# Билет 4. Методы выделения контуров. Алгоритм Кэнни

# Операторы Робертса, Прюитт, Собеля

<https://en.wikipedia.org/wiki/Roberts_cross>

In order to perform edge detection with the Roberts operator we first [convolve](https://en.wikipedia.org/wiki/Convolution) the original image, with the following two kernels:


\begin{bmatrix} 
+1 & 0 \\
 0 & -1\\
\end{bmatrix}
\quad \mbox{and} \quad 
\begin{bmatrix} 
0  & +1 \\
-1 & 0  \\
\end{bmatrix}.


Let I(x,y) be a point in the original image and G_x(x,y) be a point in an image formed by convolving with the first kernel and G_y(x,y) be a point in an image formed by convolving with the second kernel. The gradient can then be defined as:


 \nabla I(x,y) = G(x,y) = \sqrt{ G_x^2 + G_y^2 }.


The direction of the gradient can also be defined as follows:


  \Theta(x,y) = \arctan{\left(\frac{G_y(x,y)}{G_x(x,y)}\right)}.


<https://en.wikipedia.org/wiki/Prewitt_operator>

Mathematically, the operator uses two 3×3 kernels which are [convolved](https://en.wikipedia.org/wiki/Convolution) with the original image to calculate approximations of the derivatives - one for horizontal changes, and one for vertical. If we define \mathbf{A} as the source image, and \mathbf{G_x} and \mathbf{G_y} are two images which at each point contain the horizontal and vertical derivative approximations, the latter are computed as:


\mathbf{G_x} = \begin{bmatrix} 
-1 & 0 & +1 \\
-1 & 0 & +1 \\
-1 & 0 & +1 
\end{bmatrix} * \mathbf{A}
\quad \mbox{and} \quad 
\mathbf{G_y} = \begin{bmatrix} 
-1 & -1 & -1 \\
0 & 0 & 0 \\
+1 & +1 & +1
\end{bmatrix} * \mathbf{A}


where * here denotes the 2-dimensional [convolution](https://en.wikipedia.org/wiki/Convolution) operation.

Since the Prewitt kernels can be decomposed as the products of an averaging and a differentiation kernel, they compute the gradient with smoothing. Therefore, it is a [separable filter](https://en.wikipedia.org/wiki/Separable_filter). For example, \mathbf{G_x} can be written as


\begin{bmatrix} 
-1 & 0 & +1 \\
-1 & 0 & +1 \\
-1 & 0 & +1
\end{bmatrix}
=
\begin{bmatrix}
1\\
1\\
1
\end{bmatrix}
\begin{bmatrix}
-1 & 0 & 1
\end{bmatrix}
 

The *x*-coordinate is defined here as increasing in the "right"-direction, and the *y*-coordinate is defined as increasing in the "down"-direction. At each point in the image, the resulting gradient approximations can be combined to give the gradient magnitude, using:

\mathbf{G} = \sqrt{ {\mathbf{G}_x}^2 + {\mathbf{G}_y}^2 }

Using this information, we can also calculate the gradient's direction:

\mathbf{\Theta} = \operatorname{atan2}\left({ \mathbf{G}_y , \mathbf{G}_x }\right)

where, for example, **Θ** is 0 for a vertical edge which is darker on the right side.

<https://en.wikipedia.org/wiki/Sobel_operator>

The operator uses two 3×3 kernels which are [convolved](https://en.wikipedia.org/wiki/Convolution) with the original image to calculate approximations of the[derivatives](https://en.wikipedia.org/wiki/Image_Derivatives) - one for horizontal changes, and one for vertical. If we define **A** as the source image, and **G***x* and **G***y* are two images which at each point contain the horizontal and vertical derivative approximations, the computations are as follows:[[2]](https://en.wikipedia.org/wiki/Sobel_operator" \l "cite_note-2)


\mathbf{G}_y = \begin{bmatrix} 
-1 & -2 & -1 \\
 0 & 0 & 0 \\
+1 & +2 & +1 
\end{bmatrix} * \mathbf{A}
\quad
\mbox{and}
\quad   
\mathbf{G}_x = \begin{bmatrix} 
-1 & 0 & +1  \\
-2 & 0 & +2 \\
-1 & 0 & +1 
\end{bmatrix} * \mathbf{A}



where * here denotes the 2-dimensional [convolution](https://en.wikipedia.org/wiki/Convolution) operation.

Since the Sobel kernels can be decomposed as the products of an averaging and a differentiation kernel, they compute the gradient with smoothing. For example, \mathbf{G_x} can be written as


\begin{bmatrix} 
-1 & 0 & +1 \\
-2 & 0 & +2 \\
-1 & 0 & +1
\end{bmatrix}
=
\begin{bmatrix}
1\\
2\\
1
\end{bmatrix}
\begin{bmatrix}
-1 & 0 & +1
\end{bmatrix}
 

The *x*-coordinate is defined here as increasing in the "right"-direction, and the *y*-coordinate is defined as increasing in the "down"-direction. At each point in the image, the resulting gradient approximations can be combined to give the gradient magnitude, using:

\mathbf{G} = \sqrt{ {\mathbf{G}_x}^2 + {\mathbf{G}_y}^2 }

Using this information, we can also calculate the gradient's direction:

\mathbf{\Theta} = \operatorname{atan2}\left({ \mathbf{G}_y , \mathbf{G}_x }\right)

where, for example, **Θ** is 0 for a vertical edge which is lighter on the right side.

# Детектор Кэнни на маткаде

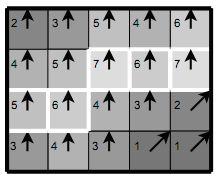
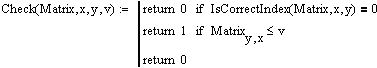
<http://habrahabr.ru/post/114589/>,

Алгоритм состоит из пяти отдельных шагов:

1. **Сглаживание**. Размытие изображения для удаления шума.
2. **Поиск градиентов**. Границы отмечаются там, где градиент изображения приобретает максимальное значение.
3. **Подавление не-максимумов**. Только локальные максимумы отмечаются как границы.
4. **Двойная пороговая фильтрация**. Потенциальные границы определяются порогами.
5. **Трассировка области неоднозначности**. Итоговые границы определяются путём подавления всех краёв, несвязанных с определенными (сильными) границами.

Перед применением детектора, преобразуем изображение в оттенки серого, чтобы уменьшить вычислительные затраты. Этот этап характерен для многих методов обработки изображений.

##### **Подавление не-максимумов**

Пикселями границ объявляются пиксели, в которых достигается локальный максимум градиента в направлении вектора градиента. Значение направления должно быть кратно 45°.  
  
Принцип подавления проиллюстрирован на рисунке выше. Почти все пиксели в примере «имеют ориентацию вверх», поэтому значение градиента в этих точках будет сравнено с ниже- и вышерасположенными пикселями. Обведённые белым контуром пиксели останутся в результирующем изображении, остальные – будут подавлены.  
Реализация проверки точки на принадлежность изображению:  
image  
Сравнение значения:  
  
Реализация подавления:  
