

画像のノイズ除去に関する基礎研究

A Basic Research on Noise Reduction from Image

(平成 19 年 9 月受理)

大槻 正伸* (OHTSUKI Masanobu) 二瓶 修一** (NIHEI Shuuichi)

Abstract

It is a difficult problem to reduce or remove various noise from image. We construct an algorithm for reducing noise from plan images (in a factory, for example) with “gravitational field model”.

And we examine the validity of this algorithm by applying it to some images which we inserted random noises artificially and we know the completely repaired ones. Consequently it is guessed that this algorithm is valid for repairing lines, curves, some figures in plan images.

1. はじめに

企業の設計現場などでは、古い図面を参考にする必要が出てくることもある。必要となる古い図面が CAD で描いてあれば（メディアの保管さえしっかりしていれば）情報の劣化はないが、紙の図面しか残っていない場合、図面そのものが劣化している可能性が大きい。

経年変化により紙そのものにしみやしわというノイズが入り込み人間でも判読しにくくなる場合も少なくない（図 1）。

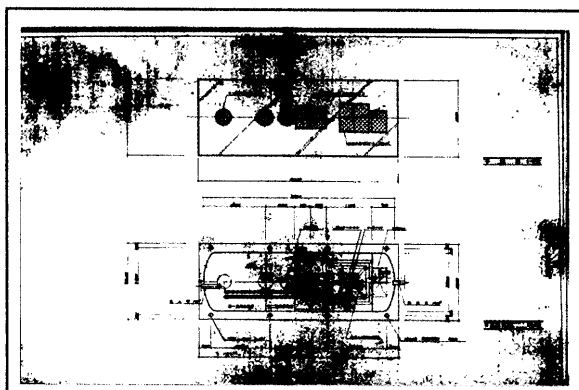


図 1 ノイズの入った図面の例

そこで、図面の修復が必要となってくる。

画像のノイズ除去プログラムは様々考案されているが、それらはどのような種類の画像にも通用することを目的としていることが多く、設計現場の図面に特化した効率のよいノイズ除去アルゴリズムに関する研究はほとんどなされていない。図面は人間が直

線、数字、各種部品等が十分判別できる程度まで修復されればよいのであるが、アルゴリズムの設計はもちろん、アルゴリズムの評価法も確立されていないのが現状である。

さて、図面のノイズ除去、修復を、人間が経験と勘を用いて行くと手間はかかるが意外に簡単であることも多い（図 2）。

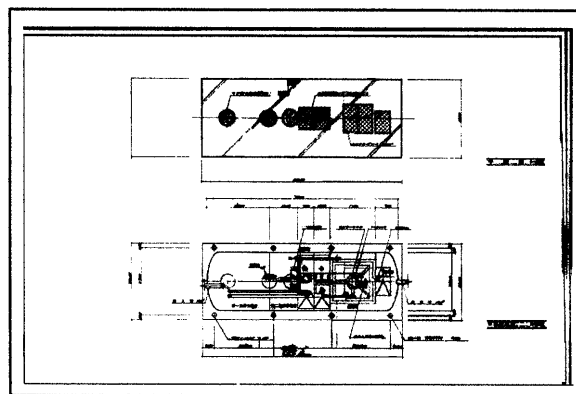


図 2 人間がノイズを取り除いた図面の例

人間は、今までの経験から「この直線は意味がありそうだ。」とか「ここは、もとは円だったものが一部何らかの原因でノイズが入り欠けたに違いない」とかが明らかに分かるパターン認識力や知識を持っているからである。

しかしながら、人手で図面の修復を行っては相当の時間がかかり効率が悪い。

そこで古い図面をスキャナで取り込み、デジタルデータ化し、その上で、自動的にノイズ除去をす

* 福島工業高等専門学校 電気工学科 (いわき市平上荒川字長尾 30)

** 福島工業高等専門学校 専攻科 機械・電気システム工学専攻 (いわき市平上荒川字長尾 30)

る効率のよいアルゴリズムの開発が望まれるのであるが、人間の経験と勘をいかにコンピュータに埋め込むかが問題になり、このようなアルゴリズム開発は大変難しいものとなっている。

