

# コンピュータショナルフォトグラフィ —ライトフィールドカメラ Lytro の動作原理とアルゴリズム—

蚊野 浩 (Hiroshi Kano)

京都産業大学 コンピュータ理工学部  
Kyoto Sangyo University, Department of Computer Science and Engineering

**要旨** コンピュータショナルフォトグラフィは、コンピュータ技術とレンズ技術、撮像素子技術が高度に融合した新しい写真技術である。Lytro はその代表的な製品であり、写真画像を撮影した後に、その写真の焦点位置を自在に変えることができるカメラとして知られている。Lytro はライトフィールドカメラと呼ばれる。その画像センサーとマイクロレンズアレイで構成される撮像部が、カメラに入射する光線集合であるライトフィールドを取得する。ライトフィールドから最終画像を生成する処理は、光線集合に対するレンズの働きを計算することに相当する。本稿では、コンピュータショナルフォトグラフィを概観した後、Lytro の動作原理と画像を生成するアルゴリズムを解説する。

## 1. はじめに

デジタルカメラは撮像素子表面に鮮明な光像を形成し、その光像に忠実なデジタル画像を生成する装置である。そこでは、顔画像の検出や美顔処理など、高度な技術が使われている。しかし、レンズが形成する光像の忠実再現という写真撮影の原理に変わりはない。

この原理から離れて、レンズ・撮像素子・画像処理のシステムを再構築し、従来不可能であった撮影手法を可能とする技術分野をコンピュータショナルフォトグラフィ (CP) [7]とよぶ。CP の代表的手法としてライトフィールドカメラ[1][2][3][5]が知られている。これはライトフィールド (光線空間) として多数の光線を取得し、その光線集合に一種の画像処理を施すことで最終画像を得るカメラである。主な機能は、後処理によって焦点距離を変えた画像を生成するリフォーカス機能である。

本稿では、CP の概要に触れた後、市販のライトフィールドカメラである Lytro[4]について詳しく解説する。

## 2. コンピュータショナルフォトグラフィ

CP とはいうが、整理された技術体系があるわけではない。現状、いくつかの具体的な手法の集まりである。

CP とよばれるものの中で、比較的、古くから知られているものに、Wavefront コーディング[8]がある。これは、図 1 に示すように、撮像レンズの一部に位相板とよばれる光学素子を挿入する。この位相板の働きによって、レンズを通過した光線はカメラの焦点位置近傍で

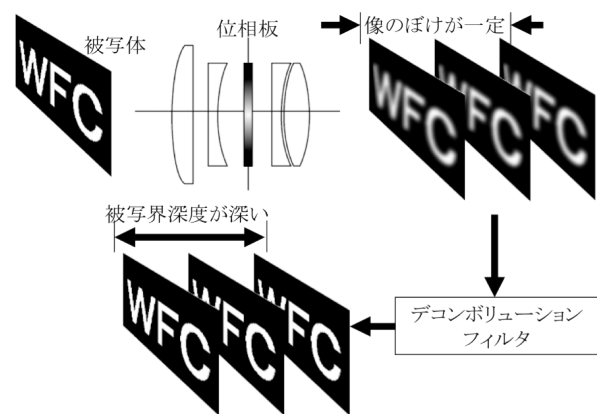


図1 Wavefront コーディングによる被写界深度の拡大

像のぼけが一定の状態を、比較的長く保つようになる。この像をデコンボリューション (画像復元) することで、シャープな像を得る。このような手法によって、通常よりもはるかに被写界深度が深い写真撮影が可能になる。

