

**平均化フィルタ**

```

PImage f, g;

void setup(){
  f = loadImage("lena.png");           // ファイルから画像を読み込む
  f.filter(GRAY);                       // 画像 f をグレースケールに変換
  g = createImage(f.width, f.height, RGB); // 画像 f と同じサイズで画像 g を用意
  size(f.width*2, f.height);           // 実行画面サイズを指定

  for(int y = 1; y < f.height-1; y++){
    for(int x = 1; x < f.width-1; x++){
      float a;
      a = red(f.get(x-1,y-1)) + red(f.get(x, y-1)) + red(f.get(x+1,y-1))
        + red(f.get(x-1,y)) + red(f.get(x, y)) + red(f.get(x+1,y))
        + red(f.get(x-1,y+1)) + red(f.get(x, y+1)) + red(f.get(x+1,y+1));
      a = a / 9;
      g.set(x, y, color(a));           // 画像 g に画素値を設定
    }
  }

  void draw(){
    image(f, 0, 0);                   // 左に画像 f (原画像) を貼る
    image(g, f.width, 0);             // 右に画像 g (変換後画像) を貼る
  }
}

```

**メディアンフィルタ**

```

PImage f, g;

void setup(){
  f = loadImage("lena.png");           // ファイルから画像を読み込む
  f.filter(GRAY);                       // 画像 f をグレースケールに変換
  g = createImage(f.width, f.height, RGB); // 画像 f と同じサイズで画像 g を用意
  size(f.width*2, f.height);           // 実行画面サイズを指定

  for(int y = 1; y < f.height-1; y++){
    for(int x = 1; x < f.width-1; x++){
      float a[] = new float[9];         // 配列 a[0]~a[8]を用意
      a[0] = red(f.get(x-1,y-1)); a[1] = red(f.get(x, y-1)); a[2] = red(f.get(x+1,y-1));
      a[3] = red(f.get(x-1,y)); a[4] = red(f.get(x, y)); a[5] = red(f.get(x+1,y));
      a[6] = red(f.get(x-1,y+1)); a[7] = red(f.get(x, y+1)); a[8] = red(f.get(x+1,y+1));

      for(int i = 0; i < 8; i++){         // 配列 a の要素を小さい順に並べ替え
        for(int j = i+1; j < 9; j++){
          if(a[i] > a[j]){
            float temp = a[i];
            a[i] = a[j];
            a[j] = temp;
          }
        }
      }
      g.set(x, y, color(a[4]));           // 画像 g に画素値を設定
    }
  }

  void draw(){
    image(f, 0, 0);                   // 左に画像 f (原画像) を貼る
    image(g, f.width, 0);             // 右に画像 g (変換後画像) を貼る
  }
}

```

## 1 次微分フィルタ

PImage f, g1, g2, g3;

```
void setup(){
    f = loadImage("lena.png");           // ファイルから画像を読み込む
    f.filter(GRAY);                       // 画像 f をグレースケールに変換
    g1 = createImage(f.width, f.height, RGB); // 画像 f と同じサイズで画像 g1 を用意
    g2 = createImage(f.width, f.height, RGB); // 画像 f と同じサイズで画像 g2 を用意
    g3 = createImage(f.width, f.height, RGB); // 画像 f と同じサイズで画像 g3 を用意

    size(f.width*2, f.height*2);          // 実行画面サイズを指定

    for(int y = 1; y < f.height-1; y++){
        for(int x = 1; x < f.width-1; x++){
            float dx, dy, norm;
            dx = red(f.get(x+1, y)) - red(f.get(x, y)); // 横方向の 1 次微分
            dy = red(f.get(x, y+1)) - red(f.get(x, y)); // 縦方向の 1 次微分
            norm = sqrt(dx*dx + dy*dy);                 // ベクトル(dx, dy)の大きさ
            dx = abs(dx);                               // 絶対値にする
            dy = abs(dy);                               // 絶対値にする
            g1.set(x, y, color(dx));                    // 画像 g1 に画素値を設定
            g2.set(x, y, color(dy));                    // 画像 g2 に画素値を設定
            g3.set(x, y, color(norm));                  // 画像 g3 に画素値を設定
        }
    }
}

void draw(){
    image(f, 0, 0); // 左上に画像 f (原画像) を貼る
    image(g1, f.width, 0); // 右上に画像 g1 (横方向 1 次微分) を貼る
    image(g2, 0, f.height); // 左下に画像 g2 (縦方向 1 次微分) を貼る
    image(g3, f.width, f.height); // 右下に画像 g3 (ベクトルの大きさ) を貼る
}
```

## ラプラシアンフィルタ

PImage f, g;

```
void setup(){
    f = loadImage("lena.png");           // ファイルから画像を読み込む
    f.filter(GRAY);                       // 画像 f をグレースケールに変換
    g = createImage(f.width, f.height, RGB); // 画像 f と同じサイズで画像 g を用意

    size(f.width*2, f.height);            // 実行画面サイズを指定

    for(int y = 1; y < f.height-1; y++){
        for(int x = 1; x < f.width-1; x++){
            float a;
            a =
                + red(f.get(x, y-1))
                + red(f.get(x-1, y)) - 4*red(f.get(x, y)) + red(f.get(x+1, y))
                + red(f.get(x, y+1));
            a = abs(a); // 絶対値にする
            g.set(x, y, color(a)); // 画像 g に画素値を設定
        }
    }
}

void draw(){
    image(f, 0, 0); // 左側に画像 f (原画像) を貼る
    image(g, f.width, 0); // 右側に画像 g (変換後画像) を貼る
}
```