

13. Определение расстояния до объектов с помощью предложенной методики по снимкам реальных сцен

Чтобы найти расстояние до объекта интереса, необходимо автоматически/интерактивно выделить объект.

13.1. Определение расстояния до объектов с помощью предложенной методики по снимкам реальных сцен (автоматическое выделение объекта интереса)

Для решения этой подзадачи был задействован универсальный метод, описанный в статье [95].

Содержательная постановка задачи:

Дано трёхканальное изображение с дорожной сценой. Детектировать дорожный знак на изображении.

Математическая постановка задачи:

Дано:

$Image = \{R_{im}, G_{im}, B_{im}\}$ - кортеж,

где R_{im}, G_{im}, B_{im} - матрицы одинакового размера.

Найти:

$A[Image] \rightarrow XY$,

где $XY = \{center_1, center_2, ..., center_N\}$,

$center_i$ - координаты центров знаков.

Алгоритм:

RB -нормализация:

Изображение преобразуется из красно-зелено-голубого (RGB) в «нормализованное красно-синее» изображение Ω_{RB} . Операция осуществляется для каждого пикселя исходного изображения (Рис. 124):

$$\Omega_{RB} = \max \left(\frac{R}{R + G + B}, \frac{B}{R + G + B} \right)$$

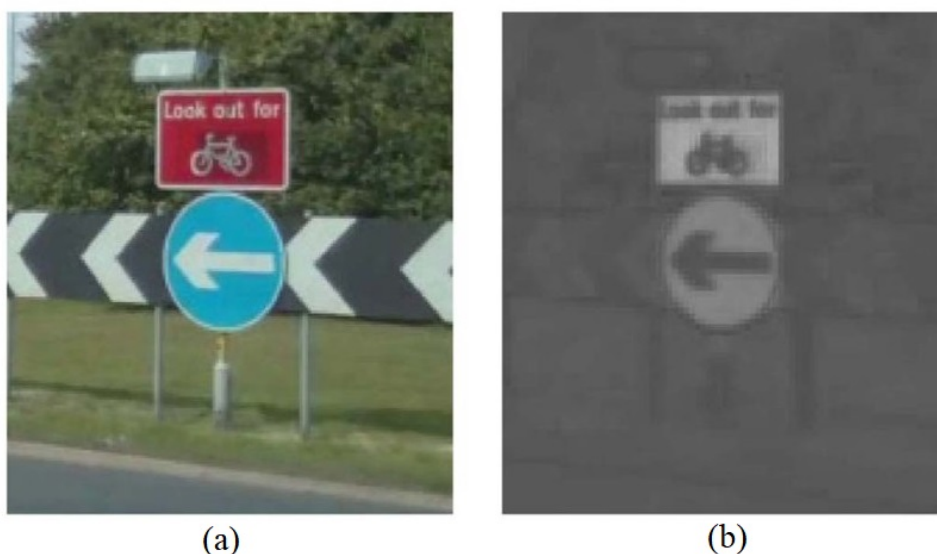


Рис. 124: Изображение преобразуется в нормализованное красное/синее цветовое пространство Ω_{RB} ((a)- исходное изображение, (b) - нормализованное изображение) [95].