写真撮影における失敗の多くは、ピンぼけと手ぶれである。Wavefront コーディングはピンぼけを防ぐ技術である。一方、手ぶれを防ぐための CP 手法として符号化露光[7]が知られている。最も性質が悪いぶれは、撮像面が等速で移動するぶれである。これによって、画像中の高周波成分が急速に失われる。逆に性質が良いぶれというものがある。これは、チャープ信号のように高周波から低周波までを含むぶれである。実際にこのようなぶれを撮像素子に与えるのではなく、ぶれが発生する撮影期間中の露光をチャープ信号などで変調する、という考え方である。このように露光を変調して撮影すると、一見、手ぶれ写真に見えるが、実は、元の信号が十分に残っている。そこで、画像復元を行うと、きれいにぼけが修復される。

### 3. Lytro の画像センサー部

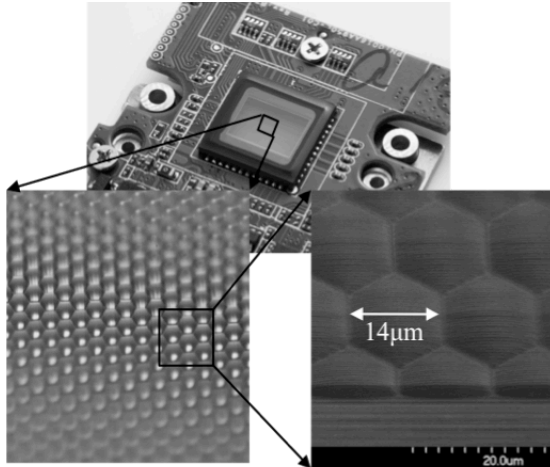


図2 画像センサー部の拡大写真  
(写真:日経BP社,  
豊通エレクトロニクス ヴァン・パートナーズ,)

図2にLytróの画像センサー部の拡大写真を示す。画像センサーはデジタルカメラ用CMOSセンサーであり、その画素ピッチは $1.4\mu\text{m}$ である。画像センサーをハニカム構造のマイクロレンズが覆っている。そのピッチは $14\mu\text{m}$ である。画像センサーの $3280\times 3280$ 画素の領域が $330\times 380$ 個のレンズによってカバーされており、一つのマイクロレンズの直径は、おおむね10画素分である。

マイクロレンズは保護ガラスと一体化しており、CMOSセンサー上に若干の空間をおいて配置されている。保護ガラスの上面側が平坦で、底面側がマイクロレンズアレイになっている。保護ガラスとマイクロレンズアレイの厚さは約 $430\mu\text{m}$ である。また、CMOSセンサー表面にはベイヤー型のカラーフィルタアレイが配置されている。