以上のような背景をもとに、我々は図面のノイズ除去、修復のアルゴリズムに関する基礎研究を行っている。

以下、2. では、ノイズ除去アルゴリズムの評価法について考察する。ここでは様々な画像を要素とする空間に距離を導入し、2つの画像間の「距離」が計測できるようにする。そしてお手本となる画像と修復された画像との距離を計測することによるアルゴリズムの評価法を考える。

3. では、「画像を力学系の重力場とみなし、それを利用してノイズを除去するアルゴリズム」について考える。

そしてこの方法によるノイズ除去がどの程度有効であるかについて基礎的な実験を行う。

2. ノイズ除去アルゴリズムの評価法

ここでは、ノイズ除去アルゴリズムの評価法について考察する。

まず、画面、画像、画像の集合、画像間の距離について数学的に定義する。

【定義1】(画面、画像)

X, Y : 有限集合とする。

X を「画面」という。

$x \in X$ を「画素」という。

$p: X \rightarrow Y$ を「 X を画面とする画像」という。

あるいは、単に「 X 上の画像」という。

$p(x)$ を画素 x に対する「画素値」という。

□ (定義1)

画面、画像等の具体例をあげる。

$$X = \{(i, j); i=1, 2, \dots, n, j=1, 2, \dots, m\}$$

$$Y = \{0, 1, 2, \dots, 255\}$$

とすれば、 $n \times m$ 画素のグレイスケール(モノクロ)画像となる(図3)。

Y を $\{0, 1, 2, \dots, 255\}$ とし、 $p(x) = (r, g, b) \in Y$ とすれば、通常の BMP ファイル形式のカラー画像と同等となる。

本論文では、図面はグレイスケール画像と仮定する。

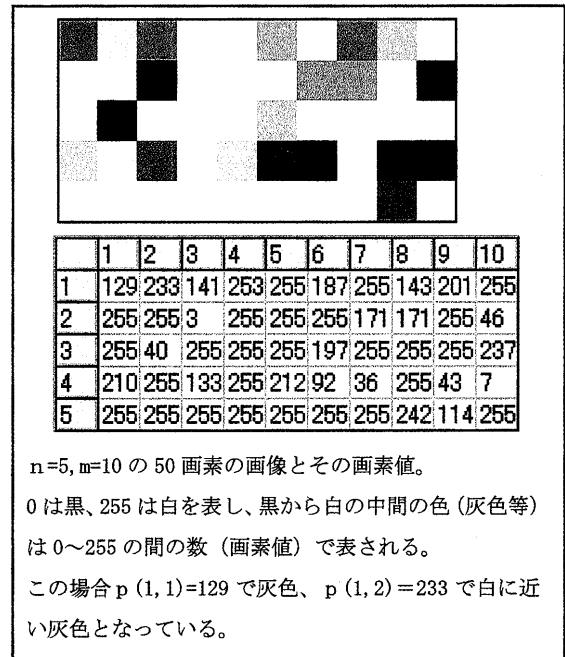


図3 グレイスケール画像の例

【定義2】(各画素間の距離)

「画素 x における距離」とは、画素 x に対応した、画素値の集合 Y に導入された距離のことであり、 d_x で表すこととする。すなわち、画素 x における距離 d_x とは3つの画像 p_1, p_2, p_3 があるとき、

$$(0) \quad d_x(p_1(x), p_2(x)) \geq 0$$

$$(1) \quad d_x(p_1(x), p_2(x)) = 0 \Leftrightarrow p_1(x) = p_2(x)$$

$$(2) \quad d_x(p_1(x), p_2(x)) = d_x(p_2(x), p_1(x))$$

$$(3) \quad d_x(p_1(x), p_2(x)) + d_x(p_2(x), p_3(x)) \geq d_x(p_1(x), p_3(x))$$

が成り立っている d_x のこととする。

□ (定義2)

