

超解像処理のための黒色粉末の散布による画素形状のコード化

笹尾 朋貴[†] 日浦 慎作^{††} 佐藤 宏介[†]

[†] 大阪大学 大学院基礎工学研究科 〒 560-8531 豊中市待兼山町 1-3

^{††} 広島市立大学 大学院情報科学研究科 〒 731-3194 広島市安佐南区大塚東 3-4-1

E-mail: [†]{sasao,sato}@sens.sys.es.osaka-u.ac.jp, ^{††}hiura@hiroshima-cu.ac.jp

あらまし 本研究では、撮像素子の各画素形状をコード化することで超解像を高性能化する手法を提案する。従来の研究では撮像素子の各画素が矩形であることを前提としていた。そのため、各画素のサンプリングは点サンプリングではなく正方形の畳み込みによって原画像の高周波成分が失われることになる。そこで本研究では、撮像素子の上に細かい黒色粉末を振りかけることで各画素の受光分布のランダムなコード化を行い、正方形の畳み込みによる原画像の高周波成分の損失を低減する手法を提案する。黒色粉末を振りかけた場合、どのようなコードになっているか未知となるため、その推定方法を示し、実験によりコードの推定を確かめた。更に、推定したコードを用いて作成した超解像画像を評価する。

キーワード 超解像, 黒色粉末, コード推定, 撮像素子, Richardson-Lucy deconvolution

1. はじめに

近年, CCD や CMOS センサなどの撮像素子の微細化が進んでおり, コンパクトデジタルカメラや携帯電話搭載のカメラなどの小型のカメラでも解像度が飛躍的に向上している。しかし CCD などの撮像素子は画素数が大きくなるほど読み出しに時間がかかる, 感度が低下するためぶれやすくなるなどの問題があり, 動物体の撮影が困難になる。この問題の一つの解決法として画素数の小さい撮像素子を用いて撮影し, 超解像によりセンサの解像度以上の高解像度画像を得ることが望まれる。

ここで, 撮像素子のそれぞれの画素が点サンプリングの場合, サンプリング定理よりそのサンプル点の密度によって取り込むことができる空間周波数が決定される。そこで像を僅かにずらして何度も撮影することでサブピクセル単位で異なる画像をキャプチャーしてサンプル点の密度を上げるのがマルチフレーム超解像の考え方である。しかし実際の撮像素子の受光セルが点ではなく面積を持っているため, 各画素のサンプリングは有限の面積内の平均となってしまう。そのため図 1(a) に示すように正方形の畳み込みによって原画像の高周波成分が失われることになる。つまり原画像の高周波成分が失われた画像に対して超解像を行うことになるため, 従来研究では失われた情報を何らかの先験的知識を用いて推定することで解像度を向上させることになる。ただしこの方法では, 入力画像が使用した知識と異なっていると生成された高解像度画像は誤ったものになってしまう。

その他に, 画素の大きさを小さくして高周波成分を残存させる方法が考えられる。しかし, 図 2 に示すように解像度が高くなると, 各画素の大きさが小さくなるために露光量が減少し, ノイズが増加してしまう問題がある。

そこで我々は, 従来手法が積分区間全体について一様に積分していたことに対し, 図 1(b) のように一様に積分しないことで, 情報ができるだけ画像上に残存するようにする方法を提案した [1]。

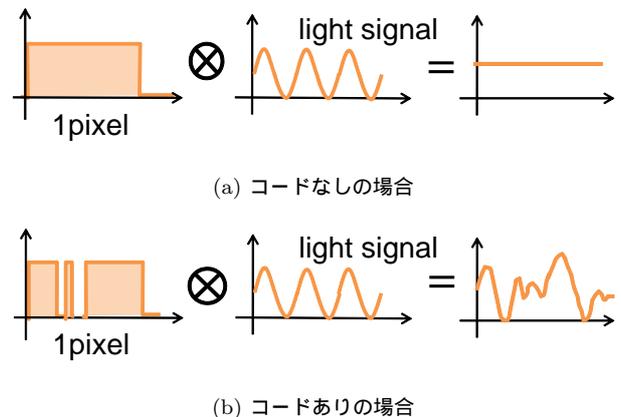


図 1 撮像素子のセルにおける畳み込み演算

これをコード化 (符号化) と呼び, 本研究では撮像素子のそれぞれの画素形状をコード化し, 高周波成分の損失を防いで超解像を行うことに対応する。図 2 に示すように, 各画素をコード化した場合, 高周波成分の損失を防ぐことができ, さらに超解像により従来システムより SN 比が高くなると考えられる。コード化の方法として撮像素子の上に細かい黒色粉末を振りかけて受光分布をランダムにコード化する手法を提案する。コード化を行うと一部の光を遮断することになり, それにより露光量が減少することが問題点としてしばしば指摘される。しかし, 全区間を一様に積分する方法では必ず高周波成分の損失が発生するため, コード化により光量を少し減らしながらも高周波成分の損失を防ぐことにより, 超解像