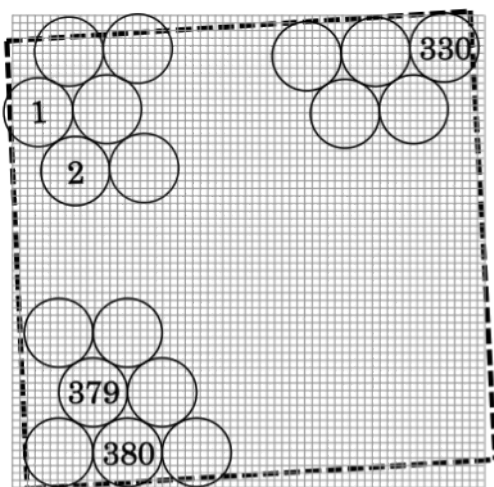


図3 画像センサーと  
マイクロレンズアレイの配置

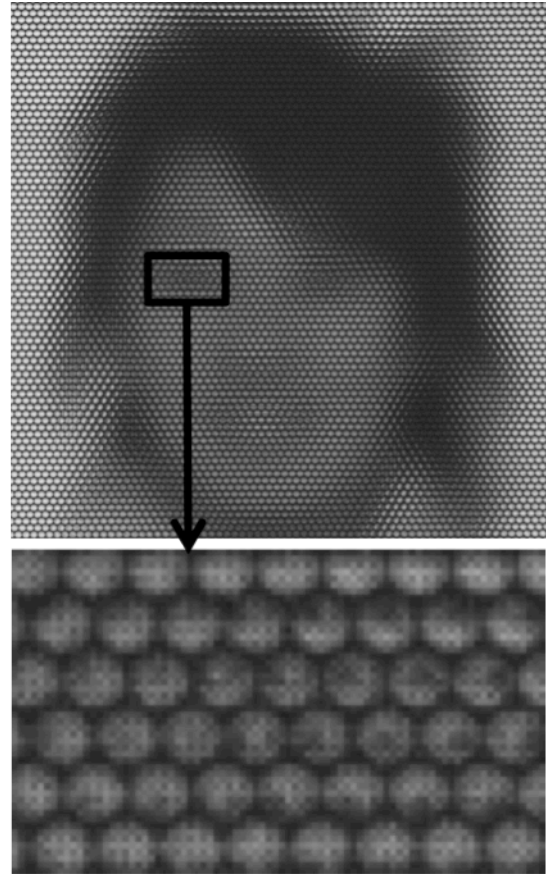


図4 生画像の一部とその部分拡大

図3に画像センサーとマイクロレンズアレイの位置関係を示す。これらは、完全には整列していない。カメラの筐体ごとにずれと傾きが異なる。これは、マイクロレンズを画像センサーに装着するとき生じる、機械的な誤差が原因であると思われる。

図4にLytróで取得される生画像の一部とその部分拡大を示す。生画像にはマイクロレンズアレイの構造が強く反映されている。拡大画像に見られる円構造が一つのマイクロレンズに対応する。その直径が約10画素であることを、この画像からも確認することができる。円構造の中のそれぞれの画素が、後述するように、一本の光線に対応する。マイクロレンズの境界付近に位置する画素は暗くなっており、その領域では光線の取得が難しいことを示す。

### 4. ライトフィールドの取得

画像センサー部と主レンズによってライトフィールドが取得される様子を図5に示す。図において、Aの位置にある被写体が主レンズによってマイクロレンズアレイの位置に焦点を結ぶとする。このとき、被写体から発してマイクロレンズに到達した光線は、その下に位置す

る画素によって方向と明るさが記録される。一つのマイクロレンズに対応する全ての画素を平均化することで、Aの位置に焦点を合わせた粗い写真画像が生成される。

図5で、Bの位置にある灰色マークに注目すると、この位置を通過する3本の光線は、異なるマイクロレンズを介して画像センサーに記録される。それらの画素値を平均化すると、Bに焦点を合わせた写真画像の画素を生成できる。これは、Bの位置に焦点を合わせるとAの像がぼけるという現象を、ライトフィールドを用いて計算したことに対応する。

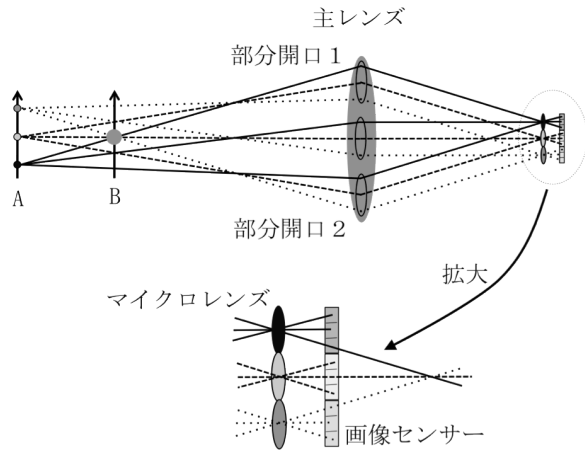


図5 ライトフィールドの取得

図5のマイクロレンズがカバーする3つの画素で、最も下に位置する画素に対応する光線を主レンズまで追跡すると、部分開口1と記述した領域に達する。したがって、これらの画素だけで生成される画像は、部分開口1を通過した光線による像である。同様に、マイクロレンズがカバーする画素で、最も上に位置する画素だけで生成される画像は、部分開口2を通過し