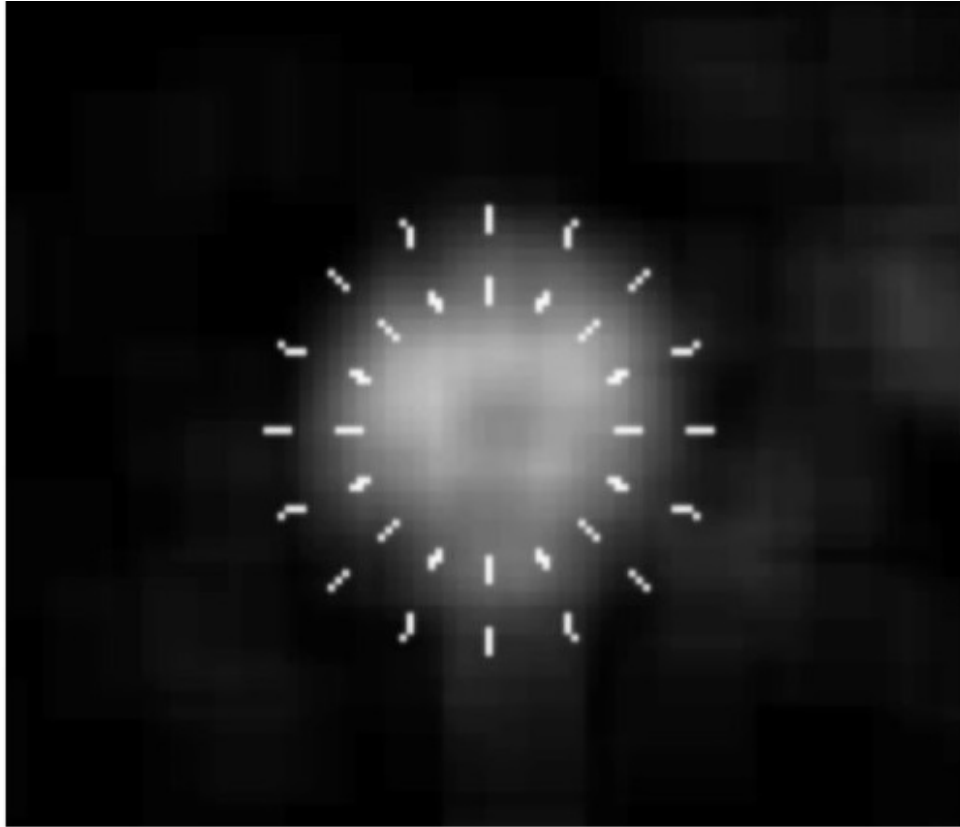


Рис. 125: Максимальный отклик при наложении [95].

Свёртка:

Свёртка позволяет получить максимальный отклик при наложении на знак (Рис. 125).

2D-свёртка:

Преобразования:

- Инверсия значения интенсивности:

$$y(n_1, n_2) = 255 - x(n_1, n_2);$$

- Смещение интенсивности с константой:

$$y(n_1, n_2) = x(n_1, n_2) + 100;$$

- Взвешивание:

$$y(n_1, n_2) = x(n_1, n_2)w(n_1, n_2);$$

- Среднее по N -neighborhood:

$$y(n_1, n_2) = \text{average}(N(x(n_1, n_2))) .$$

Два важных свойства преобразований, которые используются на изображениях:

- **линейность** ($linearity = L$);
- **инвариантность сдвига** ($shift\ invariance = SI$).

Линейность:

$$T[x_1(n_1, n_2) + x_2(n_1, n_2)] = T[x_1(n_1, n_2)] + T[x_2(n_1, n_2)]$$

$$T[\alpha x(n_1, n_2)] = \alpha T[x(n_1, n_2)]$$

Замечание: взвешивание линейно, инверсия интенсивности нелинейна
 Пространственная инвариантность:

$$T[x(n_1, n_2)] = y(n_1, n_2)$$

$$T[x(n_1 - k_1, n_2 - k_2)] = y(n_1 - k_1, n_2 - k_2)$$

Замечание: взвешивание не является *SI*, инверсия интенсивности - *SI*.

Замечание: усреднение по окрестности - *LSI*.

2D Unit Impulse - функция (*Delta*-функция) на \mathbb{Z} определяется следующим образом (Рис. 126):

$$\delta(n_1, n_2) = \begin{cases} 1 & \text{when } n_1 = 0 \text{ and } n_2 = 0 \\ 0 & \text{otherwise} \end{cases}$$