による解像度が向上することを提案する。

本稿では、我々の手法 [1] を実機で実現するため、実装に伴うコードの推定方法を示す。また、推定したコードを用いて作成した超解像画像の評価を行う。

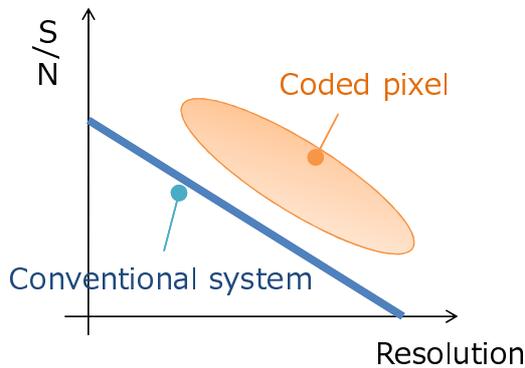


図 2 超解像における解像度と SN 比の関係

2. 関連研究

超解像に関する研究は、対象となる 1 枚の画像のみを用いて空間解像度を向上させる手法 [2], [3] と、動画像などの複数の画像を用いて空間解像度を向上させる手法 [4], [5] に大別できる。前者を Single Frame 超解像、後者を Multi Frame 超解像と呼ぶことにする。また、本研究に関連するものとして、画素形状に関連する超解像手法 [6] ~ [8] が挙げられる。本章では、これら関連分野における研究を紹介し、本研究の位置づけを明らかにする。

2.1 Single Frame 超解像

1 枚の画像のみを用いる手法では、最近傍補間やバイリニア補間などのように、隣接する画素値や周辺の画素値の統計量を利用して補間する画素値を決定し、空間解像度を向上させる。これらの手法は、平滑化を行うことが基本となるため、生成される画像はエッジ部分や画像の細部でぼけてしまい鮮明さに欠けるという問題がある。川合ら [2] は、画像を切り出す局所ウィンドウの大きさを変えながら固有ベクトルを生成した場合に、この固有ベクトルがほとんど変化しない性質を利用して空間解像度を向上させる手法を提案している。また Hong ら [3] は、画像の局所的な統計量を用いた手法を提案している。このようなアプローチは計算コストが低いという特長があるが、一般に複数の画像から空間解像度を向上させる手法に比べて空間解像度の向上効果は低い。

2.2 Multi Frame 超解像

対象物体に対するサンプリング点の位置が少しずつずれた複数の観測画像を対象画像にサブピクセル精度で対応付けることで高解像度画像の画素値を推定し、空間解像度を向上させる。Nayar ら [4] は、画像間の幾何学的

な位置ずれを既知とするために、撮像素子の位置を制御可能な特殊なカメラを用いた提案している。また Raskar ら [5] は、開口部にマスクを置き、さらに意図的にボケを与えることで観測画像のわずかな画素の変位を得る手法を提案しているが、このような特殊な機器を用いない場合は画像間の対応関係を求める必要がある。

2.3 画素形状に関する研究

画素形状に関連する超解像研究の一つに Penrose Pixels と呼ばれる Ben-Ezra らの手法 [6] がある。Penrose Pixels では、各画素の大きさやサイズが異なるため、矩形の画素形状と比べて非周期的にサンプリングすることができ、超解像の精度を向上させている。これは、図 1(b) で示したように、原画像の情報が画像上に残存することによって貢献しているためである。しかし、Penrose Pixels では画素の構造がそれぞれ異なるため、配線を繋ぐことが難しく、製造が困難という問題点がある。また、Penrose Pixels が 10 種類ほどの形状しかないことに対し、本研究で提案するランダムコードではすべての画素形状が異なり、様々なカメラの動きに対しても広く使用することができる。

その他に、組合せ画素混合を利用した田中らの手法 [7] がある。組合せ画素混合では、一度の撮影で 9 画素混合画像と間引き読み出し画像を生成するが、前者はエイリアシングの影響が弱い画像、後者はエイリアシングの影響が強い画像となる。超解像において、位置合わせ処理にエイリアシングの影響は好ましくなく、再構成処理ではエイリアシングの影響を利用して高解像度画像を再構成しているため、画素混合画像を位置合わせ処理に利用し、間引き読み出し画像を再構成処理に利用して効果的な超解像処理を行っている。しかし、間引き読み出し画像は 9 画素の代表画素値として一つの画素のみを読み出すため、9 画素全体から見ると 1/9 の受光量となってしまう。撮影された画像のノイズが増加しやすいという問題がある。そのため、ノイズの多い間引き読み出し画像を使用した再構成処理では、超解像画像の精度が落ちてしまう。