た光線による像である。このようにマイクロレンズがN個の画素をカバーするとき、マイクロレンズに対して同じ位置にある画素を配列してできるN個の小画像は、主レンズをN個の部分開口に分割して取得されるN個のステレオ画像群を形成する。

### 5. ライトフィールドのレンダリング

Lytroが取得する生画像であるライトフィールドから最終の写真画像を生成するプロセスをライトフィールドレンダリングとよぶ。ここでは、2種類のレンダリング手法を説明する。第一の方法は、レンダリングする焦点距離に応じて、N個のステレオ画像を適量平行移動させた後、それらを平均化する方法である。これをシフト加算法とよぶことにする。第二の方法は、取得した光線を、適当な焦点距離に設定した画像面に投影する方法である。これを光線束投影法とよぶことにする。

シフト加算によるレンダリングを説明する。4章で述べたように、Lytroが取得するライトフィールドを多数のステレオ画像群に分解することができる。図6の左側部分がその様子を説明している。全てのマイクロレンズから同じ位置にある画素を抽出して再配列する。このようにして構成された小画像群は、主レンズの異なる部分開口から被写体を観察した画像群である。したがって、開口位置と被写体までの距離に応じた視差を有する。これらのステレオ画像群を、仮想的な焦点距離に合わせて、適切な量だけ平行移動させる。そして、移動後の画像群を加算平均する。移動量に応じた奥行きに存在する被写体は像の位置が合うのでシャープな像を形成し、そうでない被写体の像はぼける。なお、実際のカメラでピントを前後させた場合

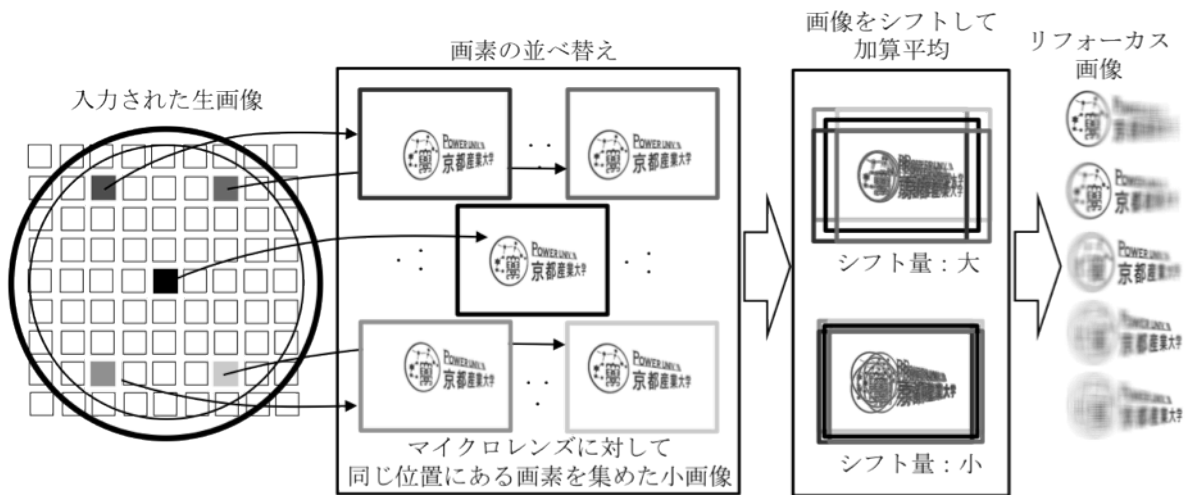


図6 ステレオ画像群への分解とそのシフト加算によるレンダリング

には、焦点が合う位置が変わるだけでなく、像の大きさが若干変化する。ここでは、この効果は考慮していない。

加算シフトによるレンダリングアルゴリズムは、単純で計算コストが小さい。しかし、最終画像の画素数はマイクロレンズの画素数と等しくなるので、通常のデジタルカメラで撮影した画像と比較すると、非常に小さいものになる。また、マイクロレンズと生画像の画素の位置関係は、マイクロレンズごとに微妙に異なる。したがって、画素の再配列において、画素値の補間が必要になる。その結果、最終画像の解像度を少し劣化させてしまう。