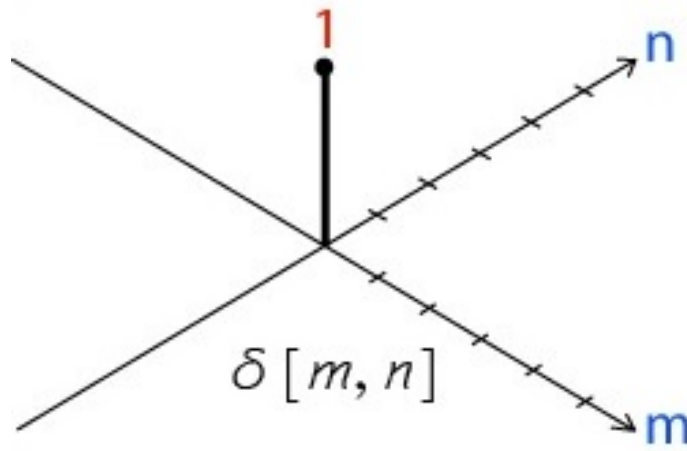


Рис. 126: *2D Unit Impulse* - функция (*Delta*-функция) на \mathbb{Z} .

Для любой двумерной функции $x(n_1, n_2)$:

$$x(n_1, n_2) = \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} \delta(n_1 - k_1, n_2 - k_2) x(k_1, k_2)$$

Импульсный отклик (Рис. 137) является результатом преобразования *LSI*, если вход - это *Delta*-функция:

$$\delta(n_1, n_2) \rightarrow T \rightarrow h(n_1, n_2)$$

Если T - это система *LSI*:

$$T[x(n_1, n_2)] = y(n_1, n_2)$$

Тогда свертка определяется следующим образом:

$$y(n_1, n_2) = x(n_1, n_2) * h(n_1, n_2) = h(n_1, n_2) * x(n_1, n_2) = \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} x(k_1, k_2) h(n_1 - k_1, n_2 - k_2)$$

Вывод формулы свёртки:

$$y(n_1, n_2) = T[x(n_1, n_2)] = T\left[\sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} x(k_1, k_2) \delta(n_1 - k_1, n_2 - k_2)\right] =$$

$$= \underbrace{\sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} x(k_1, k_2) T[\delta(n_1 - k_1, n_2 - k_2)]}_{\text{Linearity}} = \underbrace{\sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} x(k_1, k_2) h(n_1 - k_1, n_2 - k_2)}_{\text{Spatial Invariance}}.$$

Свойства свёртки:

Коммутативность: $f * g = g * f$;

Ассоциативность: $f * (g * h) = (f * g) * h$;

Дистрибутивность: $f * (g + h) = f * g + f * h$;

Ассоциативность со скалярным умножением: $\alpha(f * g) = (\alpha f) * g$.

2D-свёртка на практике (Рис. 127):

На практике и ядро, и изображение имеют конечный размер.

Пусть h и \hat{h} с размерами ядер $(2r_1 + 1) \times (2r_2 + 1)$, где \hat{h} повёрнутая версия h на 180° :

$$h = \begin{bmatrix} a_{-r_1, -r_2} & \dots & a_{-r_1, r_2} \\ \vdots & \ddots & \vdots \\ a_{r_1, -r_2} & \dots & a_{r_1, r_2} \end{bmatrix}$$

и

$$\hat{h} = \begin{bmatrix} a_{r_1, r_2} & \dots & a_{r_1, -r_2} \\ \vdots & \ddots & \vdots \\ a_{-r_1, r_2} & \dots & a_{-r_1, -r_2} \end{bmatrix}$$

$$\begin{aligned} y(n_1, n_2) &= \sum_{k_1=-r_1}^{r_1} \sum_{k_2=-r_2}^{r_2} x(k_1, k_2) h(n_1 - k_1, n_2 - k_2) = \sum_{k_1=-r_1}^{r_1} \sum_{k_2=-r_2}^{r_2} h(k_1, k_2) x(n_1 - k_1, n_2 - k_2) = \\ &= \sum_{k_1=-r_1}^{r_1} \sum_{k_2=-r_2}^{r_2} \hat{h}(k_1, k_2) x(n_1 + k_1, n_2 + k_2) \end{aligned}$$

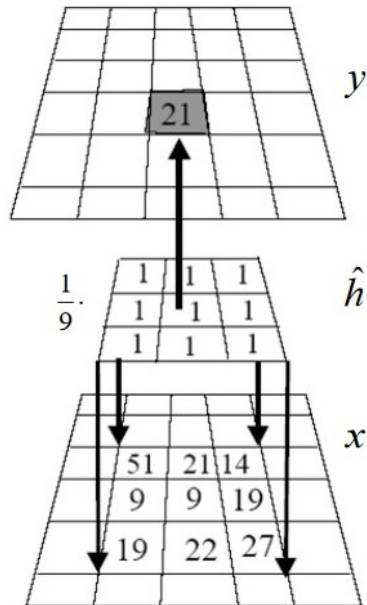


Рис. 127: 2D-свёртка на практике.

