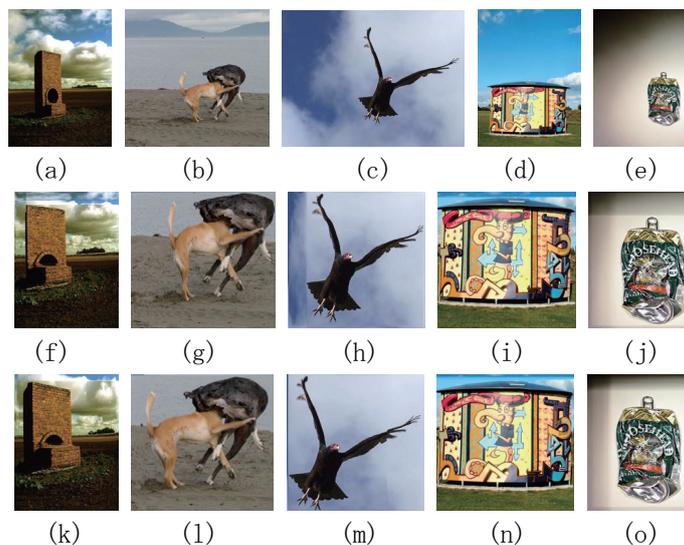


図 5: 他の画像での結果: (1 行目) 入力画像, (2 行目) Cho らの手法[2], (3 行目) 提案手法

次に、複雑な背景をもつ画像での結果を図6に示す。図6(a)は入力画像であり、図6(b), (c)はそれぞれChoらの手法[2], 提案手法による結果である。このように背景が複雑な画像では、対象物の領域を通過する行や列が優先的に除去されていくので、対象物の形状を保存した画像縮小は困難である。このような画像のリサイジングには、saliency map [5, 6]などの付加的な情報が必要になると考えられる。

