

9 空間フィルタリングとモーフォロジー

9.1 空間フィルタリング

フィルタというのは、例えば、コーヒーのフィルタを考えていただければ良いと思うのですが、コーヒーの粉のように必要ではないものは通さずに、必要な成分のみを抽出するものです。空間フィルタリングというのは、空間的にこのようなフィルタリングを行うものです。画像に対して何らかの処理を行い、必要な情報のみを抽出します。では、このようなフィルタはどのようにして実現するのでしょうか？

空間フィルタリングを行う場合、しばしば、ある画素の周りの4画素、あるいは8画素といった近傍画素の画素値を利用して画素値を書き換えるという処理が行われます。4近傍の場合は、上下、左右の画素値を利用して、画素値の書き換えを行います。8近傍の場合は、斜め方向も加えた8個の画素値を利用して、画素値の書き換えを行います。

ここでは、簡単のため、グレースケール画像を対象とします。グレースケール画像では、明るさが0から255で表現されます。このグレースケール画像で、座標 (x,y) での明るさを書き換えるとします。座標 (x,y) での、変換前の明るさを $I(x,y)$ 、変換後の明るさを $I'(x,y)$ とします。また、自分自身と8近傍の画素値を使って画素値の書き換えを行うとします。この場合、x座標が $x-1$ から $x+1$ 、y座標が $y-1$ から $y+1$ の範囲の画素の明るさを利用することになります。横3画素、縦3画素の範囲なので、全体で9画素が処理の対象になります。

空間フィルタリングの簡単な例として、ある画素の左上の明るさ、上での明るさ、右上での明るさと足していき、右下までの9画素分を足し合わせて、9で割るというフィルタを考えます。これは、ある画素の左上での係数である $1/9$ 、上での係数である $1/9$ と、係数のみを取り出し、

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

のように表現することができます。

では、このようなフィルタを使うと、処理の結果、どのような画像が得られるでしょうか。

9個の画素値を足して9で割っているので、このフィルタは、対象とする画素とその周りの8画素の画素値の平均化を行います。このため、このフィルタは平均化フィルタと呼ばれます。図9-1は、平均化フィルタを3回繰り返しかけた結果です。画像全体が平滑化され、ぼかしがかかったような状態になっていることが分かります。平均化フィルタを繰り返しかけることで、よりぼかしを強くすることができます。



図9-1 ぼかし処理

画像の平滑化には、ガウス分布に基づき、中央に近い画素の重みを大きくした

	1/16	2/16	1/16	
	2/16	4/16	2/16	
	1/16	2/16	1/16	

のようなフィルタが用いられる場合もあります。

画像のノイズの中には、画像の中に白や黒の点が見れる、いわゆるゴマ塩ノイズと呼ばれるものがあります。平滑化フィルタは、このようなゴマ塩ノイズの軽減などにも利用されます。近傍の画素と平均化されるので、白い画素は多少暗くなり、黒い画素は多少明るくなるので、白黒のノイズが目立たなくなります。

空間フィルタリングの別の例を見てみましょう。3×3 のフィルタで、左側の縦3つが-1、中央の縦3つが0、右側の縦3つが1の

	-1	0	1	
	-1	0	1	
	-1	0	1	

のようなフィルタを考えます。このフィルタを用いて、どのような画像処理を行えるでしょうか。

このようなフィルタがどのように働くかを考える場合は、どういったときに大きな値になって、どういった場合に値が小さくなるかを考えると、おおまかにその働きを推定することができます。まず出力が最大値になるのはどのような場合か考えてみましょう。これは、マイナスの係数にあたる画素の値が小さく、プラスの係数にあたる画素の値が大きい場合になります。具体的には、例えば、画像の3×3の大きさの領域で、右側の縦3つが255、その他が0の場合に最大値になります。この画像には、縦に続くエッジ（輪郭）が存在すると考えられます。

次にフィルタの出力が最小になる場合を考えてみましょう。フィルタの絶対値をとるとすると、出力の最小値は0になります。では、どのような場合に出力が0になるか考えると、画像の3×3の大きさの領域で、例えば、下側の横3つが255、その他が0の場合に最小値0になります。これは、下側の横3つが白、その他が黒の画像になります。この画像には、横方向に続くエッジが存在すると考えられます。以上のことから、このフィルタの出力の絶対値をとると、縦エッジが存在する部分では大きな値、それ以外では小さな値になると考えられます。すなわち、このフィルタは、縦エッジの抽出フィルタとして働くと考えられます。

図9-2は、エッジ抽出フィルタをかけた結果になります。（この図では、フィルタをかけた結果の絶対値をとったあと、定数倍して、エッジ部分が見やすいように変換しています。）このようなエッジ抽出の結果は、例えば、輪郭線を指定することによる画像検索などで利用できます。

図9-2 エッジ抽出

横方向に続くエッジについても考える場合は、

	-1	-1	-1	
	0	0	0	
	1	1	1	

のようなフィルタを考え、2つのフィルタの絶対値の和や、2乗平均が大きい部分を輪郭部分とすることができます。

9.2 モーフォロジー

基本的な画像処理の別の例として、モーフォロジー（モルフォロジー）があります。モーフォロジーは、膨張と収縮という基本的な演算の組み合わせにより、画像の処理を行うものです。ここでは、白黒の2値画像を対象とし、白が背景色、黒が物体色とします。

膨張演算は、もとの画像に黒が含まれる場合に、その画素の周りの画素を黒にする演算です。例えば、ある画素が黒だった場合に、その周りの8画素を黒にします。これにより、物体の領域が膨張します。膨張させる際の形は、構造要素と呼ばれます。上下左右だけに膨張させるといったこともできます。膨張演算により、物体の領域を大きくすることができます。