В подзадаче автоматического детектирования знака на каждом изображении видеопотока свёртка разделяется на 2 карты контрольных пикселей: внешняя/внутренняя карта. В каждой карте количество пикселей одинаково. Каждая карта даёт сумму по элементам S_1 и S_2 .

Отклик O :

$$O = ABS(S_1 - S_2).$$

При наложении свёртки на знак отклик O максимален (Рис. 128).

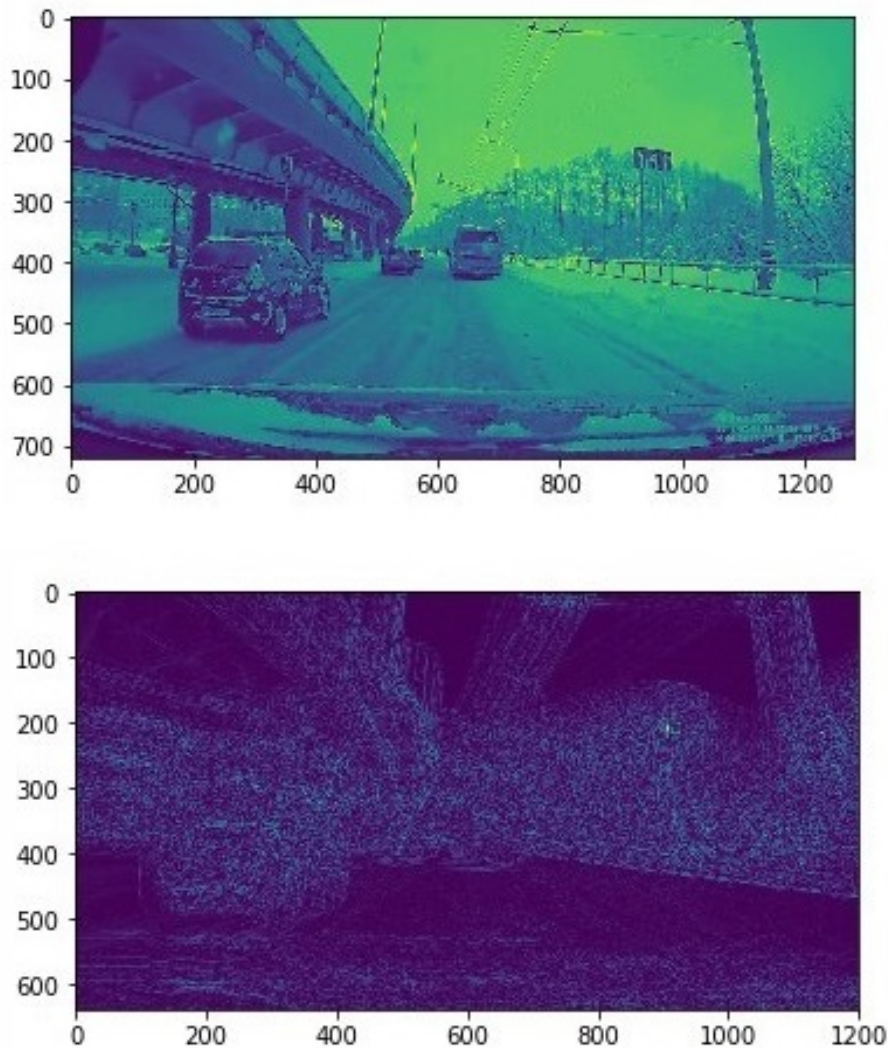


Рис. 128: Наложение свёртки на изображение, при котором отклик O максимален в области, где находится дорожный знак.