【補題A】

$\phi: X \rightarrow \mathbb{R}^+$ (\mathbb{R}^+ は正の実数の集合) とする。

$I(X) = \{p; p \text{ は } X \text{ 上の画像}\}$ とする。

すなわち、 $I(X)$ は X を画面とする画像全体の集合である。

いま、各画素 $x \in X$ に対して、画素 x における距離 d_x が与えられているものとする。

$p_1, p_2 \in I(X)$ に対し、

$$d(p_1, p_2) = \sum_{x \in X} \phi(x) d_x(p_1(x), p_2(x))$$

と定義すると、 d は、 X 上の画像の集合 $I(X)$ 上の距離となる。

<証明>

$$(0) \quad d(p_1, p_2) \geq 0$$

$$(1) \quad d(p_1, p_2) = 0 \Leftrightarrow p_1 = p_2$$

$$(2) \quad d(p_1, p_2) = d(p_2, p_1)$$

は明らかである。

$$(3) \quad d(p_1, p_2) + d(p_2, p_3) =$$

$$\sum_{x \in X} \phi(x) d_x(p_1(x), p_2(x)) +$$

$$\sum_{x \in X} \phi(x) d_x(p_2(x), p_3(x))$$

$$\cong \sum_{x \in X} \phi(x) d_x(p_1(x), p_3(x)) = d(p_1, p_3)。$$

したがって、 d は集合 $I(X)$ に導入された距離となっている。□ (補題 A)

以降、ノイズのない画像を p とする。これは本来は過去の描きあがったばかりの正しい画像——我々は知ることができないものである。また、 p にノイズの入った画像を q とする。これは我々が与えられる画像である。さらにノイズを除去し、修復した画像を r とする。 r は q をもとにアルゴリズムで修復した画像である。

また、今回は距離 d としては専ら、

$$d_x(p_1, p_2) = |p_1(x) - p_2(x)|$$

$\phi_x = \frac{1}{|X|}$ ($x \in X$) という単純な距離を使うこととする。

これは、各画素 $x \in X$ で、どの程度画素値に違いがあるかの平均を意味する。

X 上の画像の集合 $I(X)$ に距離を導入することで、お手本の画像 p を基準とし、 $d(p, q)$ と $d(p, r)$ が決まるから、修復によりどの程度目標に近づいたかが計測できるようになった。

しかし、実際は、どのような距離を導入し、どの程度距離が縮まれば見やすい図面といえるのか等の議論がまた必要になる。これはこれで、認知科学の大きな問題を含んでおり、今回は詳しくは議論せず、上記のような単純な距離を導入し、どの程度距離が縮み、それは直観的に十分よく画像修復されるか、等の検討を加えることとする。

3. 重力場モデルを用いたノイズ除去アルゴリズム

ここでは、与えられたノイズのある画像を、一種の重力場とみなし、「質量」のある画素を「自由落下」させることによりノイズ除去を行うアルゴリズムについて考える。このような考え方により、画像の輪

郭抽出を行った例があるが¹⁾、図面のノイズ除去に応用した例はほとんどない。

図面のノイズ除去には、各種フィルタリング処理^{2), 3)} や、ハフ変換³⁾ 等により直線検出を行い検出された直線以外をノイズとみなし除去するアルゴリズム等が考えられる。しかし、フィルタリング処理のみではなかなかノイズがきれいに除去できず、かえって画像がにじんだりぼやけたりするか、あるいは逆に尖鋭化する場合など、尖鋭化したのはよいが本来つながべき直線が切れしまったりすることがよくおこってしまう。また直線検出のみでは、(図面には円や曲線もあり得るのは当然であるから) 不十分であることも多い。

そこで、今回は重力場モデルを用いて、図面のノイズ除去に広く応用できるノイズ除去アルゴリズムを提案し、その有効性をいくつかの試験画像で検証するのである。

重力場モデルを用いたノイズ除去アルゴリズムのイメージについて述べる。まず図4(上)のような、直線とノイズの点(いくつかの画素)があったものとする。もとの図面は中央の太い直線——実際は長方形であるが、図面の上では直線と想定されるので、直線とよぶことにするのみとする。

ここで、この図を宇宙空間に物体が配置されたものと見ると、質量のあるちいさな物体——ちりやほこり(ノイズ画素)は質量の大きな物体(本来の図面の主役である太い直線)にむかって自由落下を始める(図4(中))。

