マルチメディア工学

２０１６年度後期

第5回　画像復元･生成

画像の劣化①

カメラ撮影された画像は「焦点ぼけ」や「ぶれ」による劣化が少なからず生じる

劣化画像　$g(x,y)$　は　元画像　$f(x,y)$　、フィルタ関数　$h(x,y)$　を用いて次のように表すことができる

$$g(x,y)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}f(x-\varepsilon,y-\vartheta)h(\varepsilon,\vartheta)d\varepsilon d\vartheta=f(x,y)*h(x,y)$$

記号　$*$　は　畳み込み積分

劣化画像　$g(x,y)$　のフーリエ変換　$G(u,v)$、　元画像　$f(x,y)$　のフーリエ変換　$F(u,v)$

フィルタ関数　$h(x,y)$　のフーリエ変換　$H(u,v)$　を使うと、次のように表される

$$G(u,v)=F(u,v)H(u,v)$$

画像の劣化②

次に示す二次元デルタ関数　$\delta(x,y)$　を考える

$$\delta(x,y)=\begin{cases}\infty & (x,y)=(0,0)\\ 0 & etc\end{cases}$$

$$\delta(x,y)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\delta(x,y)dxdy=1$$

この二次元デルタ関数は原点(0,0)のみに値を持つ関数であり、畳み込み積分に関して次のような性質がある

$$f(x,y)*\delta(x,y)=f(x,y)$$

つまり、フィルタ関数が二次元デルタ関数ならば、画像は劣化しない

画像の劣化③

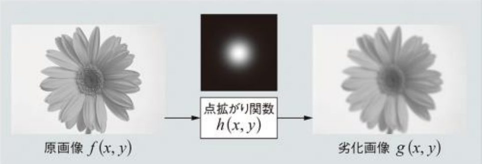
フィルタ関数が二次元デルタ関数ではなく、空間的に広がりのある関数ならば画像は劣化する

劣化を促すフィルタ関数　$h(x,y)$　は点広がり関数と呼ばれる

これにより、劣化した画像は次のように表される

$$g(x,y)=f(x,y)*h(x,y)+n(x,y)$$

$n(x,y)$はノイズを表す



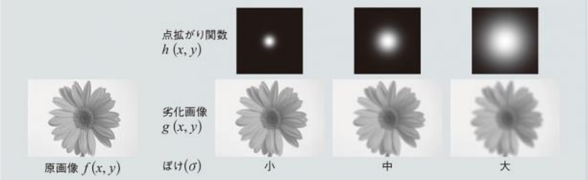
焦点ぼけの点広がり関数

焦点ぼけを近似する関数として、ローパスフィルタと同じガウス分布がある　(rは原点からの距離)

$$h(x,y)=\frac{1}{2\pi\sigma^2}\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)=\frac{1}{2\pi\sigma^2}\exp(\frac{r^2}{2\sigma^2})$$

この関数のフーリエ変換　$H(u,v)$

$$H(u,v)=\exp(-2\pi^2\sigma^2(u^2+v^2))$$



逆フィルタの画像復元

点広がり関数が既知であるとき、逆フィルタ$K_{INV}(u,v)$を劣化画像に適用することで、原画像の二次元フーリエ得変換を求めることができる

$$K_{INV}(u,v)=\frac{1}{H(u,v)}$$

$$K_{INV}G(u,v)=\frac{1}{H(u,v)}F(u,v)H(u,v)=F(u,v)$$

この結果を二次元フーリエ逆変換することで、原画像の復元画像を得ることができる

逆フィルタには欠点がある

点広がり関数$H(u,v)$が０または０に近い値となると、逆フィルタが発散してしまい、ノイズが発生する

その結果、復元された画像に大きなノイズが表れる

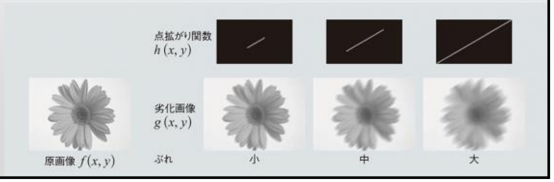
ぶれの点広がり関数

全てのぶれを近似するのは困難だが、カメラの動きが方向θ、幅wならば、次の点広がり関数で近似できる

$$h_\theta=\begin{cases}\frac{1}{w} & |x\cos\theta+y\sin\theta|\le\frac{w}{2}\ \cap\ x\sin\theta-y\cos\theta=0\\ 0 & \text{それ以外}\end{cases}$$