Данная свёртка выполняется намного быстрее, так как не использует ресурсоёмких операций деления и умножения, а также работает с простым целым типом данных *int16*.

Конечное детектирование знака:

Результат успешной свёртки с подобранными параметрами свёртки (радиус окружности)-яркая область, которая является знаком (Рис. 128).

Для детектирования яркой области используется свёртка с ядром лапласиана от двумерной гауссовой функции (LoG)³⁷ [116, 117].

³⁷Свёртка изображения с оператором $\nabla^2 h(r)$ эквивалентна тому, как если бы изображение сперва сворачивалось с гауссовой функцией, а затем рассчитывается оператор Лапласа.

Поскольку оператор Лапласа может обнаруживать рёбра, а также шум (изолированный, вне диапазона), изображение сглаживается свёрткой с гауссовым ядром шириной σ .

Симметричное гауссово ядро (Рис. 129, Рис. 135):

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

где σ — среднеквадратическое отклонение гауссиана, измеряемое в пикселях.

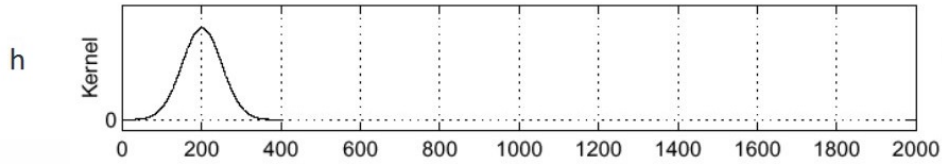


Рис. 129: *Gaussian blur kernel* [116].

Свойство ассоциативности свертки:

$$\frac{\partial}{\partial x}(h * f) = \left(\frac{\partial}{\partial x}h\right) * f$$

Т.о., нет необходимости применять 2 свёртки к сигналу f (Рис. 130) (применяется только одна с производной от оператора Гаусса (Рис. 131)) (Рис.132).

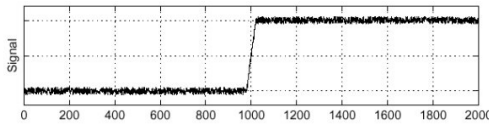


Рис. 130: Сигнал f [116].

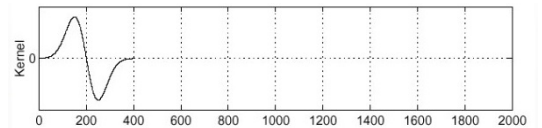


Рис. 131: Производная от гауссова ядра $\left(\frac{\partial}{\partial x}h \approx p * h\right)$ [116].

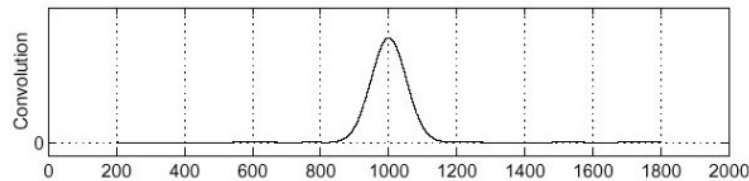


Рис. 132: $\left(\frac{\partial}{\partial x}h\right) * f$ [116].

Случай второго порядка - *Laplacian of Gaussian (LoG)*:

Smoothing + Laplace = conv. with LoG operator (Рис. 132, Рис. 133).

f - сигнал (Рис. 130);

h - *Gaussian* ядро размытия (Рис. 129);

I - *Laplace*-ядро;

$\left(\frac{\partial^2}{\partial x^2}h\right) * f$ - Рис. 134.

Так как операция свертки ассоциативна, сначала производится свёртка гауссовского сглаживающего фильтра с лапласианским фильтром. Затем гибридный фильтр - *LoG*-фильтр применяется к изображению.

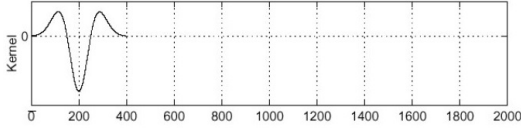


Рис. 133: $\frac{\partial^2}{\partial x^2}h \approx I * h$, где h -ядро размытия Гаусса, I -ядро Лапласа [116].