シフト加算によるレンダリングは実装が容易であるが、最終画像の解像度はそれほど良くない。より望ましい画像を生成する光線束投影法について説明する。

図7の上部に、画像センサーとマイクロレンズアレイによって光線を取得する様子を、より詳しく説明した。4章では、画像センサーの各画素が、一本の光線を取得するように説明したが、より正確には、ある程度の幅を持つ光線束が取得される。光線束投影によるレンダリングは、図7の下部に示すように、取得された光線束を、設定された焦点距離にある仮想的な画像面に投影する手法である。

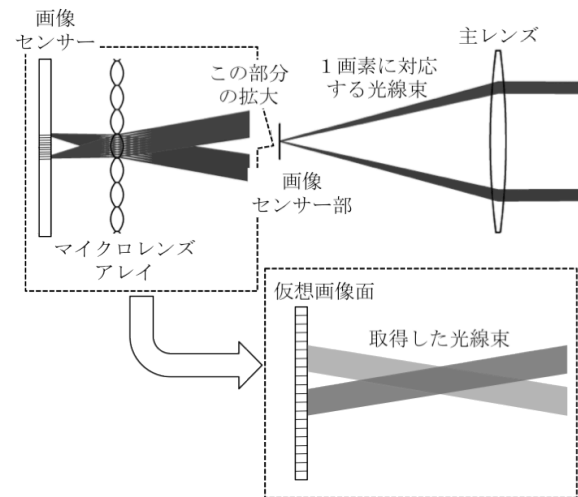


図7 光線束の投影によるレンダリング

各画素が取得する光線束を次のように計算することができる。図8は画像センサー部の拡大図である。ここで、画像センサー面とマイクロレンズアレイ面の距離がマイクロレンズの焦点距離 $f$ に等しく、画素は点であるとする。図中の他の寸法記号の意味を次に示す。

- $a$ : マイクロレンズの直径。
- $q$ : 注目画素の位置。それをカバーするマイクロレンズの光軸からの符号付き距離で示す。

- $r$ : マイクロレンズ光軸と主レンズ光軸の距離。
- $\theta$ : 取得される光線束と光軸のなす角度。

この時、画素によって取得される光線束は平行光になり、光軸に対して $\theta = \arctan(-q/f)$ で交わる。光線束の直径は $a \cdot \cos \theta$ 、位置はマイクロレンズアレイ面において主レンズ光軸から $r$ の距離になる。画素が面積を持つことや、画像センサー面とマイクロレンズアレイ面の距離が一般の場合を考慮すると計算は複雑になるが、同様の手順で計算することが可能である。

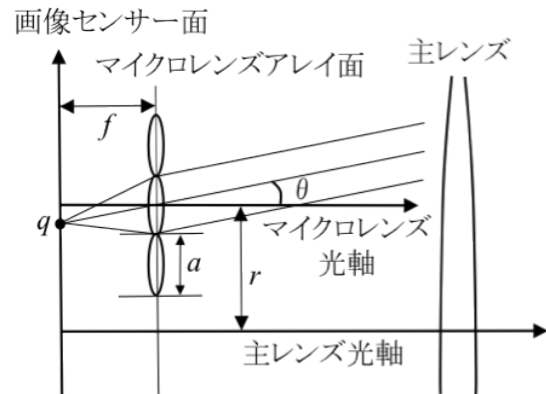
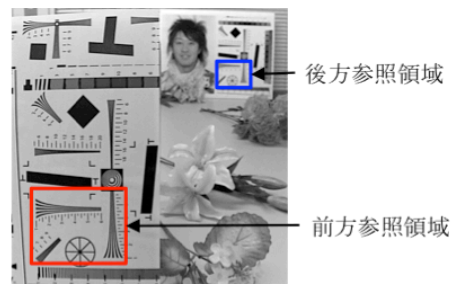
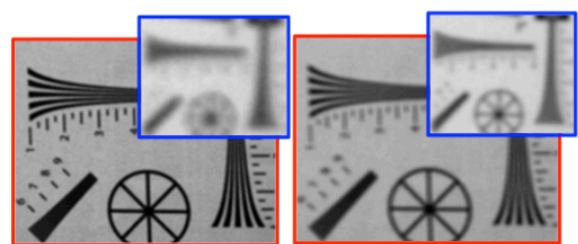