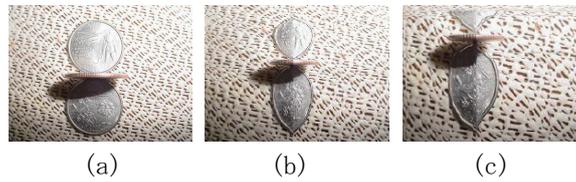


図6: 背景が複雑な画像での結果: (a) 入力画像, (b) Choらの手法[2], (c) 提案手法



(a) (b)
図7: 拡大用の画像

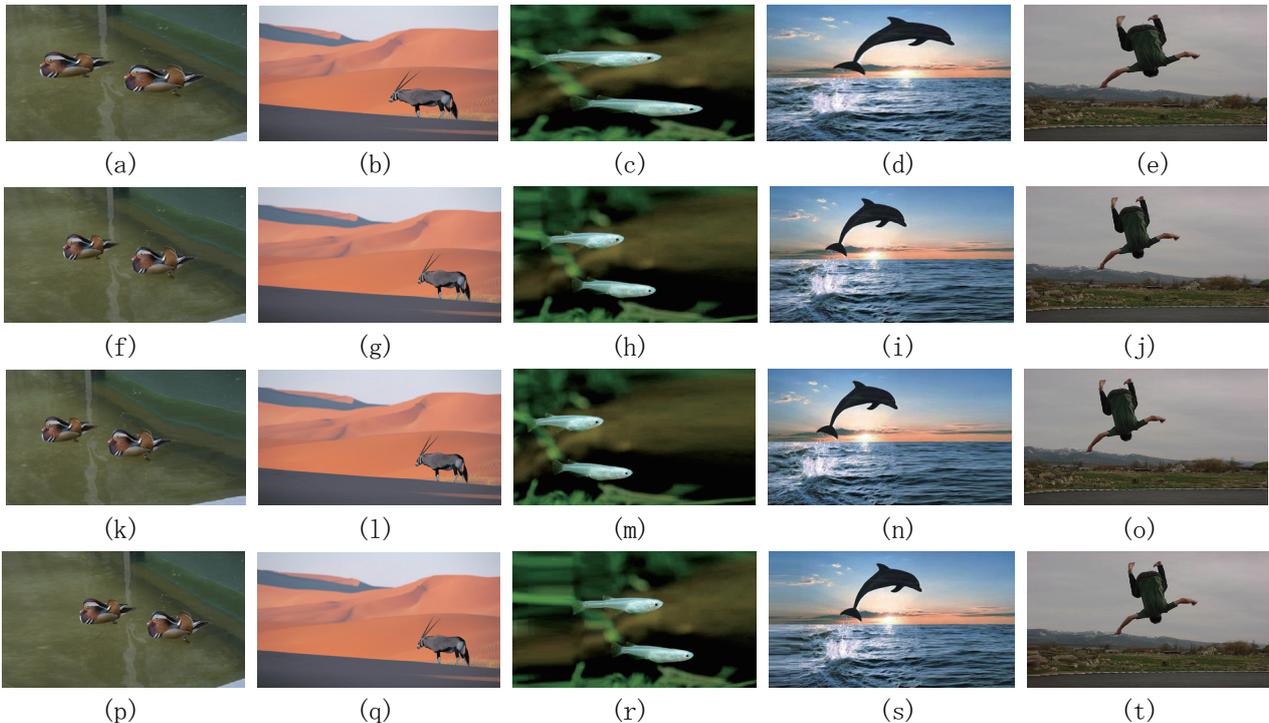


図8: 画像拡大の結果: (a)-(e) 線形補間, (f)-(j) $\phi(x)=1$, (k)-(o) $\phi(x)=x$, (p)-(t) $\phi(x)=1/(x+\epsilon)$

2-4節に示した画像の拡大の例を示す。図1と図7の画像を拡大した結果を図8に示す。この例では、結果の画像のアスペクト比が9:16になるように列を追加して横幅を広げた。図8(a)-(e) (1行目)は、横幅を線形に引き伸ばした結果であり、対象物も横方向に伸びている。図8(f)-(j) (2行目)は、 $\phi(x)=1$ とした結果である。このようにすると、選択された K 列の各列の横に同数の列が追加される。 K の値は、対象物の領域を通過する列が K 列の中に含まれないように、小さめの値に設定する必要がある。ここでは $K=n/4$ とした。図8(k)-(o) (3行目)は、 $\phi(x)=x$ とした結果である。このようにすると、選択された K 列の中で隣接列間の距

離が大きいところに多くの列が追加される。背景の中で複雑な領域を拡大したい場合には、このような方法が有効である。図 8 (p)-(t) (4 行目)は、 $\phi(x)=1/(x+\varepsilon)$ とした結果である。ここで ε は 0 による除算を避けるための定数であり、 $\varepsilon=1$ とした。このようにすると、選択された K 列の中で隣接列間の距離が小さいところに多くの列が追加される。つまり、変化の少ないところに列を追加することによって、追加の形跡をできるだけ目立たないようにしている。図 8(c)と(m)では、上のメダカの後部が伸びてやや不自然になっているが、その他の画像では、対象物のアスペクト比を保った画像の拡大ができていているといえる。また、例えば、図 8 の 4, 5 列目の画像では、 $\psi(x)$ を変えることによって、画像上での対象物の相対的な位置が変わっている。対象物の相対位置を制御可能な画像拡大法が望ましいが、これについては今後の課題としたい。

3 クリッピングされた矩形領域内外のカラーヒストグラムに基づく顕著性マップ

本章では、重み付きヒストグラムインタセクション(weighted histogram intersection: WHI)最小化による画像のクリッピング法に基づく顕著性マップを提案する[8]。すなわち、クリッピングされた矩形領域の内側と外側の各領域でカラーヒストグラムを求め、各色の矩形領域内に占める割合を顕著性として用いる。Achanta ら[6]の顕著性マップとの比較実験により、提案手法の有効性を確認する。

3-1 WHI 最小化によるクリッピング

一枚の限定色画像が与えられるとし、その色数を C とする。 $c \in \{1, \dots, C\}$ 番目の色の出現頻度(画素数)を \tilde{h}_c とし、カラーヒストグラムを $\tilde{h}=[\tilde{h}_1, \dots, \tilde{h}_C]$ とする。また、 \tilde{h} の要素の総和を 1 に規格化したものを $h=[h_1, \dots, h_C]$, $h_c = \tilde{h}_c / \sum_{c=1}^C \tilde{h}_c$ とする。二枚の画像 A, B の規格化したカラーヒストグラムをそれぞれ $h^A=[h_1^A, \dots, h_C^A]$, $h^B=[h_1^B, \dots, h_C^B]$ とすると、これらの重み付きヒストグラムインタセクション(weighted histogram intersection: WHI)は

$$s_w(h^A, h^B; w^A, w^B) = \sum_{c=1}^C w_c^A w_c^B \min\{h_c^A, h_c^B\} \quad \dots \quad (13)$$

で定義される[7]。ここで、 $w^A=[w_1^A, \dots, w_C^A]$, $w^B=[w_1^B, \dots, w_C^B]$ はそれぞれ画像 A, B の重みを並べたベクトルである。