この関数のフーリエ変換$H_\theta(u,v)$

$$H_\theta=\frac{\sin(\pi w(u\cos\theta+v\sin\theta))}{\pi w(u\cos\theta+v\sin\theta)}$$

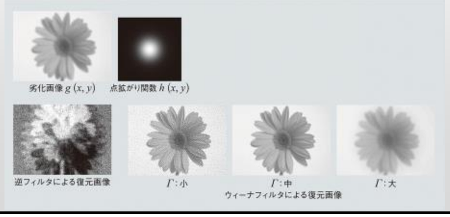


ウィーナフィルタの画像復元

復元画像と元画像の誤差を最小にする　ウィーナフィルタ　$K_W(u,v)$は次のように表される

$$K_W(u,v)=\frac{1}{H(u,v)}\frac{|H(u,v)|^2}{|H(u,v)|^2+\Gamma}$$

定数　Γ　が小さければ鮮鋭に復元されるがノイズが増え、大きければぼやけるがノイズが減る



ノイズ除去

ノイズを含んだ画像は次のように表される

$$g(x,y) = f(x,y) + n(x,y)$$

$n(x,y)$はノイズを表し、一般にノイズは確率密度分布を用いてモデル化される

ノイズ除去を行うためには、ノイズの大きさ（ガウス分布の分散）を推定しなければならない

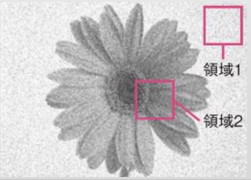
ノイズのない画像の分散　σ_f^2、　ノイズの分散　σ_n^2 とすると

$$\sigma^2 = \sigma_f^2 + \sigma_n^2$$

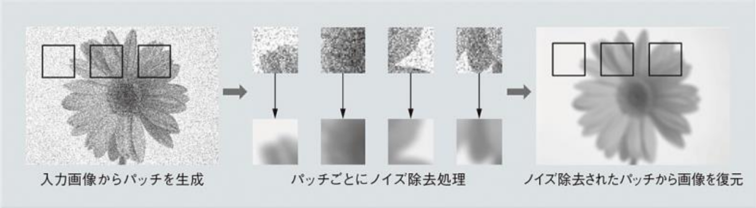
ノイズを含む画像のある領域の画素値の分散　σ^2

領域１の画素値の分散を計算することで、画像ノイズが推定できる

領域２はテクスチャを含むため、画像ノイズの推定はできない



パッチベース手法



画像全体をモデル化したり、拘束を表す汎関数を設計するのは極めて難しい

そこで、パッチと呼ばれる小領域からなるデータを抽出し、個別にノイズ除去処理をする

拘束付き最小二乗法

画像の多くの領域は滑らかであるという拘束（滑らかさ拘束）を仮定する

点広がり関数が既知であるとき、拘束付き最小二乗法のコスト関数は次のようになる

$$E = \frac{1}{\sigma^2}||g(x,y) - f(x,y)||^2 + \lambda_f \rho_f(f(x,y))$$

$$E = \sum_{k=1}^{N} \frac{1}{\sigma_k^2}||g_k - HW_k f||^2 + \lambda_f \rho_f(f)$$

