



文・構成：  
倉地紀子

## ブラーと コンボリューション

撮影画像におけるブラーがどのようなように生まれるか振り返ると、これは主に光の回折やレンズの誤差などの光学的な要因によって作り出されており、そのメカニズムは非常に複雑だ。そこでこれをシンプルにモデル化するために、ポイント・スプレッド・ファンクション (PSF) という関数が用いられる。PSFは、カメラ・望遠鏡・顕微鏡などのイメージングシステムに固有の関数で、ちょうど天体望遠鏡で星が「点」ではなくぼやけた円に見えるように、ある点光源を撮影した時に光源がどのような広がり方で写し出されるかを表す。通常は光源からの距離を引数とする関数だが、場合によっては方向性をもつため、汎用的に使える2次元平面上の関数に変換される。学術的にいえば、ブラーは「PSFによるコンボリューション」と定義されるものだ。

ある関数によるコンボリューションとは、直感的にはその関数で重み付けをして足し合わせることを意

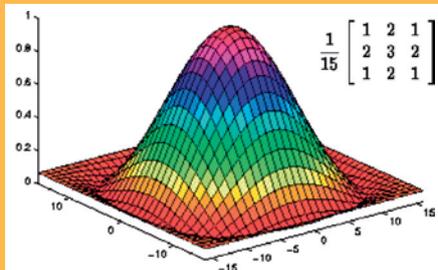
# 撮影画像からモーションブラーを取り除く Computational Photography 第3回

SIGGRAPH2006で発表されたコーデッド・エクスポージャーという手法は、シャッターに工夫を凝らすことで、モーションブラーを効率的に取り除くことを実現した。今回は、この新しい手法のコンセプトとその可能性を紹介する。

### ブラーを作り出すカーネル

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

ブラーを作り出すもっとも一般的なボックスフィルタ。あるピクセルの値にその周りの8つのピクセルの値を足し合わせた平均値を、新たなピクセル値とする



### 式 A-5

$$K(u,v) = \sum_{n=0}^{n-1} \sum_{m=0}^{m-1} k(n,m) e^{-2\pi j(\frac{un}{N} + \frac{vm}{M})}$$

このカーネルは、元のピクセル値の重みを3、その上下左右のピクセル値の重みを2、対角線上のピクセル値の重みを1として、すべてを足し合わせて平均化した値を新たなピクセル値とする。図は、式A-3のイメージ・コンボリューションをフーリエ空間上で表したものの(式A-5をグラフ化)で、高さが周波数を表す。一定周波数以上の周波数は現れておらず、周波数の高い領域の情報が切り取られていることがわかる

### コンボリューション

#### 式 A-1

$$(f * g)(t) = \int f(t)g(t-\tau) d\tau$$

#### 式 A-2

$$(f * g)(m) = \sum_n f(n)g(m-n)$$

2つの関数fとgとのコンボリューションf \* gは、式A-1のように表される。式A-2は、式A-1を離散的な形(変数と実数値ではなく整数値としたかたち)に直したものの

### 式 A-3

$$Q(n,m) = \sum_{a=-\infty}^{\infty} \sum_{b=-\infty}^{\infty} I(n+a, m+b) k(-a, -b)$$

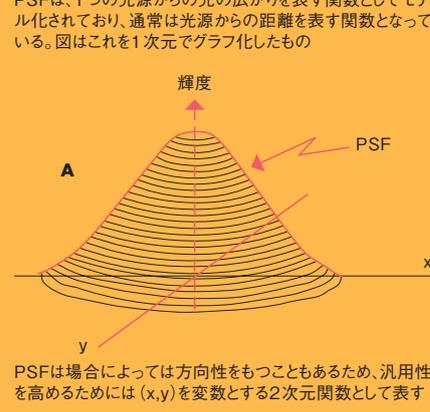
I11	I12	I13	I14	I15	I16	I17	I18	I19
I21	I22	I23	I24	I25	I26	I27	I28	I29
I31	I32	I33	I34	I35	I36	I37	I38	I39
I41	I42	I43	I44	I45	I46	I47	I48	I49
I51	I52	I53	I54	I55	I56	I57	I58	I59
I61	I62	I63	I64	I65	I66	I67	I68	I69

