

УДК 004.93

Аналитическая модель обработки графических изображений в системах жизнеобеспечения города для выявления поврежденных объектов

Алексеев В.В.* , Лакомов Д.В.**

Тамбовский государственный технический университет,

ул. Советская, 106, Тамбов, 392000, Россия

**e-mail: vvalex1961@mail.ru*

***e-mail: LaDenV@yandex.ru*

Аннотация

В статье исследована возможность применения модернизированного алгоритма Канни дополненного оператором Робинсона для распознавания в автоматическом режиме беспилотным летательным аппаратом объектов изображения и обнаружении нарушения целостности строительных и жилищно-коммунальных объектов. Оператор Робинсона, с помощью свёртывания исходного изображения по четырём направлениям, позволяет точнее искать контура, по сравнению с другими операторами (Робертса, Собеля, Прюитта, Лапласиана гауссиана), использующими 2 направления свёртки. Приведены результаты исследования работы данного метода при обработке изображений различных объектов жилищно-коммунального хозяйства в ИК-диапазоне. На основе этого исследования даны рекомендации по выбору параметров работы алгоритма. Выводы, сделанные в данной статье, могут быть использованы в системах

технического зрения, а также в системах поддержки принятия решений, применяемых для выявления повреждений анализируемых объектов.

Ключевые слова: распознавание, изображение, образ, контур алгоритм, Канни, оператор, Робинсон, неопределённость.

Введение

В современных системах жизнеобеспечения города раннее выявление нарушения целостности или повреждений объектов БПЛА в автономном режиме затруднено тем, что воздействие негативных внешних и внутренних факторов вносит в этот процесс неопределённость, приводящую к размытости изображений. В связи с этим применяются алгоритмы и модели, позволяющие уменьшить влияние неопределённости при анализе изображений. Одним из таких алгоритмов является алгоритм Канни.

Анализ изображения - это процесс выделения нужной информации из изображения с помощью автоматических систем. При анализе изображения огромную роль играют контуры объектов изображения. Контуры несут в себе всю основную информацию об элементах изображения, которая слабо зависит от цвета и яркости. Контур – это пространственно-протяженный разрыв, перепад или скачкообразное изменение значений яркости [1].

Одним из самых распространённых детекторов контуров является оператор Канни, разработанный Джоном Канни в 1986 году. Канни разрабатывал свой алгоритм, основываясь на трёх критериях:

1. Точное обнаружение. Канни основывал этот критерий как повышение значения отношения сигнал/шум. Выделение максимально возможного числа истинных границ и минимального количества ложных границ.

2. Чёткая локализация. Точное определение линии границы. Минимизация фрагментации границы. Пиксели, выделенные как граничные, должны располагаться максимально близко к истинному положению границы.

3. Детектор должен реагировать на каждую границу только один раз [2].

Этапы алгоритма Канни

Выполнение алгоритма Канни начинается с предварительного этапа, на котором изображение преобразуется в градации серого [3]. Для этого достаточно представить изображение в YUV модели. В модели YUV цвет представляется в виде трёх величин: яркости- Y; U и V – две цветоразностные величины. Величина Y содержит яркостную информацию, а по величинам U и V восстанавливается исходный цвет. Переход из стандартной модели RGB (“red”, “green” “blue”) в модель YUV осуществляется по формулам:

$$Y = 0.299 * R + 0.587 * G + 0.114 * B;$$

$$U = -0.14713 * R - 0.28886 * G + 0.426 * B + 128;$$

$$V = 0.615 * R - 0.51499 * G - 0.10001 * B + 128.$$

где R, G, B - интенсивности красного, зелёного и синего цветов [4].

Первый этап алгоритма Канни – сглаживание, т.к. для получения устойчивого результата необходима фильтрация шумов. Для сглаживания применяются алгоритмы размытия изображения.

Второй этап алгоритма – поиск градиентов. Алгоритм Канни применяет четыре фильтра для определения вертикальной, горизонтальной и двух диагональных границ. Каждому пикселю ставится в соответствие значение градиента его яркости. Для этого с помощью оператора определения границ определяется вектор градиента яркости каждой точки и норма этого вектора. Угол вектора округляется и принимает одно из следующих значений: 0, 45, 90 или 135 градусов. Такое округление будет необходимо для следующего этапа [5].

Третий этап алгоритма – подавление немаксимумов. Пикселями границы являются пиксели, в которых достигается локальный максимум градиента в направлении вектора этого градиента [6].