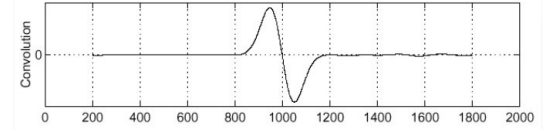


Рис. 134: $\left(\frac{\partial^2}{\partial x^2}h\right) \star f$ [116].

Обнаружение краёв с шумоподавлением:

- 1 шаг: уменьшение шума (сглаживание изображения) путем свертки с гауссовским фильтром;
- 2 шаг: обнаружение краёв сверткой с ядром Лапласа (т.е. обнаружение точек пересечения нулевого уровня для локализации границ).

Формализация [116, 117]:

Для подавления шума перед использованием *Laplace*:

$$\Delta[G_\sigma(x, y) * f(x, y)] = [\Delta G_\sigma(x, y)] * f(x, y) = LoG * f(x, y)$$

Первый знак равенства связан с тем, что

$$\frac{d}{dt}[h(t) * f(t)] = \frac{d}{dt} \int f(\tau)h(t - \tau)d\tau = \int f(\tau)\frac{d}{dt}h(t - \tau)d\tau = f(t) * \frac{d}{dt}h(t)$$

Чтобы получить $\Delta G_\sigma(x, y)$ - *Laplacian of Gaussian* (Рис. 138), необходимо рассмотреть (Рис. 136):

$$\frac{\partial}{\partial x}G_\sigma(x, y) = \frac{\partial}{\partial x}e^{-\frac{x^2+y^2}{2\sigma^2}} = -\frac{x}{\sigma^2}e^{-\frac{x^2+y^2}{2\sigma^2}},$$

а также³⁸

$$\frac{\partial^2}{\partial^2 x}G_\sigma(x, y) = \frac{x^2}{\sigma^4}e^{-\frac{(x^2+y^2)}{2\sigma^2}} - \frac{1}{\sigma^2}e^{-\frac{(x^2+y^2)}{2\sigma^2}} = \frac{x^2 - \sigma^2}{\sigma^4}e^{-\frac{(x^2+y^2)}{2\sigma^2}},$$

Аналогичным образом:

$$\frac{\partial^2}{\partial^2 y}G_\sigma(x, y) = \frac{y^2 - \sigma^2}{\sigma^4}e^{-\frac{(x^2+y^2)}{2\sigma^2}}.$$

Т.о., *LoG* определяется следующим образом:

$$LoG \triangleq \Delta G_\sigma(x, y) = \frac{\partial^2}{\partial^2 x}G_\sigma(x, y) + \frac{\partial^2}{\partial^2 y}G_\sigma(x, y) = \frac{x^2 + y^2 - 2\sigma^2}{\sigma^4}e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$

Laplacian of Gaussian (определение коэффициентов ядра с дискретной аппроксимацией двумерной функции) (Рис. 138).

Чтобы определить расстояние до некоторого объекта (например, дорожного знака, который детектирован при помощи свёртки гауссовского сглаживающего фильтра с лапласианским фильтром с подобранными параметрами [117, 116] (Рис. 140, Рис. 141)), требуется хотя бы 2 снимка (на этих фотографиях объекты интереса имеют разные размеры), полученных с разных точек, которые лежат на линии, направленной на объект (например, Рис. 138, Рис. 139).

³⁸Для простоты нормировочный коэффициент $\frac{1}{\sqrt{2\pi}\sigma^2}$ опущен.

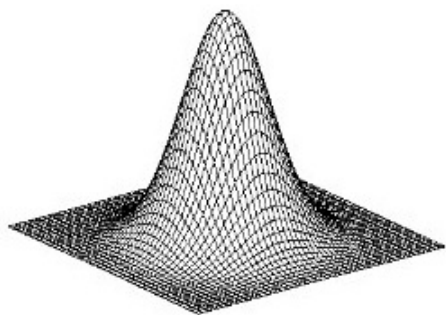


Рис. 135: 2D-ядро размытия Гаусса [116].