縦 m 画素、横 n 画素の画像を縦 \tilde{m} 画素、横 \tilde{n} 画素の矩形でクリッピングするとする。また、矩形の左上の画素の座標を (\tilde{i}, \tilde{j}) とする。画像を矩形の内側と外側の二つの領域に分け、各領域の規格化ヒストグラムをそれぞれ $h^I(R), h^O(R)$ とする。ここで、 h の右肩の I と O はそれぞれ矩形の内側(inside)と外側(outside)を意味し、 $R=(\tilde{i}, \tilde{j}, \tilde{m}, \tilde{n})$ は位置 (\tilde{i}, \tilde{j}) と大きさ (\tilde{m}, \tilde{n}) で定まる矩形を表す。画像のクリッピングは、これら二つの規格化ヒストグラムの WHI を最小化するような矩形の位置と大きさを求める問題として、次のように定式化される[7]：

$$\min_R s_w(h^I(R), h^O(R); w^I(R), w^O(R)) \quad \dots \quad (14)$$

ここで、 $w^I(R)$ と $w^O(R)$ はそれぞれ矩形 R の内側と外側に対応する重みベクトルであり、それらの要素は

$$w_c^I(R) = h_c^I(R), \quad w_c^O(R) = h_c^O(R)^{-1} \quad \dots \quad (15)$$

で与えられる。

3-2 矩形領域内外のカラーヒストグラムに基づく顕著性マップ

(14) 式の解 R が求まると、それに応じて矩形領域内外のカラーヒストグラム

$\tilde{h}^I(R)=[\tilde{h}_1^I(R),\dots,\tilde{h}_C^I(R)], \tilde{h}^O(R)=[\tilde{h}_1^O(R),\dots,\tilde{h}_C^O(R)]$ が定まる．限定色画像の画素 (i, j) の色番号を c_{ij} とすると， c_{ij} 番目の色の画素の総数は $\tilde{h}_{c_{ij}}^I(R)+\tilde{h}_{c_{ij}}^O(R)$ で与えられる．その中で，矩形領域内の画素の色の占める割合が高ければ，その色は矩形領域内に含まれる対象物の色である可能性が高い．そこで， (i, j) の顕著性を

$$S_{ij} = \frac{\tilde{h}_{c_{ij}}^I(R)}{\tilde{h}_{c_{ij}}^I(R) + \tilde{h}_{c_{ij}}^O(R)} \quad \dots \quad (16)$$

とし，これをすべての画素について計算して顕著性マップ $S=[S_{ij}], i=1,\dots,m; j=1,\dots,n$ を作成する．

3-3 実験例

最初に，提案手法の基礎になる画像のクリッピングの例を図9に示す．



図9：画像のクリッピング：(a) 元画像，(b) クリッピングされた矩形領域(白枠)

図9(a)の画像を3-1節の方法でクリッピングすると，図9(b)の白枠で囲まれた矩形領域が選択される．次に，図9(a)の画像の顕著性マップを図10に示す．

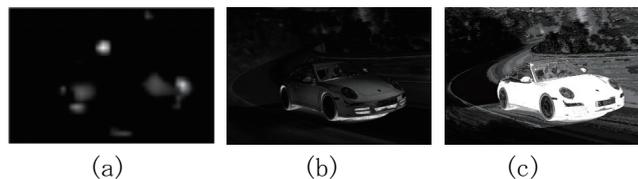


図10：顕著性マップ：(a) Itti [5]，(b) Achanta [6]，(c) 提案手法

図10(a)はIttiらの手法[5]による結果であり，Achantaら[6]も指摘しているように，元の画像よりも顕著性マップの解像度が低くなる．図10(a)の計算には，SaliencyToolbox [9]を用いた．図10(b)はAchantaらの手法[6]による結果であり，元の画像と同じ解像度の顕著性マップが得られる．図10(b)の計算には，Achantaが公開しているソフトウェア[6]を用いた．図10(c)は提案手法による結果であり，従来手法と比較すると，自動車の領域がより明確に抽出されていることがわかる． $C=64$ とし，メディアンカット法で減色した．他の画像での結果を図11に示す．