また、画素をランダムに配置した画像から原画像を復元する手法として、中静ら [8] のスパース性を用いた手法が挙げられる。この手法では、小領域を画像から多数切り出し、その構造を基底として学習する。その上で、ランダムに配置された画素の間の欠損部分を基底の線形結合により補う方法で原画像を復元している。この際にスパース性の尺度を用いることで、一般的な直交基底を用いた場合に比べ、より高品質な画像を復元することに成功している。

2.4 本研究の位置づけ

本章で紹介した全ての研究では、撮像素子における各画素の受光セルを矩形（正方形）のまま扱っているため、原画像の高周波成分が失われた画像に対して超解像

を行っていることになる．そこで本研究では，撮像素子における画素形状をコード化することで各画素のサンプリング関数（受光分布）のコード化を行い，正方形の畳み込みによる原画像の高周波成分の損失を低減させ，超解像を行うことを目的とする．図 3 に提案システムの流れを示す．

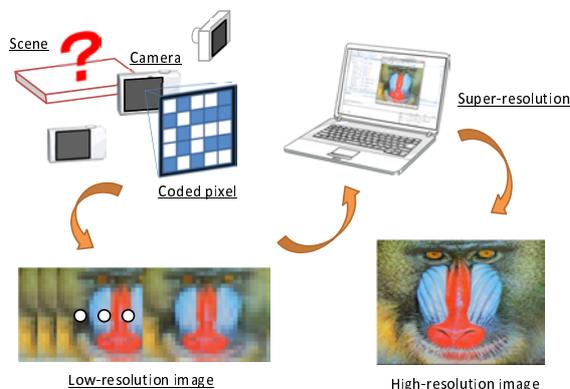


図 3 提案システムの流れ

3. 画素形状のコード化による超解像処理

本稿では，撮像素子における画素形状をコード化することで複数の低解像度画像から高解像度画像を推定する手法を提案する．撮像素子の上に細かい黒色粉末を振りかけてランダムにコード化することで各画素のサンプリング関数のコード化を行い，正方形の畳み込みによる原画像の高周波成分の損失を低減する．

3.1 コードと原画像の情報損失との関係

超解像においてカメラから入力される画像は必ずしも望ましいサンプル点配置を持つとは限らない．たとえば，カメラの動きを図 4 のように横方向に制限した場合のことを考える．

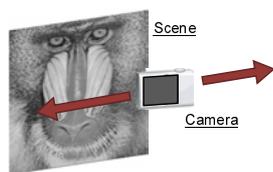


図 4 カメラの動きを横移動のみにした場合

ここで例として，原画像を 20×20 の解像度とし，カメラの 1 画素は原画像の 5×5 に対応するものとする．図 5 の赤枠を拡大したものを (a) とし，撮像素子をそれぞれコード化したものによる受光分布を表したものを (b)，(c) とする．

まず (a) はコード化していない一般的なカメラと同様の撮像素子で，このまま横にずらし画像をキャプチャーしていくと原画像のすべての画素を通過することができるが，先に述べたように画素の幅に対応した高周波成

分が失われることになる．それに対し (b) のピンホールコードの撮像素子では，積分範囲が小さいために高周波成分が残りやすいが，このまま横にずらし画像をキャプチャーしていくと，3 行目以外の原画像の情報を得ることができない．つまり (a)，(b) の方法では縦方向の解像度を向上させることができない．そこで (c) のようにランダムにコード化した撮像素子を用いてキャプチャーすることで，高周波成分の損失を減らしながらも，原画像のすべての画素を通過することができ，かつ (a)，(b) に比べて情報損失の少ない観測画像が得られると考えられる．よって，カメラの動きが横や斜めなどに制限された場合，ランダムにコード化した撮像素子を用いる効果は大きい．