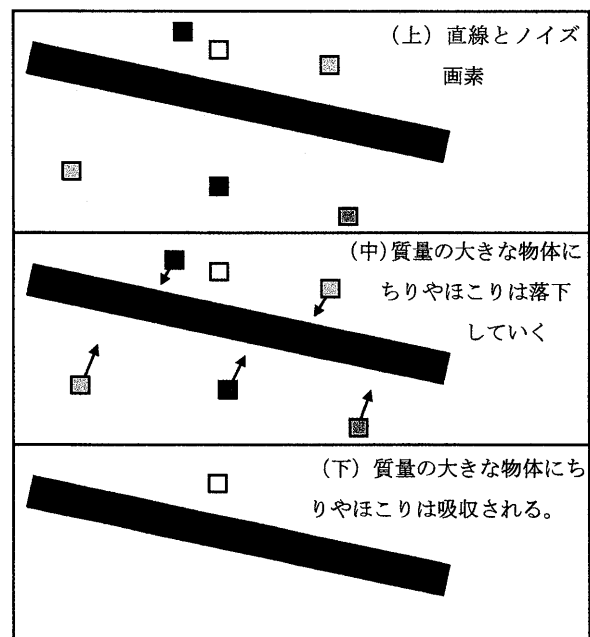


図4 ノイズ除去アルゴリズムのイメージ

そうして最後には、物体（ノイズ画素）は直線まで落下し直線と同化する。

また白い画素（今回は1つしか描いていないが、背景として多数存在しているものとする）は質量がないため落下せずにとどまる。すなわち背景の白はもとのままである。

このような動きをノイズのある図面上で再現し、実際にノイズを「自由落下」させていけばノイズ除去ができるはずである。このような考え方でノイズ除去を行うアルゴリズムを「重力場モデルを用いたノイズ除去アルゴリズム」とよぶことにする。

この方法ならば、直線に限らず有効な曲線（有効な線は十分な「質量」を持っているものと考えられる）も残すことができると予想される。また、フィルタリング処理の短所である、直線のぼやけ、尖鋭化しすぎも避けられると考えられる。

今回は正確には次のようにアルゴリズムの有効性を確認する実験を行った。

画面 X を $n \times m$ の長方形とする。

(1) 画像 p を作成する。

p は簡単な図形（直線等）を含むものとする。これが、もとの（ノイズを含まない）画像とする。通常は p は与えられないが、今回の実験では明らかになっており、ノイズ画像や、修復画像との距離を計測することができる。

(2) 画像 p に一定割合（例えば確率 0.1）で各画素を乱数でおきかえる（乱数は 0~255 の値をとる一様乱数とする）。この画像を q とする。これがノイズの入った、通常与えられると想定される画像である。

(3) 画像 q にアルゴリズムを適用しノイズ除去を行う。こうして得られた画像を r とする。

(4) 距離 d をもって、 $d(p, q)$ と $d(p, r)$ を計測する。

以上により、どの程度もとの画像に近づいたかを判断し、アルゴリズムの有効性を確認する。

以下で、上記 (3) のアルゴリズム本体について述べる。

<使用する記号等>

$$X = \{(i, j) ; i=1, 2, \dots, n, j=1, 2, \dots, m\}$$

X は $n \times m$ 画素をもつ画面である。

$q[i, j]$ を与えられた（ノイズを含む）画像を表す 2次元配列とする ($i=1, 2, \dots, n, j=1, 2, \dots, m$)。

$q[i, j]$ には、 i 行 j 列要素の画素の画素値 (0~255 の整数) が入っているものとする。

$x=(i, j)$ のとき画素値は $q[i, j]$ あるいは $q(x)$ で表され、「質量 $m(x)$ 」は $m(x) = (255 - q(x)) / 255$ で定義する。これは、黒い点、色のついた点が質量をもっていることとしたいが、黒は画素値 0 であるから上記によって質量を考えるのである。

<アルゴリズム本体>

// * 各画素 $w \in X$ に対し重力場 (V_x, V_y) 計算 *

for $i \leftarrow 1$ to n do

for $j \leftarrow 1$ to m do

begin

$w \leftarrow (i, j)$;

$$V_x[w] \leftarrow \sum_{z=(k,l) \in X} G \frac{m(z)}{|w-z|^2} \frac{(l-j)}{|w-z|}$$

$$V_y[w] \leftarrow \sum_{z=(k,l) \in X} G \frac{m(z)}{|w-z|^2} \frac{(k-i)}{|w-z|}$$

end ;

// * $|w-z|$ は点 w と点 z のユークリッド距離

// * $m(z)$ は画素 z の質量 = $(255 - q(z)) / 255$

// * 初期画像設定 $r[*,*]$ が修復画像

for $i \leftarrow 1$ to n do

for $j \leftarrow 1$ to m do

if $q[i, j] < th$ then $r[i, j] := 0$

else $r[i, j] \leftarrow 255$;