K11	K12	K13
K21	K22	K23

### 式 A-4

$$Q_{57} = I_{57}K_{11} + I_{58}K_{12} + I_{59}K_{13} + I_{67}K_{21} + I_{68}K_{22} + I_{69}K_{23}$$

式A-2を画像に適用したものを、イメージ・コンボリューションと呼ぶ。画像Iの関数Kによるコンボリューションを表したのが、画像Qとなる



## 参考文献

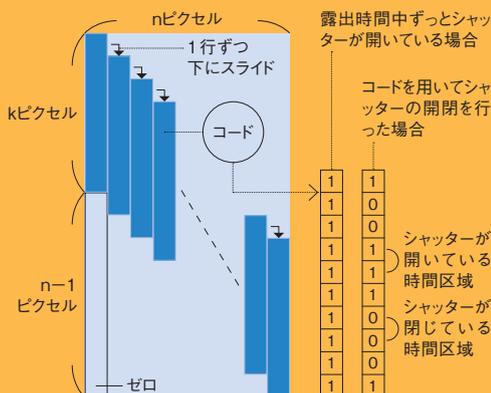
論文1「Coded Exposure Photography: Motion Deblurring using Fluted Shutter」  
Ramesh Raskar, Amit Agrawal, Jack Tumblin, SIGGRAPH2006  
project by Mitsubishi Electric Research in Boston  
<http://www.merl.com/people/raskar/deblur/>

味しており、これを画像生成に適用したものはカーネルを用いたフィルタリングに相当する。カーネルのなかで最も代表的なボックスフィルタでは、あるピクセルに対してその近傍の8つのピクセルの値を足し合わせて平均値をとる。このようなカーネルは周波数の低い部分だけを抽出することからノイズの軽減やアンチエイリアスに役立つ。その一方で、高周波数の部分は切り落とされてしまうため、ブラーのなかった画像から物体本来の形を表す画像を復元するのが非常に難しい。

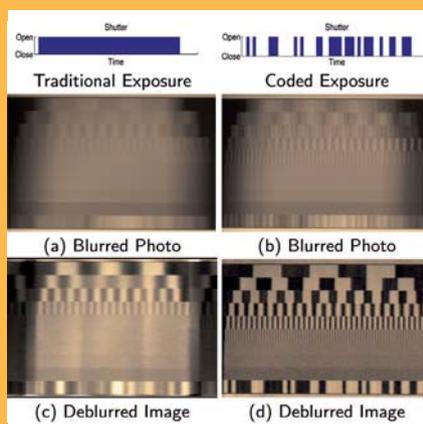
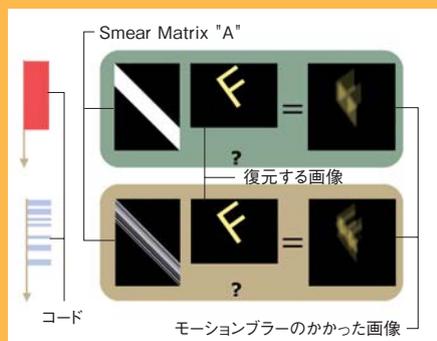
## モーションブラー

PSFはブラーを作り出す光学的な要因だけを考慮していたが、もう一つの要因にレンズが露出している間のカメラと被写体の位置変化がある。一般的にモーションブラーと呼ばれるものは、後者の要因によって作り出されたブラーを指す。カメラによる撮影では、レンズが露出している間に、レンズのそれぞれの位置を通過した光の総量が、イメージプレートの対応する位置に記録される。露出している間カメラも物体も静止している場合には、レンズの同じ位置を常に同じ強さの光が通過するが、どちらかが動いている場合にはレンズの同じ位置を通過する光の強さは変化する。したがって、イメージプレートの各位置に記録される光の量は、静止している場合の光の量にある関数で重みをつけてすべて足し合わせたものに相当し、これはちょうど時間を変数とした関数によるコンボリューションにあたる。

モーションブラーも考慮した場合



コードによるコンボリューションを表す行列A (Smear Matrix) は、1次元のモーションブラーを表す行列として作成する。ブラーするピクセル数をk、物体が占めるピクセル数をnとすると、行列の1列目には、上からkピクセルの領域にコードが入り(シャッターが開いている区間のコードは1、閉じている区間のコードは0)、それに続いてn-1ピクセルの領域にゼロが入る。1列目の要素数(行列の行数)は、コードがいかに細かく区切られているかによって決まる。2列目以降は、1列目を1行ずつ下にスライドすることで作成する。これによって、時間軸に沿ったコンボリューションの計算を表す行列が作成される。この行列は、実質的にはn+k-1ピクセルの領域を時間軸に沿って足し合わせる計算と同じ演算を行う。ただ、物体を時間軸に沿ってスライドさせるかわりに、行列Aではコードを時間軸に沿ってスライドさせることによって、時間軸に沿ったコンボリューションの計算を行う