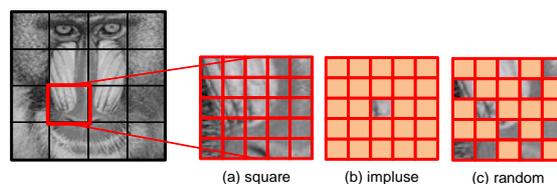


図 5 コードによって得られる情報の違い

3.2 Richardson-Lucy deconvolution の拡張

本研究では既存の超解像アルゴリズムの一つである Richardson-Lucy deconvolution (以下 RL 法) を用いて超解像を行う．

RL 法は，既知の PSF によって畳み込まれた原画像を復元するための反復処理である． p_{ij} を PSF， u_j を画素位置 j における原画像の画素値， d_i を観測された画像の画素位置 i における画素値としたとき，観測画像 d_i は

$$d_i = \sum_j p_{ij} u_j \quad (1)$$

のように表すことができる．基本的な考えは， d_i が観測され p_{ij} が既知であるときに尤も確からしい u_j を推定するものである．ここから，

$$u_j^{(t+1)} = u_j^{(t)} \sum_i \frac{d_i}{c_i} p_{ij} \quad (2)$$

ただし，

$$c_i = \sum_j p_{ij} u_j^{(t)} \quad (3)$$

として，反復的に u_j を求めることができる．初期高解像度画像 $u_j^{(0)}$ は任意の値でよく，式 (3) による推定観測画像 c_i と観測画像 d_i との比から式 (2) のように高解像度画像を求めることになる．この反復が収束すれば高解像度画像 u_j の最尤解に収束することが経験的に示されている．

PSF である p_{ij} が一つ以上の未知パラメータに依存す

る問題では，RL 法を利用することができないため，より一般的なアルゴリズムである EM アルゴリズムによりこの種の問題の解法を示しているが，本研究では p_{ij} がコードにあたり，その値を推定することにより既知とするため，RL 法を利用することができる．

ここで，RL 法による高解像度画像の推定において，理想的な推定結果は原画像と同一のものとなる．このとき，推定した高解像度画像から式 (3) で生成した観測画像 c_i は実際に観測された画像 d_i とまったく同じものになっているはずである．つまり， $d_i = c_i$ となるため，次の式 (4) が言える．

$$\frac{d_i}{c_i} = 1 \quad (4)$$

このとき

$$u_j^{(t+1)} = u_j^{(t)} \sum_i p_{ij} \quad (5)$$

となり，復元画像 u_j は更新する必要がないため式 (6) のようになる必要がある．

$$\sum_i p_{ij} = 1 \quad (6)$$

しかし，ランダムコードのような各画素における PSF が異なる場合では，式 (6) は成り立たない．そこで，RL 法を拡張して (7) のように再定義し，シフト変の PSF の場合においても式 (6) が成立するようにする．

$$u_j^{(t+1)} = u_j^{(t)} \sum_i \frac{d_i}{c_i} \frac{p_{ij}}{\sum_i p_{ij}} = \frac{1}{\sum_i p_{ij}} u_j^{(t)} \sum_i \frac{d_i}{c_i} p_{ij} \quad (7)$$

さらに，ゼロ除算を防ぐため式 (8) のように分子・分母に定数 Const を足し合わせる．

$$u_j^{(t+1)} = \frac{1}{\sum_i p_{ij}} u_j^{(t)} \sum_i \frac{d_i + \text{Const}}{c_i + \text{Const}} p_{ij} \quad (8)$$

式 (8) はゼロ除算を防ぐだけでなく，推定画像 c_i と観測画像 d_i の黒色成分が大きくそれぞれの画素値が 0 に近くなるときにも， $\frac{d_i}{c_i}$ が小さすぎる，または大きすぎる値になることを防ぐことができる．本研究では，定数 $\text{Const} = 10$ とした．このように RL 法を拡張したとき，ランダムコードのような各画素における PSF が異なる場合に対しても RL 法の使用が可能となる．

4. 実機への実装

画素形状をランダムに符号化する手法 [1] を実機で実現するためには，撮像素子上に細かな黒色粉末を振り掛け，画素形状をコード化が必要がある．しかし，このようにコード化を行う場合，以下のような課題がある．

- 黒色粉末の大きさは，受光セルの大きさよりも小さくなければならないこと

- 黒色粉末はランダムに付着するため，それぞれの画素の形状は相異なり，またそれぞれがどのような形状になっているか未知であること

以下，上記の課題を解決しつつ，コードの実装と推定を行う方法について詳しく述べる．

4.1 コードの実装

実機で提案手法 [1] を確認するために，撮像素子の上に細かい黒色粉末を振りかけて各画素形状をランダムにコード化することを考える．カメラは画素サイズ $6.7 \mu\text{m} \times 6.7 \mu\text{m}$ のカメラ (Lumenera 社，産業用 USB2.0 カメラ/Lu120&Lu125) を使用した．また，黒色粉末は 1 画素よりも小さい必要があるため，大きさ約 $6 \mu\text{m}$ のレーザプリンタのトナーを使用した．トナーは粒子と粒子がくっつくと面積が大きくなるため，均一に分散させる必要がある．そこで，トナーに風を吹きつけることで粉塵にし，コード化を行った．図 6 は，撮像素子にのせたトナーを顕微鏡で観察したものである．この図より，1 画素よりも小さな大きさでトナーがのっていることが確認できる．