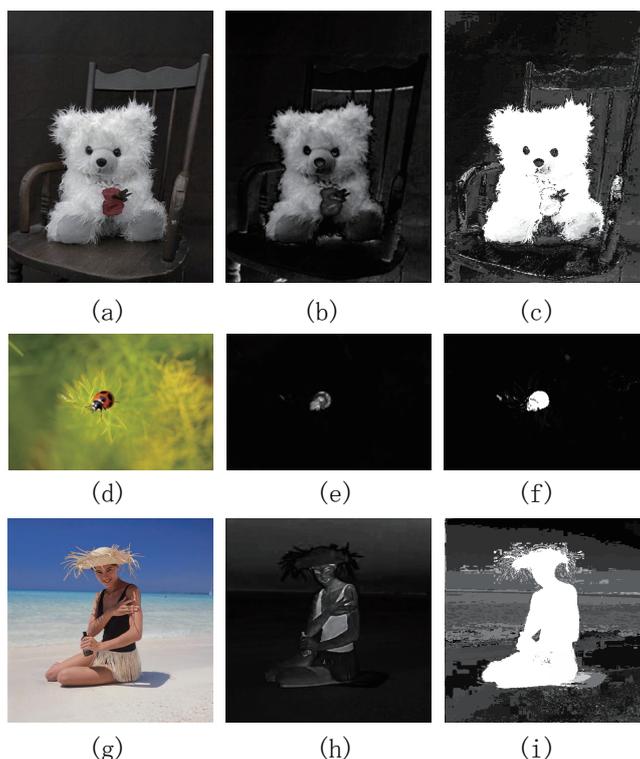


図 11: 他の画像例: 元画像((a), (d), (g)), Achanta らの顕著性マップ[6]((b), (e), (h)), 提案手法((c), (f), (i))

図 11(a), (d), (g)の画像から Achanta らの手法[6]によって求めた顕著性マップを図 11(b), (e), (h)に示す. また, 提案手法による顕著性マップを図 11(c), (f), (i)に示す. これらの例でも, 提案手法によって対象物の領域がより明確に抽出されている.

4 色顕著性マップを利用したシームカービング

画像の内容を保存しながら画像サイズを変える方法として, シームカービング[1]がよく知られている. Achanta ら[10]は, 大域的な顕著性マップを用いて画像内容の保存性を高めた. 本章では, 色顕著性マップ[8]を用いる方法を提案し, Achanta ら[10]の手法との比較実験例を示す[11].

4-1 色顕著性マップ

3-2 節の(16)式で定義される顕著性マップを色顕著性マップと呼ぶことにし, 次節では, 色顕著性マップをシームカービングに利用する.

4-2 シームカービング

縦 m 画素, 横 n 画素からなる画像の横幅を縮小することを考える. 画素 (i, j) の色の Lab 色空間における座標を f_{ij} ($i=1, \dots, m; j=1, \dots, n$) とする. 累積最小エネルギー(cumulative minimum energy [1]) $M = [M_{ij}]$ を次の手順で計算する. まず, 1 行目の値を

$$M_{1,j} = \begin{cases} S_{1,j} & \text{if } j \in \{1, n\} \\ S_{1,j} + g_{1,j} & \text{otherwise} \end{cases} \quad \dots \quad (17)$$

によって計算する. ここで $g_{ij} = \|f_{i,j-1} - f_{i,j+1}\|$ である. 次に, 2 行目以降の値を $M_{ij} = S_{ij} + \min_{l \in \{-1, 0, 1\}} \{\tilde{M}_{i-1, j+l}\}$ によって順次計算する. ここで

$$\tilde{M}_{i-1,j+l} = \begin{cases} g_{ij} + M_{i-1,j} & \text{if } l=0 \\ g_{ij} + \|f_{i-1,j} - f_{i,j+l}\| + M_{i-1,j+l} & \text{otherwise} \end{cases} \quad \dots \quad (18)$$

である。最後に、 $j^* = \arg \min_j \{M_{mj}\}$ を求め、1 行目から m 行目の j^* に至るパスを逆に辿ることによって最適なシームを求め、そのシームを削除する。以上の手順によるシームの削除を繰り返して画像の横幅を縮小する。