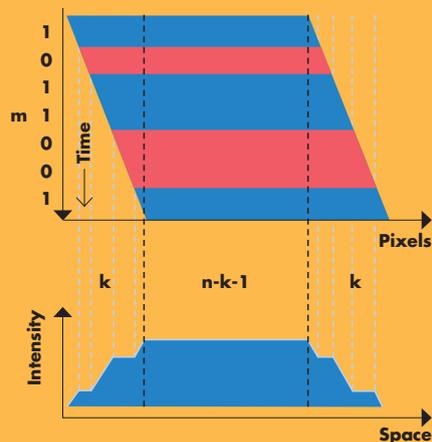
左列は、1枚の画像を撮影する間ずっとシャッターを開けて撮影した画像(a)と、(a)を用いてブラーのかかっていない画像を復元したもの(c)。右列は、1枚の画像を撮影する間にシャッターを短い時間間隔で開閉しながら撮影した画像(b)と、(b)を用いてブラーのかかっていない画像を復元したもの(d)。画像(c)ではブラーがうまく取り除かれていないが、画像(d)ではブラーが取り除かれ、物体本来の形状がほぼ正確に復元されている



露出時間の間、物体が一方向に進んでいる場合には、時間軸に沿ったコンボリューションは、ブラーのかかっていない物体そのものの形を表す画像とコードとのコンボリューションとしてモデル化される。上が通常の撮影、下がコードに従ってシャッターの開閉し撮影を行った場合を示す

コードを用いて作成された行列A(「1」が入っている部分が白、「0」が入っている部分が黒)を掛け合わせることで、垂直方向にブラーのかかった画像が生成される。上列が通常の撮影、下列がコードに従ってシャッターを開閉して撮影を行った場合を示す

## コードド・エクスポージャー



モーションブラーが作り出されるプロセス(時間軸に沿ったコンボリューション)を1次元で示したもの。進行方向にnピクセルの長さをもつ物体が、露出時間mの間にkピクセル進んだとすると、時間軸に沿ったコンボリューション(露出時間mの間にわたって各時間のピクセル値を足し合わせる)によって撮影画像の「k+(n-k-1)+k=n+k-1」ピクセルにブラーが発生する。モーションブラーを取り除く作業は、このコンボリューションの逆演算を行って、撮影画像のn+k-1ピクセルの情報から、コンボリューションが起こる前のnピクセルの情報を復元する作業に相当するものだ。縦軸の1001101は、後述するコード(露出時間の中にシャッターを開閉するタイミング)を示している

## コードド・エクスポージャー

時間軸に沿ったコンボリューションは、物体が静止している状態で撮影された画像の中で物体の進行方向に並ぶ近傍のピクセルを足し合わせて平均化する作業、言い換えるとボックスフィルタリングの1Dのケースによく似ている。すなわち周波数の高い部分の情報が失われており、いったん失われてしまった情報を逆変換によって復元することはできないため、ブラーのかかった画像からブラーのかかっていない物体本来の形を表す画像を復元することは不可能となる。コードド・エクスポージャーとは、時間軸に沿ったコンボリューションを、周波数が高い部分の情報も保つことのできるフィル

タリングにすることを考え出された手法なのだ。

通常のカメラ撮影では、一枚の画像を撮影する間シャッターをずっと開けておくのに対して、コーデッド・エクスポージャーではシャッターを何度も開閉する。この撮影画像は、見た目にはシャッターをずっと開けて撮影した画像とほぼかわらないブラーのかかった画像となるが、この画像では周波数の高い領域の情報も失われない。すなわち、コーデッド・エクスポージャーで撮影された画像を用いると、ブラーのかかっていない画像を正確に復元できる。

また、周波数の高い領域の情報がどれだけ多く保たれているかは、シャッターを開閉するタイミングによつて変わってくる。コーデッド・エクスポージャーの「コード」とは、このタイミングのことを指す。論文では、探し出した最適なコードを用いると、非常に高い汎用性をもって、ブラーのかかっていない物体の形状そのものを表す画像が復元できることを実証した。