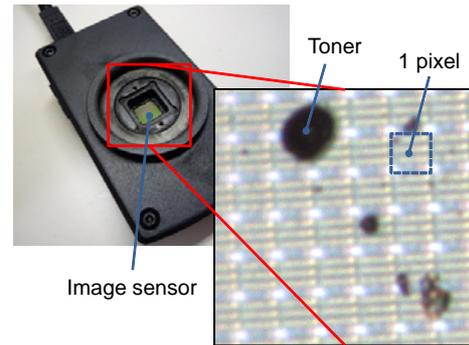


図 6 顕微鏡で観察した撮像素子上のトナー

4.2 コードの推定

コードの推定を行う方法として，図 6 のように顕微鏡によりトナーの位置を測る方法があるが，位置合わせが難しく，画素数も非常に多いことから現実的ではない．そこで，カメラで撮影した画像から各画素の形状を求めため，以下のような方法を考えることができる．

- (1) 非常に小さい点光源をカメラの前に置き，それを撮影する．すると，その点光源の位置に対応した画素の画素値が上昇する．
- (2) その点光源を微小に動かしながら，撮影を繰り返す．
- (3) このとき各画素の受光範囲内に点光源が入ると画素値が上昇し，一方，受光範囲から逸脱したり，光が黒色粉末によって阻まれたりすると画素値が下降する．

つまり，点光源に対応した位置にトナーの粉末があると画素値が小さくなるため，このときの画素値の変化と，

点光源の位置の関係から、それぞれの画素の形状を知ることができる(図7)。

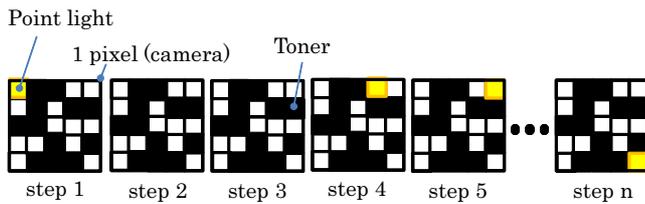


図7 コード推定の概要図

点光源を正確に移動させることが難しいことと、また後に述べるように、複数の点光源を同時に点灯させる必要があることから、本研究では光源としてディスプレイを使用する。カメラとディスプレイは図8のように正対するように配置し、ディスプレイ上の表示パターンとカメラによる撮影画像の対応関係から、それぞれの画素の受光分布形状を求める。



図8 カメラとディスプレイの対応関係

しかし、先に書いたような点光源の移動による方法、つまりディスプレイの全画素を一つずつ表示させていく方法は計測に非常に長時間を要する。また、ある点光源を点灯させたときに、それによって画素値が影響を受ける画素は、点光源の像の位置にある画素に限られるはずである。しかし実際には、ディスプレイの画素は大きさを持ち、さらに図9のようにディスプレイとカメラの画素はその境界が正確に重なっているとは限らないため、そのずれにより、一つのディスプレイ画素が複数のカメラ画素に影響を与える可能性がある。

そこで図10に示すように、あるカメラ画素は、ディスプレイ上の一つの画素と、それを中心としたある一定の大きさの領域内にある画素からのみ影響を受けると仮定する。つまり、その領域の外にあるディスプレイ画素の輝度には応答しないと仮定する。この領域を影響領域と呼ぶこととする。このようにすると、ディスプレイの点灯パターンに対する画素値の応答は、図11に示すような疎行列で表されることになり、影響領域の外に対応する要素は0の値を持つ。画素形状の計測およびデータ表現としては、このうち非零となる領域の位置と、その内部の値のみを計測し格納する。