// * 今回はノイズの入った画像を 2 値化して初期画像作成

while (画像 $r[*,*]$ の状態が収束しない) do

**** 以下 1 ステップ処理 ****

begin

for $i \leftarrow 1$ to n do

for $j \leftarrow 1$ to m do

begin

$w \leftarrow (i, j)$;

if ($r[w]$ に質量がある) and

(点 w のまわりの質量が閾値以下)

then

begin

$w \leftarrow w +$

$$(V_x[w], V_y[w]) * \Delta t^2 / 2 ;$$

end ;

$c[w] \leftarrow r[w]$

// * 配列 $c[*,*]$ はワークとして使用

end ;

```
//*ワーク配列 c [*,*]を修復画面 r [*,*]にコピー
for i← 1 to n do
  for j← 1 to m do
    begin
      w←(i,j);
      r [w]←c [w]
    end
  end
end
```

<アルゴリズムの大まかな流れ>

- (1) まずは、ノイズの入った画像 $q [*,*]$ を重力場として見て、各点 w に質量のある物体をおくと、それにどの方向に力が働くのか、その力のベクトルを $(V_x[w], V_y[w])$ (ここで $w \in X$) という配列に計算する。
- (2) 修復初期画像 $r [*,*]$ を設定する。今回は $q [*,*]$ を 2 値化した画像とした。この他にも $r [w]$ を、 $q [w]$ そのものに初期化すること、ある確率 (例えば 0.5) でランダムに画素 0 (質量 1) を配置するランダム画像で初期化とすることなどが考えられる。
- (3) 修復画像 r が収束するまで、1 ステップ処理を行うが、収束判定は、今回はお手本画像が分かっているので、 $d(p,r)$ が動かなくなることをとした。実際に応用する場合にはよい収束判定方法を考案する必要がある。
- (4) 繰り返す 1 ステップ処理は、各画素に対し、「その点に質量があり、すでにどこか大きな質量のかたまりに落ちていない場合、力を受けている方向に、 Δt (秒) ぶんだけ動かす」という処理である。「大きなかたまりに落ちている」かどうかの判定は「点 w のまわり (点 w と隣接する 8 個の画素) の質量合計が定められた閾値以上である」かどうかで判定している。このあたりの判定方法、閾値のとり方等もいろいろ考え得るところである。

なお、実際の重力場では、修復画像 r を 1 ステップ動かすと重力場自体が変化する。すなわち、精密に重力場のふるまいをシミュレートするには、1 ステップごとに $(V_x[w], V_y[w])$ のベクトル場の更新を行うべきであるが、この計算量はかなり大きいので、今回は行わなかった。重力場は一貫して最初に計算して求めた $(V_x[w], V_y[w])$ を使用することとした。

実際に、サンプル画像に、人工的にノイズを入れ

込み、このアルゴリズムを適用した例をいくつか示す。

図 5 では、直線 2 本からなる十字形をもつ画像を元画像とし、上、中、下では、それぞれ確率 0.1, 0.2, 0.3 でランダムに画素値を強制的に 0~255 とし、それでノイズとした。