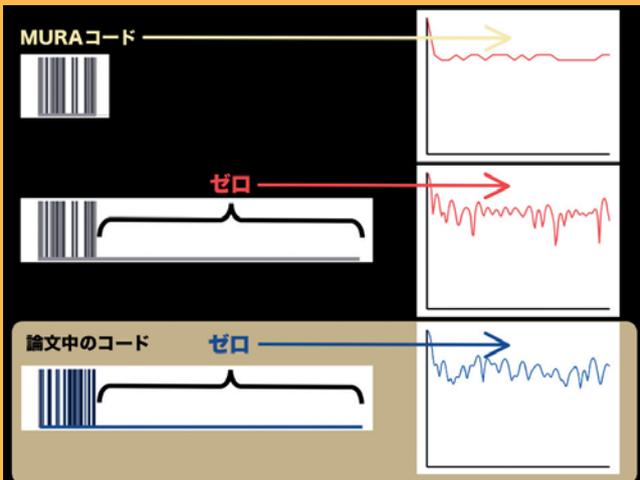
## モーションブラー！ マトリクスと コードの解析

論文では、モーションブラーを行列で表すことを考えており、モーションブラーのかかっている画像にこの行列を掛け合わせたものが、モーションブラーのかかった画像となっている。つまりブラーを取り除くためには、モーションブラーのかかった画像にこの行列の逆行列を掛ければよい。ここでいうモーションブラーを表す行列は、時間軸に沿ったコンボリューションを表す演算を意味して

## コードの解析



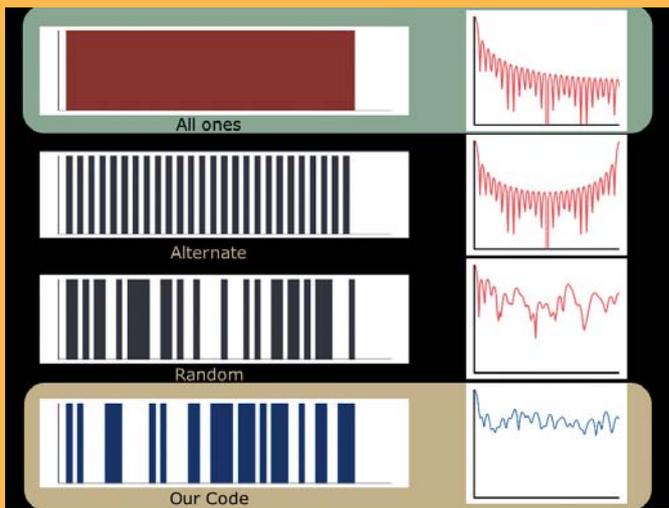
コーデッド・エクスポージャーカメラ。手前の液晶シャッターを開閉させてコードを作り出す。専用ドライバを内蔵することで、持ち運び可能となった  
(ハード設計：Hideaki Nii)



安定性のあるコード自体は、今回の論文以前にすでに考案されていた。その代表的な例がMURAコード(上)で、このコードを信号と考えてフーリエ空間上でグラフ化すると、非常に理想的な波形をしている。しかし、モーションブラーを作り出すための信号は、コードの後にゼロをくっつけたものだ。このようにゼロをくっつけた状態でグラフ化すると、MURAコードから作成された信号の波形は不安定になる。これに対して論文中のコード(下)は、ゼロをくっつけた状態でグラフ化しても安定した波形になっているのが大きな特徴だ



ブラーのかかった撮影画像にはノイズが含まれている。行列Aの逆行列がこのノイズをダイレクトに反映してしまうと、逆行列によって作り出される画像、つまりブラーのかかっていない画像にもノイズが多く含まれてしまう。行列Aの変換において、変化前のデータを変換後のデータとの間にどれだけ依存性があるかを示すものとして、行列Aから作成されるCovariance Matrixと呼ばれる行列がある。これは、この行列が対角行列に近いほど、変換前と変換後のデータ同士の間には依存性がないと判断されるものだ。画像は左から、ボックスコード、MURAコード、論文中のコードそれぞれに対する行列AのCovariance Matrixを示しており、順に対角成分以外の成分が減っていることがわかる。つまり、論文中のコードから作成された行列Aは、逆変換によって撮影データのノイズをもっとも伝達しにくい行列となっている。したがって、ブラーのかかっていない画像をもっとも正確に復元できるコードだと言える