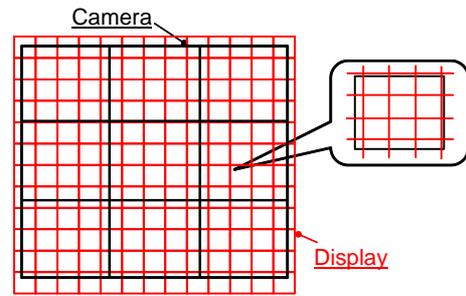


図9 シーン上におけるディスプレイとカメラの画素のずれ

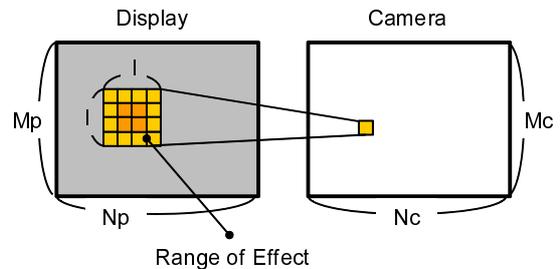


図10 対応画素を中心に設定する影響領域

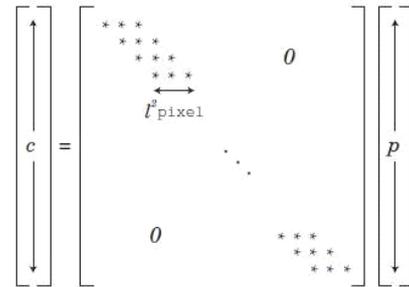


図11 影響領域設定による疎行列

影響領域の設定により、その影響領域に対応したカメラ画素は、影響領域外の点灯パターンに影響を受けない。これにより、すべてのカメラ画素に対応した影響領域内で常に一つだけの画素が点灯するようにすれば、複数の画素において同時計測が可能となる。あるカメラ画素の画素値が上昇した場合、それに影響したディスプレイ画素の位置を知るためには、そのカメラ画素に対応した影響領域内の点灯画素の位置を調べればよい。よって、ディスプレイに表示する点灯パターンとしては、図12に示すように影響領域の一辺の大きさを間隔としたドットパターンを用いる。これにより、例えば影響領域の大きさ、すなわちドットパターンの間隔が縦・横各10画素であるとすると、たかだか100回の計測によりすべての画素形状を求めることができる。

ドットパターンの点灯に先立ち、各画素の影響領域をディスプレイ上に設定する必要がある。これにはグレイコードパターンを用いた。ディスプレイ上に縦と横のグレイコードパターンを表示することで、全てのカメラ画素に対応したディスプレイ画素の位置が整数値で求められる。ただしこの位置はカメラ画素の中央に対応してい

るとは限らないため、安全のため、影響領域の大きさは、カメラの受光範囲が対応するディスプレイ上の画素数に対して十分大きくしておく必要がある。

これらにより得た計測結果は、各カメラ画素に対し、それに対応したディスプレイ画素の位置と、その周囲の影響領域内の画素からの応答としてメモリに格納する。これにより、図 11 に示した行列全体を格納する場合に対し、大幅に記憶容量を削減することが出来る。

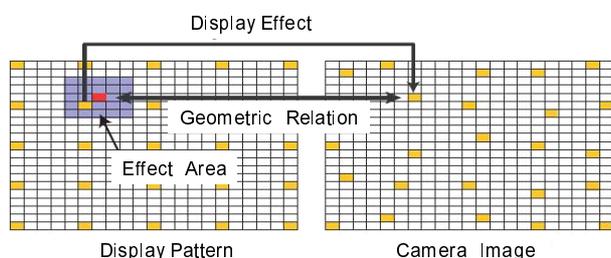


図 12 ドットパターンの投影

5. 実験

5.1 コードの推定に関する実験

まず、図 13 はコード化したカメラで白紙を撮影したものであり、撮像素子全体を均等にトナーがのっていることがわかる。そこで、4.2 節の方法により、このときのコードの推定を行うため、図 14 のようにディスプレイが映る位置にカメラを配置して実験を行った。実験環境は以下のようにして行った。

- F 値：1.4
- 露光時間：350ms
- カメラ解像度：1280*1024
- ディスプレイ解像度：1600*1200
- カメラの 1 画素はディスプレイの 15*15 画素領域から影響を受ける



図 13 白紙の撮影



図 14 撮影状態

ここで、本稿ではカメラ座標の (410,830) ~ (355,655) の範囲にディスプレイを映し、その範囲のコードを求めた場合の結果を示す。図 15 は推定したコード結果であり、図中の白い四角形は、カメラの 1 画素を表している。この図より、撮像素子の画素サイズ (6.7 μ m*6.7 μ m) に対し、トナーサイズ (約 6 μ m) がカメラの 1 画素を

ランダムにコード化できるほど小さくないため、細かいコード化ができていないことがわかる。

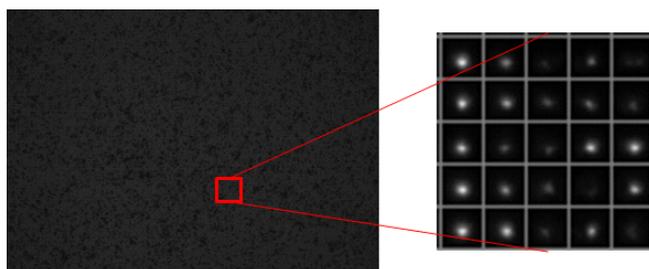


図 15 黒色粉末をのせた場合に推定したコード

そこで、カメラの 3*3 画素を仮想的に 1 画素とし、相対的にトナーサイズが小さくなるようにして実験を行った。仮想的な 1 画素の画素値は、3*3 画素の画素値の合計値としている。実験では、その仮想的な 1 画素はディスプレイの 21*21 画素領域から影響を受けると仮定し、それ以外は上記と同様に行った。このとき画素形状の推定に用いるドットパターンは 421 種類であり、これにより推定した画素形状を図 16 に示す。さらに、同様の条件で、黒色粉末を撮像素子に振り掛けていない状態でコード推定を行った結果を図 17 に示す。それぞれの図中の白線で区切られた領域は各画素の影響領域を表し、その内部で画素値が高い部分がそれぞれの画素の受光分布形状を表す。