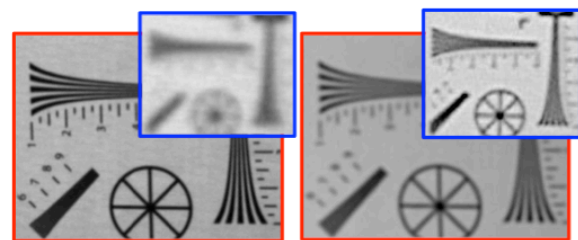
図8 画素が取得する光線束



(a) サンプル画像の全体



(b) シフト加算によるレンダリングで前方と後方にピントを合わせた場合の、参照領域の画像



(c) 光線束投影によるレンダリングで前方と後方にピントを合わせた場合の、参照領域の画像

図9 レンダリング画像の比較



図9に、シフト加算法と光線束投影によるレンダリング画像の比較を示す。図9(a)はサンプル画像の全体である。図中の前方参照領域はLytroから30cm程度、後方参照領域は2m程度の距離にある。(b)と(c)を比較することで、光線束投影によるレンダリング画像の解像度が高いことがわかる。なお、この実験を含めて、Lytroの出力ファイルを解析するためにオープンソースのlftool[6]を利用した。

## 6. Web画像のレンダリング

Lytroでシーンを撮影した後、本体をパソコンに接続すると、数十秒の処理時間を経て、2つのファイルが生成される。一つは図4に示した画像センサーの生画像である。もう一つは、複数のJPEG画像と20×20画素のデプス画像を一つにラップしたファイルである。

複数のJPEG画像は、最も手前に焦点が合った画像、中間位置に焦点が合った画像、最も奥に焦点が合った画像など、焦点位置が異なる画像である。JPEG画像の枚数は、場面の奥行きによって変化する。奥行き範囲が大きいシーンでは10枚程度、奥行き範囲が小さいシーンでは4枚程度であった。これら、焦点位置が異なるJPEG画像は、本体をパソコンに接続した時に、パソコン上で動作するソフトウェアによってライトフィールドレンダリングされたものである。

これら複数のJPEG画像とデプス画像は、Webブラウザでリフォーカス画像を生成す

るために利用していると考えられる。Lytroは、撮影した画像をWeb画面で見ながら、ユーザーがマウス操作で焦点位置を指定して、リフォーカスさせることができる。そのレスポンスは瞬時である。一方、5章で説明したレンダリング手法は、それほど複雑な処理ではないが、それでもリアルタイムで実行することは難しい。また、マウスクリックした位置の距離を計測する処理も必要である。Lytroはリアルタイムにリフォーカス画像を生成するために、あらかじめ、焦点位置が異なった複数のJPEG画像と1枚のデプス画像を計算しているのだと推察される。

図10に、推察したWebでのリフォーカス画像生成のしくみを示す。ユーザーが画面上でリフォーカス位置を指定すると、指定された被写体までの距離をデプス画像から読み出す。そのデプスの近傍に焦点距離を持つ2枚のJPEG画像を取得する。リフォーカス画像は、これらの中間的な画像であるから、適切な加重を与えて、平均することでリフォーカス画像を生成する。