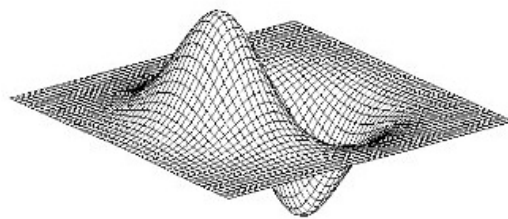


Рис. 136: Производная 2D-гауссиана [116].

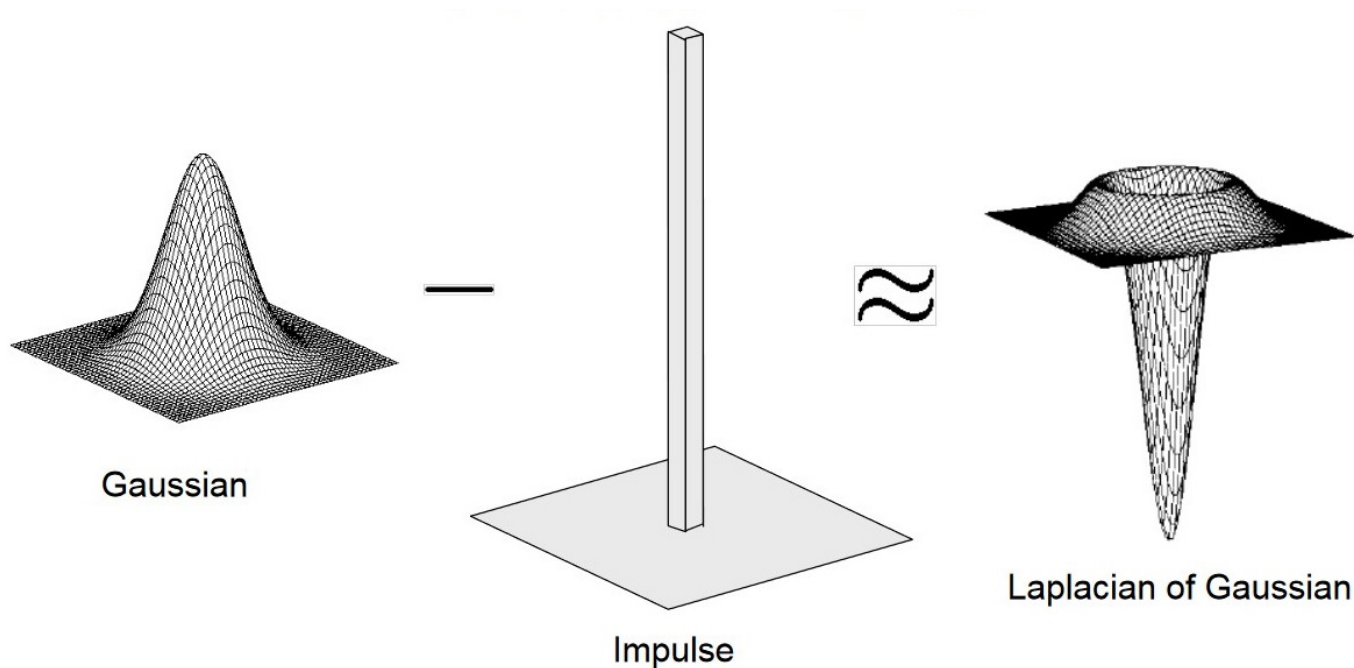


Рис. 137: $(I * h_{\sigma_1}) - (I * h_{\sigma_2}) \approx \nabla^2 I (I * h_{\sigma_3})$ [116].

В связи с тем, что скорость транспортного средства известна (зафиксирована на каждом снимке видеопотока), без труда вычисляется расстояние между кадрами сцены, сделанными Mio MiVue 765. Оно равно (т.е. расстояние между Рис. 125 и Рис. 126) 1274 см.