さまざまなコードをフーリエ空間上でグラフ化したもので、横軸に周波数、縦軸に関数値の対数をとった値を表す。最上段はコードがすべて1のケース(ボックスコード)、これは露出時間の間ずっとシャッターを開けて撮影した結果に相当する。周波数が大きくなるにつれて関数値は小さくなっており、周波数が高い部分の情報が失われることを示す。また、このグラフでは波の起伏が非常に激しく不安定になっているが、このような関数の逆関数を求めることは非常に難しく、それゆえに行列Aの逆行列を算出することも難しい。上から2段目はシャッターの開閉を一定間隔で行った場合、3段目はシャッターの開閉をランダムに行った場合に相当する。いずれも周波数が高い部分の情報が失われる傾向は減っているが、やはり波の起伏が非常に激しく不安定になっており、行列Aの逆行列を算出することは難しい。4段目は、論文で考案されたコードを用いた信号を表している。高周波数の部分の情報も失われることなく波形も安定しているため、行列Aの逆行列を求めやすい

## ブラーのかかっている 画像への復元

行列Aは一方方向へのブラーのみに対応しているが、実際のシーンでは複数方向にブラーがかかっていることも多い。その場合は撮影画像の

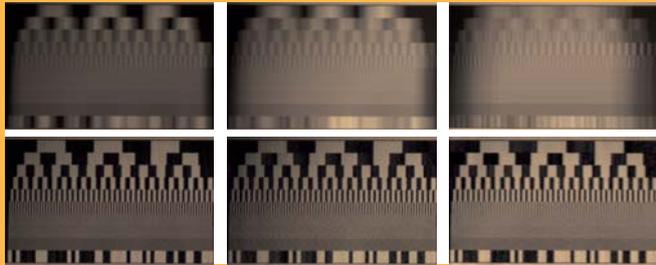
情報を見失わないことが不可欠だ。行列Aの特徴は二列目のベクトルによつて決定されるので、列目が表す信号(横軸に二列目の各要素の行数、縦軸にその要素がもつた1の値をプロットした関数)をフーリエ変換したグラフを作成し、フーリエ空間上でこのグラフを解析する。こうして安定したコードを求めつつ、さらに撮影画像に含まれるノイズを反映しにくいコードであることも検証していく。

ただし、ブラーのかかっている画像へと正確に復元するためには、行列Aの逆行列を正確に算出しなければならぬ。そのためには、行列Aによる変換が高周波数の領域の情報を見失わないことが不可欠だ。また、正確に復元するためには、行列Aの逆行列を正確に算出しなければならぬ。そのためには、行列Aによる変換が高周波数の領域の情報を見失わないことが不可欠だ。また、正確に復元するためには、行列Aの逆行列を正確に算出しなければならぬ。そのためには、行列Aによる変換が高周波数の領域の情報を見失わないことが不可欠だ。

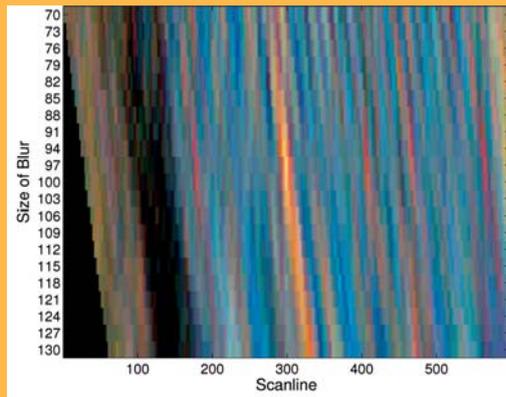
情報だけでは画像を復元できないため、ブラーのかかっている領域ごとに、ブラーのかかっている物体が動いている方向(モーションライン)をデザイナーが指定するという方法が採られた。ブラーを取り除く作業は、それぞれのモーションラインごとに行列Aを作成することによって行われ、いったん行列Aが作成されれば、ブラーのかかっている領域に行列Aの逆行列を掛けることによつて、ブラーを取り除くことができる。ただし、本来行列Aのコードの入っている行数 $m$ (コード数)はブラーのかかっているピクセル数 $k$ に等しいのが理想的だが、実際には必ずしも等しくはない。したがって、撮影画像に $m/k$ でスケールを掛けてからAの逆行列を掛け、その後 $k/m$ のスケールをかけてブラーを取り除いた画像を復元する。