もう1つの基本的な演算は収縮演算です。収縮演算は、もとの画像に白が含まれる場合に、その画素の周りの画素を白にする演算です。例えば、ある画素が白だった場合に、その周りの8画素を白にします。これにより、物体の領域は収縮します。収縮演算は、膨張演算の逆演算とすることができます。

では、このような処理によって、どのような画像処理を行えるでしょうか。図9-3は、輪郭抽出の例になります。もとの画像は、楕円形の本体に穴が開いているとします。また、右上に細い出っ張り、左上にノイズの点があるとします。この画像を膨張させます。膨張させた結果が、中央の図です。膨張すると、楕円形の本体は、膨張します。このため、楕円形の本体は少し大きくなっています。右上の細い出っ張り

についても、膨張し大きくなっています。左上のノイズを見てみると、こちらも膨張しています。では、穴の部分は怎么样了？穴の部分を見ると、穴が小さくなっていることが分かります。黒の領域が大きくなるということは、白い領域が小さくなるということで、穴の部分は白いことから、穴の部分は小さくなるわけです。ここで、膨張させた画像と、もとの画像の差をとります。すると、右側の図のようになります。この図を見ると、物体の輪郭部分が抽出できていることが分かります。

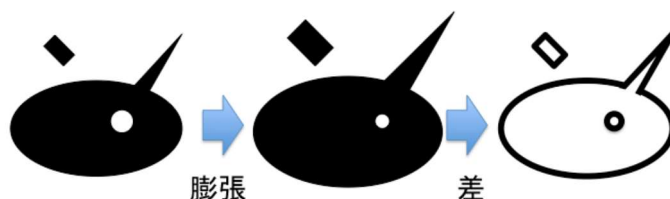


図9-3 輪郭抽出

別の例として、膨張演算と収縮演算を同じ回数、組み合わせることを考えてみましょう。例えば、膨張演算をN回繰り返す、その後、収縮演算をN回繰り返すとします。膨張演算と収縮演算は、逆の演算であるため、単純に考えると、画像はもとの状態に戻るようにも考えられます。しかし、実は、そうではありません。

膨張演算をN回繰り返す、その後、収縮演算をN回繰り返した場合の画像の変化は図9-4のようになります。もとの画像は、先ほどと同様です。膨張させると、楕円形の本体は、膨張します。右上の細い出っ張り、左上のノイズについても、膨張します。穴の部分は、黒が多くなるため、穴が小さくなります。そして、膨張演算を何度か行くと、どんどん穴の部分が小さくなり、最後は、穴が消えてしまいます。この状態から、同じ回数収縮させます。すると、楕円形の本体は、膨張した分、収縮するので、もとの大きさに戻ります。右上の細い出っ張りは、膨張した分、収縮するので、もとの大きさに戻ります。左上のノイズについても、もとの大きさに戻ります。しかし、穴の部分は、一度消えてしまったので、収縮させても復活しません。このため、膨張演算をN回繰り返す、その後、収縮演算をN回繰り返した場合、穴の部分だけが消えてしまいます。これは、穴埋めの処理になっていることが分かります。

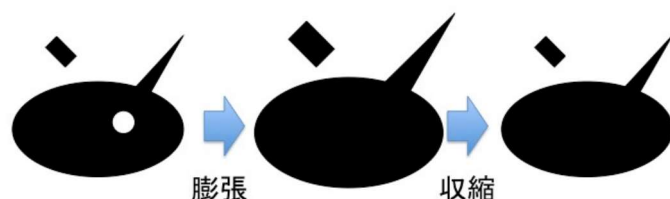


図9-4 穴埋め処理

逆の場合も考えてみましょう。収縮演算をN回繰り返す、その後、膨張演算をN回繰り返した場合の画像の変化を考えます。この場合の画像の変化は図9-5のようになります。もとの画像は、今までと同様です。収縮させると、楕円形の本体は、収縮します。穴の部分については、黒が少なくなるため、白が増える、すなわち穴は、大きくなります。右上の細い出っ張りは、収縮により細くなります。何度か収縮を繰り返すと、最後には消えてしまいます。また、左上のノイズについても、収縮により小さくなっていきます。こちらも、何度か収縮を繰り返すと、最後は消えてしまいます。この状態で、同じ回数膨張させるとします。この場合、楕円形の本体は、収縮した分、膨張するので、もとの大きさに戻ります。穴の部分は、膨張した分、収縮するので、こちらも、もとの大きさに戻ります。右上の細い出っ張りについては、一度消えてしまったので、膨張させても復活しません。左上のノイズについても、同様に、一度消えてしまったので、膨張させても復活しません。このため、収縮演算をN回繰り返す、その後、膨張演算をN回繰り返した場合、右上の細い出っ張り、左上のノイズの部分が消えてしまいます。これは、ノイズの除去を行っていることになります。このように、膨張演算と収縮演算を繰り返すことで、さまざまな画像の処理を行うことができます。

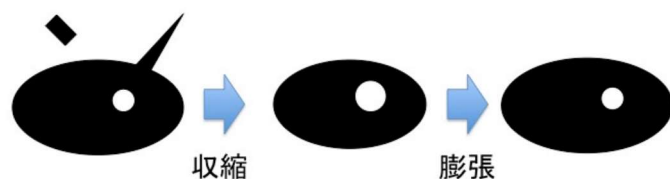


図9-5 ノイズ除去