まず、図 16,17 より、3x3 画素をまとめて一つの仮想画素としているため、元のセンサの開口径率により物理的な画素の間で感度が低い部分が存在することが分かる。また、粉末なしの場合の推定結果である図 17 より、各画素の中心ほど受光量が大きくなっていることがわかり、2次元のガウス分布状になっていることが確認できる。この結果に対し、図 16 では、カメラの仮想的な 1 画素の受光分布がランダムになっていることが確認できた。この結果より、ランダムなコードの実装を行うことができたと考える。

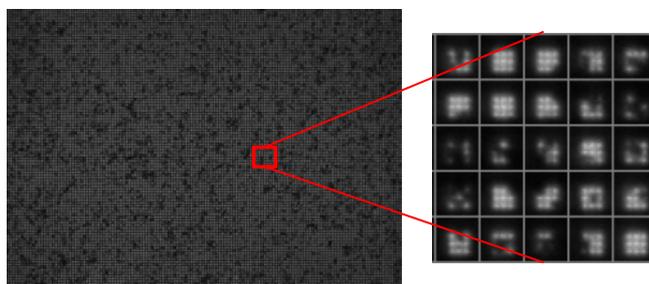


図 16 黒色粉末をのせた場合に推定した仮想的な画素に対するコード

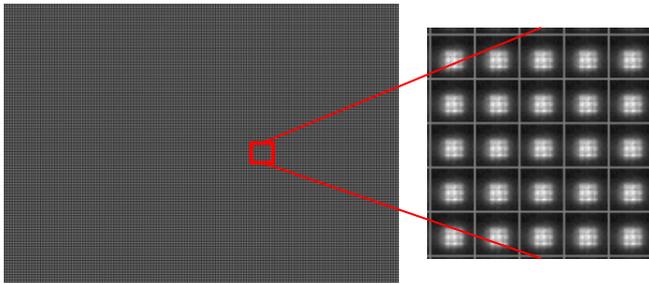
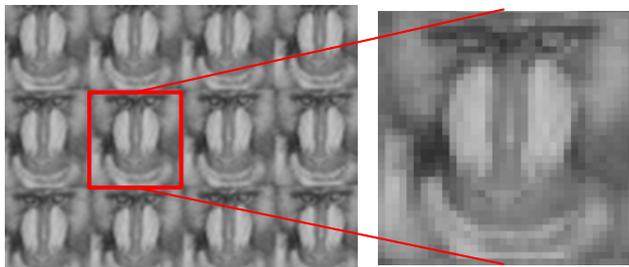


図 17 黒色粉末をのせていない場合に推定した仮想的な画素に対するコード

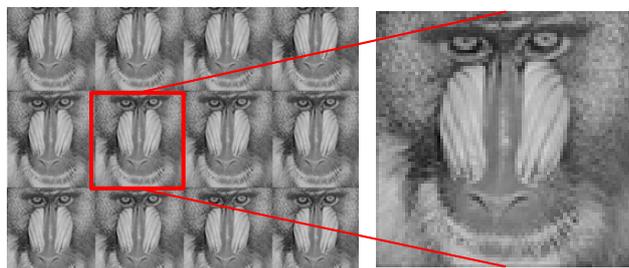
5.2 推定したコードを用いた超解像シミュレーション

提案手法 [1] を用いて、図 18(a) のような観測画像から高解像度画像 (b) を推定するシミュレーション実験を行った。シミュレーションには、5.1 節で推定したコードを使用し、観測画像はシミュレーション上で作成した。超解像の条件を以下に示す。

- 位置ずれ： $(x,y) = (-50,-50) \sim (50,50)$ の範囲をランダムに移動する
- 位置ずれは重ならないようにする
- 観測画像：961 枚
- 観測画像の解像度： $(140,100)$ pixel
- 復元画像の解像度： $(1600,1200)$ pixel



(a) 観測画像



(b) 目的画像

図 18 超解像の観測画像（低解像度画像）と目的画像（高解像度画像）

図 19,20 は、それぞれ図 16,17 で示したコードを用いて作成した超解像画像である。表 1 は、各コードの結果画像における PSNR を示しており、コード化していないコードと比べランダムコードの方が PSNR が高い値