Четвёртый этап алгоритма – двойная пороговая фильтрация. Все точки, обозначенные как локальные максимумы, проходят через два порога T_{\min} и T_{\max} , задаваемых пользователем. Если значение градиента пикселя выше верхнего порога T_{\max} , то граница в этом пикселе достоверна. Если значение градиента пикселя ниже нижнего порога T_{\min} , то пиксель подавляется. Пиксели, градиенты которых попали в промежуток между двумя порогами, принимают фиксированное среднее значение и обрабатываются на заключительном этапе [7].

Заключительный пятый этап алгоритма – трассировка области неоднозначности. На этом этапе обрабатываются пиксели, градиенты которых находятся в диапазоне между порогами. Происходит выделение групп пикселей.

Далее происходит проверка: если группа пикселей соединяется с одной из имеющихся границ, то вся группа добавляется к границе, если группа пикселей не соприкасается с какой-нибудь границей, то она подавляется. Добавление пикселя к группе осуществляется по следующему принципу: если пиксель соприкасается с группой пикселей по одному из восьми направлений, он добавляется к группе. Все пиксели, не попавшие в группы, подавляются [8].

Размытие изображения в алгоритме Канни

Одним из основных фильтров, применимых для размытия изображения, является фильтр Гаусса. Этот фильтр используется на основе ядра свёртки. Ядро свертки представляет собой матрицу, на которой определена функция Гаусса. Ядро свертки называется окном, а заданная на нем функция Гаусса – весовой или функцией окна. Каждому элементу окна соответствует число - весовой множитель. Совокупность всех весовых множителей и составляет весовую функцию. Нечетные размеры окна необходимы для однозначного определения центрального пикселя [9].

Ядро свёртки — это фильтр, который позволяет усилить или ослабить компоненты изображения. Фильтрация осуществляется перемещением окна фильтра по изображению. Весовая функция в процессе перемещения остается неизменной. В каждом положении окна происходит операция свертки – линейная комбинация значений пикселей изображения. Каждый пиксель изображения получает новое значение, основанное на группе примыкающих пикселей [10].

При каждом положении окна весовая функция поэлементно умножается на значение соответствующих пикселей исходного изображения и произведения

суммируются. Полученная сумма называется откликом фильтра и присваивается тому пикселю нового изображения, который соответствует положению центра окна [11].

Для вычисления коэффициентов весовой функции как раз используется функция Гаусса (1) для каждого коэффициента отдельно. Затем каждый коэффициент вписывается в наше ядро свёртки.

$$g(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2 + y^2}{2\sigma^2}} \quad (1)$$

где x – расстояние от центрального элемента матрицы по горизонтальной оси; y – расстояние от центрального элемента матрицы по вертикальной оси; σ – среднеквадратичное отклонение распределения гаусса [12].

Результат обработки пикселя записывается в соответствующую ячейку временной матрицы такого же размера, как и исходное изображение. Запись в отдельную временную матрицу необходима для того, чтобы исключить влияние уже обработанных пикселей на еще не обработанные.

Обязательно следует упомянуть о граничных условиях [13]. Например, у верхнего левого пикселя не существует «соседей» слева и сверху, следовательно, нам не на что умножать коэффициенты матрицы (рис.1).



Рис. 1. Левый верхний пиксель изображения

Для решения этой проблемы требуется создание промежуточного изображения. Необходимо создать временное изображение с размерами:

$$\text{width} + (\text{dim}+1) / 2 , \text{height} + (\text{dim}+1) / 2 ,$$

где width и height – ширина и высота фильтруемого изображения, dim – размерность матрицы свертки.

В центр изображения копируется входная картинка, а края заполняются крайними пикселями изображения. Размытие применяется к промежуточному буферу, а потом из него извлекается новое значение. Применение этого метода показано на рис. 2.