前述したように、この方法では時間軸に沿ったコンポジットシーンをPSFには依存しない形で復元しているため、シーンの中に複数の異なるPSFをもつ物体が存在しても、そのいずれに対しても同じ行列Aを用いてブラーを取り除く作業を行うことが可能だ。また、動いている物体の前面を遮る物体が存在する場合や、動いている物体の背景が複雑なテクスチャをもっている場合にも、かなりの自由度をもつて、ブラーをうまく取り除くこともできる。このように非常に汎用性が高く、複雑なシーンにもフレキシブルに対応できることが、この手法の最大の利点だ。また、PSFそのものを復元する場合には、非線形で複雑な計算工程が必要とされるが、今回の方法では逆行列を掛け合わせるという線形でシンプルな計算だ

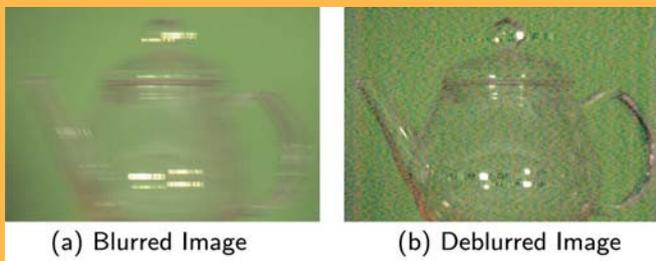
## ブラーを取り除いた画像の復元



理論的には、モーションラインに沿ってブラーのかかっている長さ $k$ とコードの長さ $m$ が近いほど、より正確にモーションブラーを取り除くことが可能だ。画像は、 $m=52$ に対して、左から $k=44, 78, 192$ に設定してブラーを取り除いたもの。 $k=44$ の場合以外は、周波数の高い領域の詳細がうまく復元されていないことがわかる

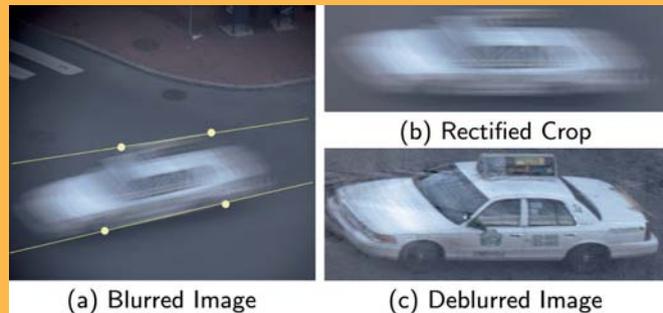


$m$ と $k$ の関係だけでなく、 $k$ の値をいかに正しく予測するかということも、正確にブラーを取り除くための鍵となる。画像は、同じ1本のスキャンラインに対して、 $k$ の値を変えてブラーを取り除いた結果。 $k$ の値が変わることによって、ブラーを取り除く精度が大きく変わっている。この場合には $k=97$ の場合にもっとも正確にブラーを取り除くことができる



ブラーのかかっている物体がスペキュラ反射や光の透過などを起こす場合には、その領域で光学的な要因に依存するPSFが一定でなくなってしまう。論文の手法は、光学的な要因に依存するPSFは空間内で一定だという仮定のもとに成り立っているため、このようなシーンではうまくブラーを取り除くことができない。画像は透明なガラスのボットのブラーを取り除こうとしたもので、アウトラインはなんとが復元できているものの、スペキュラ反射によるハイライトが強い部分では本来のスペキュラ反射の見え方がうまく復元されていないことがわかる

正しく復元するPSFは空間内で一定だと仮定している。つまり、ブラーを取り除く領域での光の分布が均一であることが前提であるため、ブラーのかかっている物体がスペキュラ反射や光の透過を起こす場合も、正確にブラーを取り除くことは難しい。とはいえこれらの制限はあるものの、撮影方法やデバイスへの工夫を凝らすことによつて、長年課題とされてきた問題をシンプルで効率的な手法として実装したことの意味は大きく、適したデバイスの開発なども含めて、今後の発展に期待がかけられている。



実際にモーションブラーのかかった画像からブラーを取り除くためには、(a)物体の動いている方向(モーションライン)を指定する(4点をクリック)。(b)モーションラインに平行になるようにブラーを取り除く領域を切り取る。(c)bを行列Aのサイズに合わせて変形、行列Aの逆行列を掛けて元の解像度に戻す。この結果、モーションブラーの取り除かれた画像が生成される

モーションブラーを取り除く物体の手前に、物体を遮る他の物体(オクルーダー)がある場合でも、モーションラインに沿ったオクルーダーの幅 $d$ が、ブラーのかかっている長さ $k$ よりも小さい場合には、論文の手法を用いてモーションブラーを取り除くことができる