Т.о., при автоматическом выделении объекта интереса (Рис. 128):

- вычисленное расстояние (снимок, который сделан дальше) до объекта интереса - знака= 2743 см;
- вычисленное расстояние (снимок, который сделан ближе) до объекта интереса - знака= 1469 см;
- вычисленный параметр (длина, вычисленная по снимку, который сделан ближе) объекта в пространстве= 152 см;
- вычисленный параметр (высота, вычисленная по снимку, который сделан ближе) объекта в пространстве= 78 см.

Вся информация о эксперименте содержится в Табл.13.

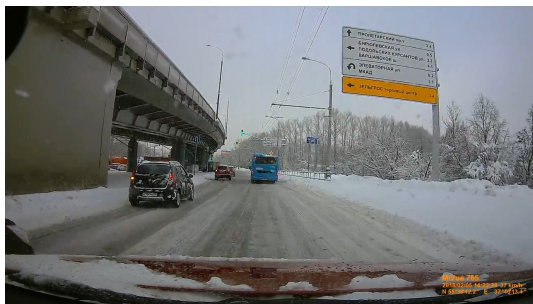
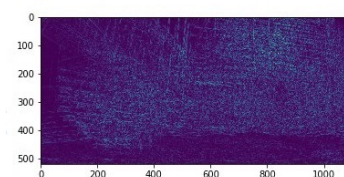


Рис. 138: Фото №1, которое сделано видеорегистратором *Mio MiVue 765*



Рис. 139: Фото №2, которое сделано видеорегистратором *Mio MiVue 765*

```
In [22]: runfile('C:/Users/Maria/Desktop/Conv.py', wdir='C:/Users/Maria/Desktop')
710
22.55153472940227
```



```
In [23]: runfile('C:/Users/Maria/Desktop/Conv.py', wdir='C:/Users/Maria/Desktop')
710
22.55153472940227
```

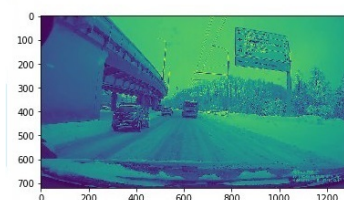
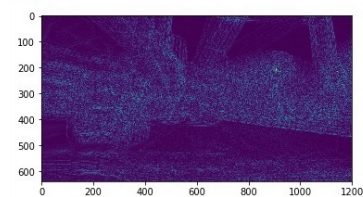


Рис. 140: Детектирование знака на Фото №1, по которому производятся вычисления расстояния до объекта неизвестного размера

```
In [5]: runfile('C:/Users/Maria/Desktop/Conv.py', wdir='C:/Users/Maria/Desktop')
904
18.717601130832815
```



```
In [6]: runfile('C:/Users/Maria/Desktop/Conv.py', wdir='C:/Users/Maria/Desktop')
904
18.717601130832815
```

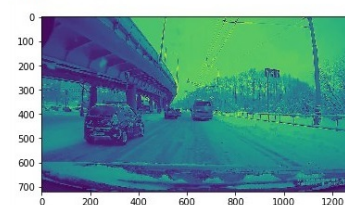


Рис. 141: Детектирование знака на Фото №2, по которому производятся вычисления расстояния до объекта

13.2. Определение расстояния до объектов с помощью предложенной методики по снимкам реальных сцен (интерактивное выделение объекта интереса на Снимке №1 и автоматическое выделение того же объекта на остальных снимках видеопотока)

В этом разделе решается подзадача интерактивного выделения объекта интереса на Снимке №1 и автоматического выделения того же объекта на остальных снимках видеопотока.

Для решения этой подзадачи задействован метод [85], применённый в разделе 8.1 для формирования обучающей выборки. Метод [85] - метод обнаружения объектов, который устойчив к изменению их масштаба и поворотам.

По факту подзадача интерактивного выделения объекта интереса на Снимке №1 и автоматического выделения того же объекта на остальных снимках видеопотока - задача регистрации изображений. Задача регистрации изображений - это определение геометрического преобразования между изображениями.

Геометрические преобразования, которые необходимо исследовать, когда изображения одной сцены сделаны с разных положений:

- поворот;
- сдвиг;
- изменение масштаба.