となっていることがわかる。これより、期待されたランダムコードによる精度向上を確認することができた。また、図 19 ではマンダリルの髭部分がぼけているのに対し、図 20 では鮮明になっていることが確認でき。

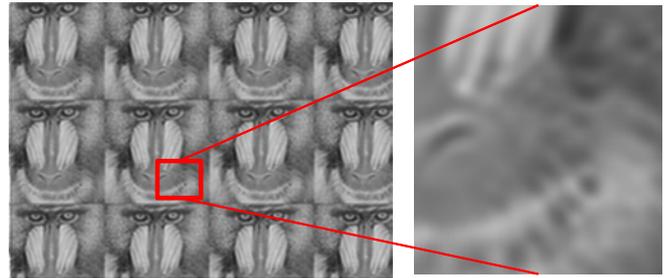


図 19 コードなしの場合の超解像結果

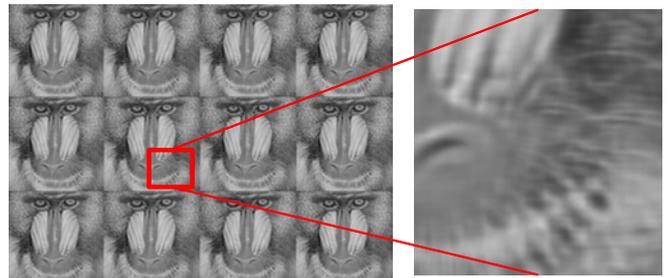


図 20 ランダムコードの場合の超解像結果

表 1 画素形状のコード化による超解像シミュレーション結果

	コードなし	ランダムコード
PSNR[dB]	17.522606	19.198424

6. ま と め

本稿では超解像において、従来研究が撮像素子における各画素を矩形（正方形）のまま扱っていることから原画像の高周波成分を損失していることに着目した。各画素のサンプリングが点サンプリングではなく有限の面積によるサンプリングとなっていることが原因で高周波成分を失っていると考え、撮像素子の上に細かい黒色粉末を振りかけて各画素のサンプリング関数のコード化を行った。黒色粉末を振り掛けた結果、どのようなコードになっているか推定する方法を示し、実験によりコードの推定を確かめた。また、推定したコードを用いたシミュレーション実験を通して提案手法の有効性を確認した。

今後の課題として、仮想的な 1 画素のコード化ではなく、実際の 1 画素のコード化を目指す。そのためには、より大きな画素サイズを持つカメラでコードの実装をする必要がある。また、本稿では複数枚の観測画像の位置ずれ量を既知としたが、実際に撮影した実画像を使用する場合、それらの画像の位置ずれ推定も必要となる。さらに、推定したコードが本当に正しいものか、その評価を行う必要がある。これらの課題を解決し、実機での超解像に重点を置いていく。

文 献

- [1] 笹尾朋貴, 日浦慎作, 佐藤宏介: 超解像のための画素形状のコード化に関する研究, MIRU2010 画像の認識・理解シンポジウム論文集, pp. 1701-1708. (2010)
- [2] 川合康裕, 天野敏之: 固有空間を用いた bplp による super resolution, 電子情報通信学会総合大会講演論文集, p.91 (2004).
- [3] M.C.Hong, T.S. and Katsaggelos, A.K.: Iterative regularized image restoration using local constraints, Proc. IEEE Workshop on Nonlinear Signal and Image Processing, pp.145-148 (1997).
- [4] M.Ben-Ezra, A.Z. and Nayar, S.K.: Jitter camera: High resolution video from a low resolution detector, Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp.135-142 (2004).
- [5] Mohan, A., Huang, X., Tumblin, J. and Raskar, R.: Sensing increased image resolution using aperture masks, IEEE Conference on Computer Vision and Pattern Recognition, 2008. CVPR 2008, pp.1-8 (2008).
- [6] Penrose Pixels; Super-Resolution in the Detector Layout Domain, M. Ben-Ezra, Z. Lin, and B. Wilburn IEEE International Conference on Computer Vision (ICCV), pp.1-8, (2007).
- [7] 田中正行, 奥富正敏: 組合せ画素混合を利用した超解像処理, 電子情報通信学会論文誌 A, Vol.90, No.8, pp.1948-1956 (2007).
- [8] M. Nakashizuka, H. Nishiura, and Y. Iiguni, "Shift-Invariant Sparse Image Representations Using Tree-Structured Dictionaries," IEICE Trans.Fundamentals, vol.E92-A, no.11, pp.2809-2818, Nov. (2009).