λ_fは拘束の強さを表すパラメータ、　ρ_fは拘束を表す汎関数

$$\rho_f(f(x,y)) = \iint |d(x,y) * f(x,y)|^2 dxdy$$

$d(x,y)$はハイパスフィルタを表す

高周波数成分のエネルギーが大きいとコストが大きくなり、高周波成分を抑制する拘束になる

画像超解像

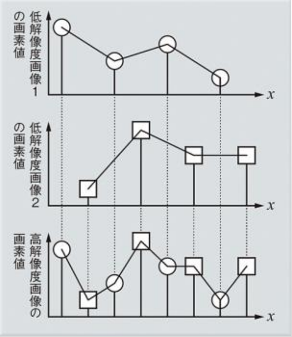
アナログ画像をデジタル画像に標本化する際に、高周波数成分が正しく表現できない場合がある

標本化によって劣化した画像から、高周波数成分を復元する処理を画像超解像という

複数フレーム超解像の方法

１．低解像度１の画像と低解像度２の画像の位置ずれ関係を推定する

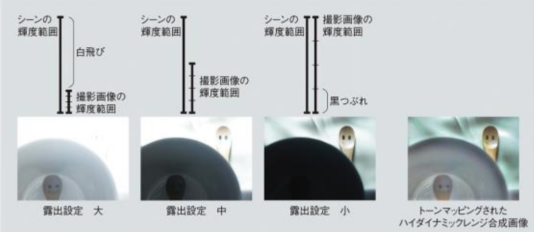
２．位置合わせを行い、高解像度画像を復元する



ハイダイナミックレンジ

ダイナミックレンジ
観測可能な最大値と最小値の比

暗い領域と明るい領域を同時に撮影するとダイナミックレンジが不足する



露出設定　大　露出設定　中　露出設定　小　トーンマッピングされた ハイダイナミックレンジ合成画像

通常、デジタル画像は８ビット（256階調）で量子化されているが、これを１０ビット以上の量子化で表現することをハイダイナミックレンジという

トーンマッピングを使用すると、暗い領域と明るい領域を合成することになる

ジョイントバイラテラルフィルタの例

入力画像　ガイド画像　バイラテラルフィルタの出力　ジョイントバイラテラルフィルタの出力

バイラテラルフィルタではノイズは除去できないが、ジョイントバイラテラルフィルタならば綺麗に除去されている

ジョイントバイラテラルフィルタ

暗い場所でフラッシュを使用しない画像は、ノイズは大きいが色を正しく表現できる

フラッシュを使用すると、ノイズが少なくなるが色を正しく表現できなくなる

フラッシュなしの画像のノイズを除去するために、フラッシュありの画像をガイド画像として利用する

これをジョイントバイラテラルフィルタという

$$g(i,j) = \frac{\sum_{n=-W}^{W}\sum_{m=-W}^{W} w(i,j,m,n) f(i+m,j+n)}{\sum_{n=-W}^{W}\sum_{m=-W}^{W} w(i,j,m,n)}$$

$$w(i,j,m,n) = \exp\left(-\frac{m^2+n^2}{2\sigma_1^2}\right)\exp\left(-\frac{(h(i,j)-f(i+m,j+n))^2}{2\sigma_2^2}\right)$$

$g(i,j)$は出力画像、$f(i,j)$は入力画像、$h(i,j)$はガイド画像、$w(i,j,m,n)$は入力画像の画素値の重み
σ_1^2, σ_2^2はそれぞれ、空間方向とガイド画像の画素値方向の重みを表すガウス分布の標準偏差

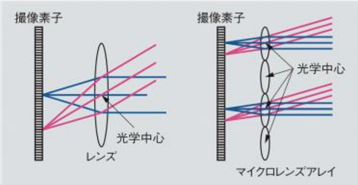
ライトフィールド撮影

カメラ撮影する時に、光線情報は一般に光源の位置情報、光線の方向、光線の波長、時刻で表される

特に、光線が一直線上にあると仮定した場合、光源の位置情報（３次元）と光線の方向（１次元）の４次元の情報が必要となり、この４次元の情報をライトフィールドという

ピンホールカメラ

ピントが１つしかない



ライトフィールド撮影

撮影後にピントをいくらでも変更できる