2枚のJPEG画像を加重平均する処理では、中間画像はそれらよりもぼけた画像になる。それを改善するために、焦点位置が異なる3枚のJPEG画像を用いて、中間画像の高域周波数を強調することも可能である。2枚から合成する方法であれ、3枚から合成する方法であれ、ここで述べた処理は高度なものではない。Webブラウザ上でリアルタイム処理を実現することが可能である。

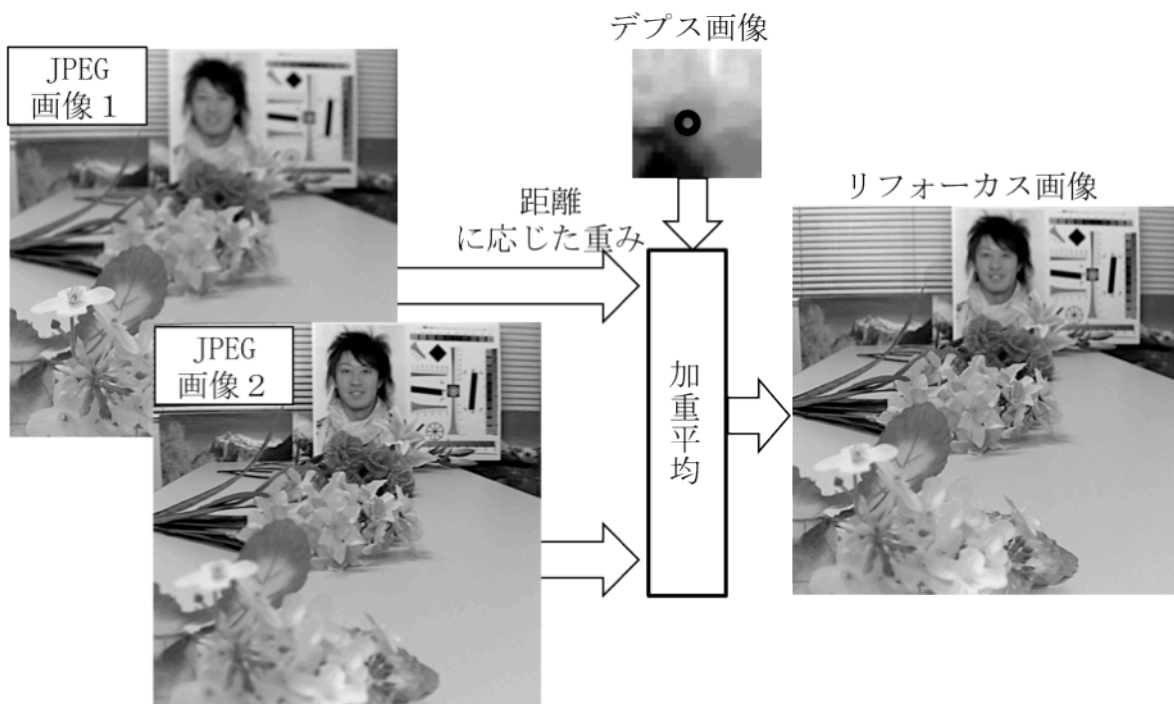


図10 Webでのリフォーカス画像生成のしくみ

## 7. Lytro の課題

Lytro は画期的なカメラである。しかし、未来的なカメラにとどまっておらず、まだ、一般ユーザーに広まっていない。ここでは、Lytro の課題について、何点か記述する。

最も大きな課題は、最終画像の画素数あるいは解像度が小さいことである。一つのマイクロレンズが直径 10 画素程度をカバーしていることから、通常のデジタルカメラと比較して、縦横の画素数が 10 分の 1 程度になる。しかも、Lytro は画像センサーの画素のかかなりの割合を利用できていない。Lytro の画像センサーは  $3280 \times 3280$  画素で、マイクロレンズアレイは  $330 \times 380$  であるから、マイクロレンズアレイあたりの画素数は 85.8 画素になる。一方、図 3 などを観察すると、マイクロレンズあたりの有効な画素数は 50 画素程度である。従って、40% 程度の画素が無駄になっていると思われる。