4-3 実験例

従来のシームカービングで図 12(a) の画像の横幅を半分に縮小すると図 12(b) になる。

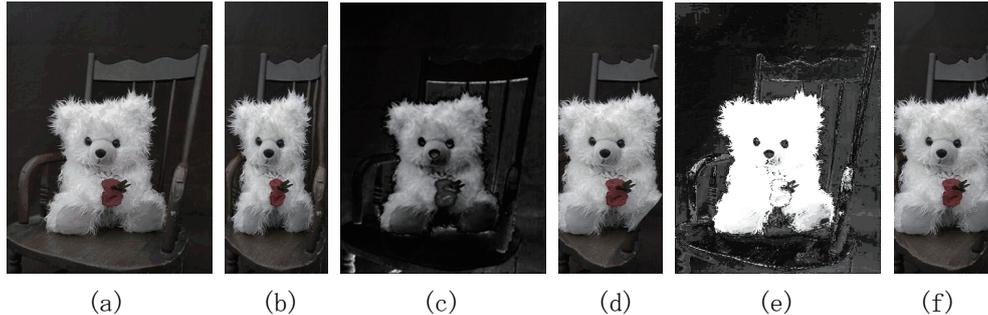


図 12: 横幅縮小の例: (a) 元画像, (b) シームカービング[1], (c) Achanta ら[6]の顕著性マップ, (d) 顕著性マップ(c)を用いた横幅縮小, (e) 色顕著性マップ, (f) 色顕著性マップ(e)を用いた横幅縮小

Achanta の顕著性マップ(図 12(c))を用いると図 12(d)になり、色顕著性マップ(図 12(e))では図 12(f)になる。図 12(f)ではぬいぐるみの横幅が保たれており、左足も元に近い形で残っている。

5 まとめ

本研究では、携帯機器などの小さな画面で画質を損なわずに高解像度の静止画や動画を観賞するための画像処理技術を開発することを主な目的として、階層的クラスタリングに基づく画像からの行/列除去による画像縮小法及びその画像拡大への応用法、クリッピングされた矩形領域内外のカラーヒストグラムに基づく顕著性マップの生成法、色顕著性マップを利用したシームカービング法を提案し、実験でその有効性を確認した。画像中の重要な領域を保存するだけでなく、更に強調するような画像のリサイジング法の検討が今後の課題である。

【参考文献】

- [1] A. Shamir and S. Avidan, "Seam carving for media retargeting," Commun. ACM, vol. 52, no. 1, pp. 77-85, 2009.
- [2] S. Cho, H. Choi, Y. Matsushita, and S. Lee, "Image retargeting using importance diffusion," Proc. ICIP, pp. 977-980, Nov. 2009.
- [3] 井上光平, 原健二, 浦浜喜一, "階層的クラスタリングに基づく画像からの行/列除去," Visual Computing/グラフィクスとCAD 合同シンポジウム, 15, Jun. 2010.
- [4] T. Liu, J. Sun, N.-N. Zheng, X. Tang, and H.-Y. Shum, "Learning to detect a salient object," Proc. CVPR, 2007.
http://research.microsoft.com/en-us/um/people/jiansun/SalientObject/salient_object.htm
- [5] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," IEEE Trans. PAMI, vol. 20, no. 11, pp. 1254-1259, 1998.
- [6] R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk, "Frequency-tuned salient region detection," Proc. CVPR, 2009.

http://ivrg.epfl.ch/supplementary_material/RK_CVPR09/

- [7] 井上光平, 原健二, 浦浜喜一, ``重み付きヒストグラムインタセクション最小化による画像のクリッピング," 信学技報, vol. 109, no. 292, IE2009-113, pp. 85-88, 2009.
- [8] 井上光平, 原健二, 浦浜喜一, ``クリッピングされた矩形領域内外のカラーヒストグラムに基づく顕著性マップ", 2010年映情学年次大, 7-1, 2010.
- [9] D. Walther and C. Koch, ``Modeling attention to salient proto-objects," Neural Networks, vol. 19, pp. 1395-1407, 2006.
- [10] R. Achanta and S. Susstrunk, ``Saliency detection for content-aware image resizing", Proc. ICIP, 2009.
- [11] 井上光平, 原健二, 浦浜喜一, ``色顕著性マップを利用したシームカービング," 2010映情学冬大, 4-11, 2010.

〈発 表 資 料〉

題 名	掲載誌・学会名等	発表年月
階層的クラスタリングに基づく画像からの行/列除去	Visual Computing/グラフィクスとCAD合同シンポジウム	2010年6月
反復クロスバイラテラルフィルタによる画像の2値化	映像情報メディア学会誌	2010年7月
クリッピングされた矩形領域内外のカラーヒストグラムに基づく顕著性マップ	映像情報メディア学会年次大会	2010年9月
重み付きヒストグラムインタセクション最小化による画像とビデオのクリッピング	映像情報メディア学会誌	2010年11月
Image and Video Clipping by Weighted Histogram Intersection Minimization	TENCON 2010	2010年11月
Image Resizing by Row/Column Removal and Addition Based on Hierarchical Clustering	TENCON 2010	2010年11月
色顕著性マップを利用したシームカービング	映像情報メディア学会冬季大会	2010年12月
行と列のクラスタリングによる画像縮小	電子情報通信学会論文誌	2011年1月