画像のサイズは $n \times m = 100 \times 100$ 画素とし、アルゴリズム中の主な定数は

$G=1$ 、 $\Delta t^2 / 2 = 5$ 、2 値化閾値=128 とした。

左の図がノイズを入れた画像、右の図が修復された画像であり、何ステップで収束したか (くり返し回数) も示してある。

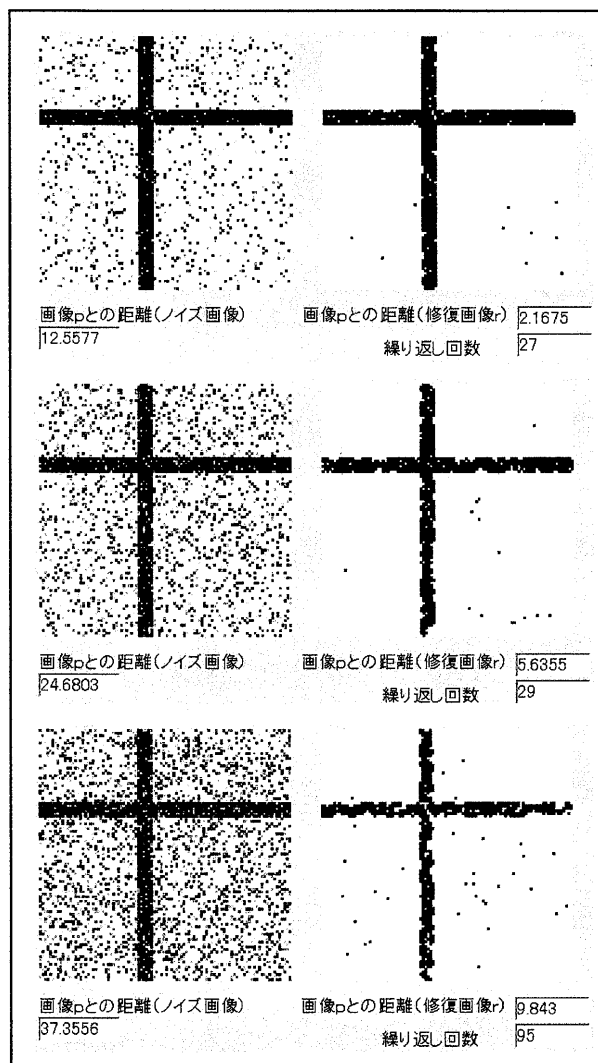


図 5 ノイズ除去実験結果例 1

(上 : ノイズ発生確率=0.1、中 : 同 0.2、下 : 同 0.3)

確率 0.1 でのノイズでは、 $d(p,q) \approx 12.6$ 、 $d(p,r) \approx 2.2$ となっており、 $\alpha = 1 - \frac{d(p,r)}{d(p,q)}$ で修復度を計測

すれば、0.835 となり、この距離修復度基準による判断では、8 割がた修復されたと見ることができる。

しかし、当然であるが、ノイズの量が多くなると、修復画像ももとの正しい直線からははずれてくる。

図5の元画像にノイズ発生確率Prを変えたときの修復の評価データを示す(表1)。これらはすべて、10回の平均値をとったものである(表で「ステップ数」とあるのは、収束するまで何ステップかかったかの平均である)。概ね元画像との距離が、ノイズの入った画像に対し、1/3~1/4程度に減少する程度の修復が実現できていることがわかる。

表1

Pr	0.1	0.2	0.3	0.4	0.5
D(p,q)	12.85	25.64	38.21	51.10	64.15
D(p,r)	2.69	5.81	10.01	14.95	20.30
ステップ数	29.6	29.7	41.4	38.7	34.3
修復度	0.79	0.77	0.73	0.71	0.68

さらにいくつか、本アルゴリズム適用例を図6に示す。図6ではノイズ発生確率はすべて0.1としてある。この程度の単純な図形でノイズもこの程度であれば人間の目にも十分修復されていると思われる。

4. 結言および今後の課題

重力場モデルを用いたノイズ除去アルゴリズムを提案し、実際にアルゴリズムを設計した。

そして、簡単な図面のモデルとなる図形(直線等)に対しどの程度ノイズ除去ができるのかについて実験した。

ノイズ除去、修復度合いの計測は、画像全体の集合に距離を導入することにより行った。

修復度を $\alpha = 1 - \frac{d(p,r)}{d(p,q)}$ で計測すると 0.1~

0.2%のランダムなノイズに対しては、おおよそ8割近くの修復度であった。修復度は、もちろん、元画像に存在する図形によっても大きく変化する。

今後の課題としては次のことがあげられる。

- (1) 画像間の距離の研究——今回は単純な距離を導入したが、どのような距離を用いるのがよいのか、また、どの程度の距離まで近づけば人間にとって「十分修復された見やすい図面」といえるのか、の認知科学的考察。
- (2) このアルゴリズムの実際の図面への応用——今回は $n \times m = 100 \times 100$ 画素の図面に対して実験を行ったが、実際の図面は 800×1000 画素程度、あるいはそれ以上のものもある。