マイクロレンズの境界に位置する画素は無駄になっているが、マイクロレンズの周辺に位置する画素は記録する光線の質が悪くなっている。図 11 は、マイクロレンズに対して、5 カ所の画素から構成される小画像を示したものである。この原稿では画質を詳細に確認することは難しいが、1-3 の画像は鮮明で明るい。それに比べて、1-1 の画像は全体に赤みがかかり、モアレやノイズが見られる。光量も少ない。3-3 の画像は全体に青みがかかり、1-1 と同様にモアレやノイズによる画質の劣化がある。

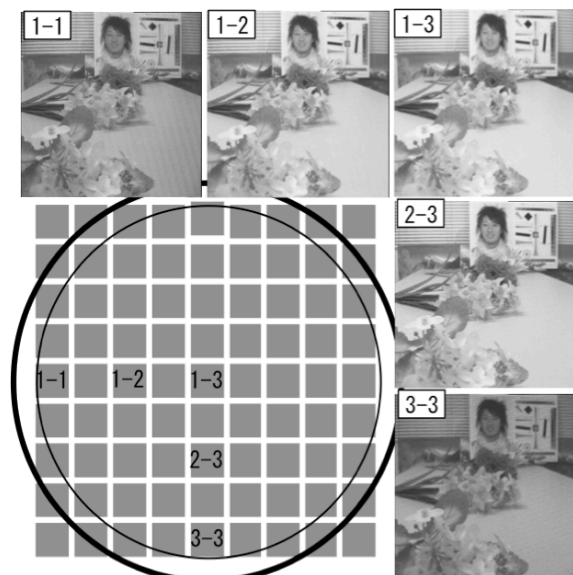


図11 画素位置による光線状態の比較

Lytro の特徴であるリフォーカス機能を楽しむためには、Lytro に慣れる必要がある。Lytro の主レンズは直径が 25mm 程度で、画像センサーは  $4.6\text{mm} \times 4.6\text{mm}$  程度である。このサイズのレンズと画像センサーでぼけ味を楽しむためには、通常の構図の中に、極端な近景（マクロ撮影）を入れるなどの工夫が必要である。そのため、通常の写真撮影では Lytro を生かすように思える。撮影後のリフォーカス機能は、静止画ではなく、動画撮影で生きてくるかも知れない。

現在の Lytro は、筐体ごとに、さまざまなキャリブレーションを行っていると思われる。最も重要なキャリブレーションは、画像センサーとマイクロレンズアレイのアラインメントである。これを正確に合わせなければ、高画質なレンダリングはできない。

## 謝辞

本調査解析は科研費 (23500231) の助成を受けたものである。

## 参考文献

- [1] B. Wilburn, N. Joshi, V. Vaish, E. Talvala, E. Antunez, A. Barth, A. Adams, M. Horowitz and M. Levoy, "High performance imaging using large camera arrays," ACM Trans. Graphics, vol. 24, no.3, pp.767-776, July 2005.
- [2] R. Ng, M. Levoy, M. Bredif, G. Duval, M. Horowitz and P. Hanrahan, "Light field photography with a hand-held plenoptic camera," Stanford University Computer Science Tech Report CSTR 2005-02, April 2005.
- [3] R. Ng, "Digital light field photography," Ph.D thesis, Stanford University, July 2006.
- [4] <http://www.lytro.com/>
- [5] E.H. Adelson and J.Y. Wang, "Single lens stereo with a plenoptic camera," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 14, no. 2, pp.99-106, Feb. 1992.
- [6] <https://github.com/nrpatel/lftools>
- [7] 日浦慎作, "コンピュータショナルフォトグラフィ," 電子情報通信学会誌, Vol.95, No. 9, pp.823-828, 2012 年.
- [8] 小松進一, "Wavefront coding による被写界深度の拡大," 情処研究報告, CVIM, 2010-CVIM-174(5), pp.1-8, 2010 年.