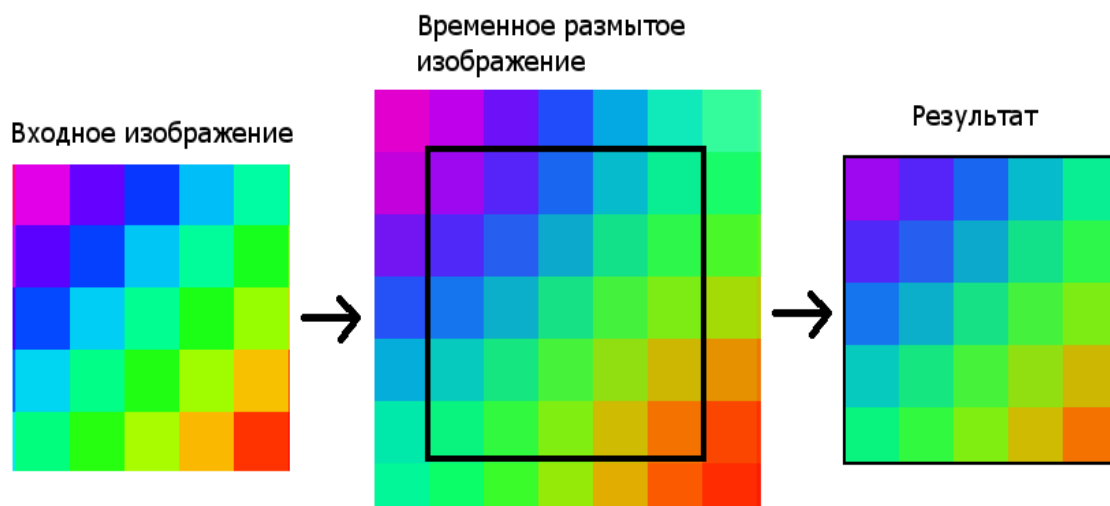


Рис. 2. Размытие с использованием промежуточного увеличенного изображения

Для того чтобы определить цвет пикселя, являющегося центром ядра, необходимо провести умножение весовых коэффициентов ядра с соответствующими значениями цвета редактируемого изображения. После этого результаты суммируются.

Полученное изображение будет размыто по сравнению с оригинальным, так как цвет каждого обработанного пикселя «распространился» среди соседних пикселей [14].

Сглаживание подавляет шум, поддерживая требование, чтобы пиксели были похожи на своих соседей. Уменьшая весовые коэффициенты для отдалённых пикселей, можно быть уверенным, что для них это требование будет не таким жёстким. Качественный анализ [11-15] приводит к таким выводам:

1) Если σ мала (например, <1), то сглаживание даст незначительный результат, поскольку весовые коэффициенты всех пикселей, находящихся не в центре, будут очень малыми.

2) При большей σ у соседних пикселей весовые коэффициенты при применении схемы взвешенного среднего будут больше, что, в свою очередь, означает, что среднее значение будет сильно стремиться к согласованию с соседями – это будет хорошая оценка значения пикселя, а за счёт размывания исчезнет большая часть шума.

3) Ядро с большой σ приведёт к тому, что вместе с шумом исчезнет и большая часть элементов изображения.

Если σ слишком мала, то ненулевым будет только один элемент матрицы. Если же велика, то размер матрицы свёртки также должен быть большим, иначе не будет учтён вклад пикселей, которые должны входить со значительными весовыми коэффициентами [15].

Поиск градиентов в алгоритме Канни

Основным элементом алгоритма Канни является математический оператор вычисления значения градиента для каждого пикселя изображения. Рассмотрим матрицу Z , размером 3×3 , содержащую значения яркости только в окрестности некоторого пикселя:

$$Z = \begin{bmatrix} z_1 & z_2 & z_3 \\ z_4 & z_5 & z_6 \\ z_7 & z_8 & z_9 \end{bmatrix}.$$

Для вычисления величины градиента в конкретной точке изображения необходимо использовать совместно обе составляющие G_x и G_y :

$$G = \sqrt{G_x^2 + G_y^2},$$

где G_x и G_y — две матрицы, где каждая точка содержит приближенные производные по x и по y ; G — значение градиента в пикселе z_5 [16].

Оператор Робертса используется для быстроты вычислений, но он проигрывает другим алгоритмам по чувствительности к шуму. Линии контуров при этом способе выделения контуров изображения получаются тоньше, чем у некоторых других методов.

Матрицы приближённых производных G_x и G_y в точке z_5 вычисляются по формулам:

$$G_x = (z_9 - z_5), \quad G_y = (z_8 - z_6).$$

Оператор использует две маски для свёртывания исходного изображения по горизонтали и по вертикали [17].

$$\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}.$$

Следующий оператор — оператор Прюитта. Данный оператор базируется на использовании масок размером 3 на 3, которые учитывают 8 возможных направлений, но прямые направления дают наилучшие результаты:

$$\begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}, \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}.$$

Матрицы приближённых производных G_x и G_y вычисляются по формулам:

$$G_x = (z_7 + z_8 + z_9) - (z_1 + z_2 + z_3), \quad G_y = (z_3 + z_6 + z_9) - (z_1 + z_4 + z_7).$$

При увеличении размера масок снижается чувствительность алгоритма к шуму. Это основное преимущество оператора Прюитта перед оператором Робертса [18].