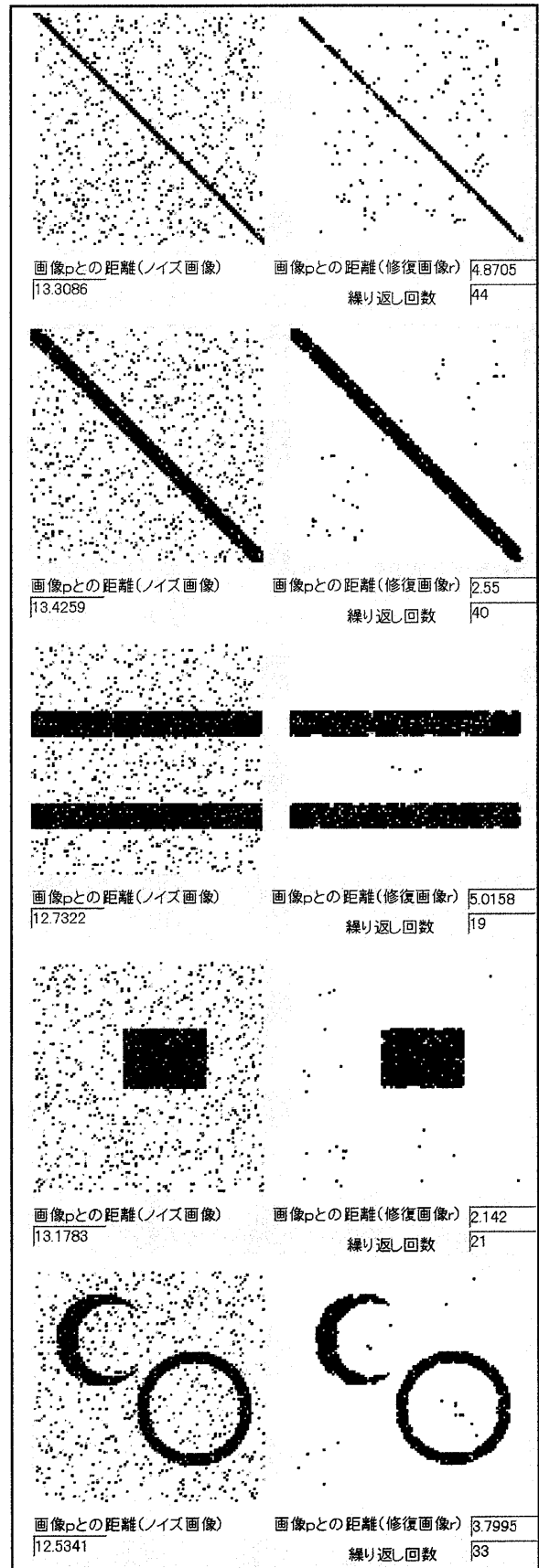


図6 ノイズ除去実験結果例2

このままこのアルゴリズムを適用したのでは、計算量が膨大になるため、図面を分割してそれぞれに適用する等の工夫が必要になると思われる。それらを含めての実際の画図面への応用（今回提案したアルゴリズムの時間計算量は、 $N = n \times m$ としたとき、重力場計算に、 $O(N^2)$ である。k分割すれば、

$$k \cdot \left(\frac{N}{k}\right)^2 = \frac{1}{k} N^2 \text{ と、計算量が } \frac{1}{k} \text{ になる。}$$

- (3) 他の処理との組み合わせの研究——重力場モデルのアルゴリズムは、直線のみならず、曲線を含む図面等にも有効と考えられるが、図5, 6でもわかるように、直線（曲線）そのものがダメージを受けて点々と一部分白になるなどの変形をしてしまう。したがって、重力場モデルのアルゴリズムとさらに適当なフィルタリング等とを組み合わせ、もう少し修復するとさらに効果的と考えられる。他のどのような処理と組み合わせるべきか、等の研究。

謝辞

このノイズ除去アルゴリズムの課題を与えて下さり、またノイズの入ったサンプル図面を提供して下さい、元ソパックシステム（株）鈴木正明氏に感謝致します。

参考文献

- 1) 湊 淳, 小澤 哲, 2値画像の輪郭抽出へのシミュレーテッドアニーリングの応用, 電子情報通信学会論文誌 D- II Vol. J88-D- II No. 1 pp148-153, 2005年
- 2) M.Seul, L.O’Gorman, M.J.Sammon, Practical Algorithms for Image Analysis, Cambridge University Press, 2000
- 3) 田村 秀之, コンピュータ画像処理, オーム社, 2002年