Оператор Собеля. Данный метод обнаружения границ использует приближение к производной. Это позволяет обнаруживать край в тех местах, где градиент самый высокий. Данный способ обнаруживает количество градиентов на изображении, тем самым выделяя области с высокой пространственной частотой, которые соответствуют границам. Это приводит к нахождению абсолютной величины градиента в каждой точке входного изображения. Данный оператор состоит из двух матриц, размером 3×3 . Вторая матрица отличается от первой только тем, что повернута на 90 градусов. Матрицы используют весовой коэффициент 2 для средних элементов, что позволяет уменьшить эффект сглаживания за счет придания большего веса средним точкам и не потерять нужные детали изображения:

Данный оператор очень похож на операторы Робертса и Прюитта. Маски оператора Собеля имеют следующий вид:

$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}, \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}.$$

Матрицы приближённых производных G_x и G_y вычисляются по формулам (2):

$$G_x = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3), G_y = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7). \quad (2)$$

Используемая оператором Собеля аппроксимация градиента довольно грубая. Но он использует небольшие целочисленные фильтры, поэтому его относительно легко вычислять. В результате он достаточно часто применяется для решения многих задач [19].

Оператор Кирша. Алгоритм основан на использовании всего одной маски, которую вращают по восьми главным направлениям: север, северо-запад, запад, юго-запад, юг, юго-восток, восток и северо-восток. Маски имеют следующий вид:

$$k_0 = \begin{bmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{bmatrix}, k_1 = \begin{bmatrix} -3 & 5 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & -3 \end{bmatrix}, k_2 = \begin{bmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix}, k_3 = \begin{bmatrix} 5 & 5 & -3 \\ 5 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix},$$

$$k_4 = \begin{bmatrix} 5 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & -3 & -3 \end{bmatrix}, k_5 = \begin{bmatrix} -3 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & 5 & -3 \end{bmatrix}, k_6 = \begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & -3 \\ 5 & 5 & 5 \end{bmatrix}, k_7 = \begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & 5 \\ -3 & 5 & 5 \end{bmatrix}.$$

Величина границы определена как максимальное значение, найденное с помощью маски. Определенное маской направление выдает максимальную величину. Например, маска k_0 соответствует вертикальной границе, а маска k_5 — диагональной. Можно также заметить, что последние четыре маски являются зеркальным отражением относительно центральной оси матрицы [20].

Оператор Робинсона, подобен методу Кирша, но является более простым в реализации в силу использования коэффициентов 0, 1 и 2. Маски данного оператора симметричны относительно центральной оси, заполненной нулями. Достаточно получить результат от обработки первых четырех масок, остальные же можно получить, инвертируя первые:

$$r_0 = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, r_1 = \begin{bmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{bmatrix}, r_2 = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}, r_3 = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -2 \end{bmatrix},$$

$$r_4 = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}, r_5 = \begin{bmatrix} 0 & -1 & -2 \\ 1 & 0 & -1 \\ 2 & 1 & 0 \end{bmatrix}, r_6 = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}, r_7 = \begin{bmatrix} -2 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 2 \end{bmatrix}.$$

Максимальное значение, полученное после применения всех четырех масок к пикселю и его окружению считается величиной градиента, а угол градиента можно

аппроксимировать как угол линий нулей в маске, которые дают максимальный отклик [21].

Оператор Лапласа был предложен в 1982 году. Данный алгоритм является второй производной, определяемой по формуле (3):

$$\Delta^2 f = \frac{d^2 f}{dx^2} + \frac{d^2 f}{dy^2}. \quad (3)$$

Оператор осуществляется в два шага. На первом шаге сглаживается изображение, а затем вычисляется функция Лапласа, что приводит к образованию двойных контуров. Определение контуров сводится к нахождению нулей на пересечении двойных границ. Маски оператора Лапласа:

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}, \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}.$$

Лапласиан обычно использует нахождение пикселя на темной или светлой стороне границы [22].

В работе [23] приведены результаты применения алгоритма Канни с каждым из приведённых операторов. Они представлены на рисунке 3:

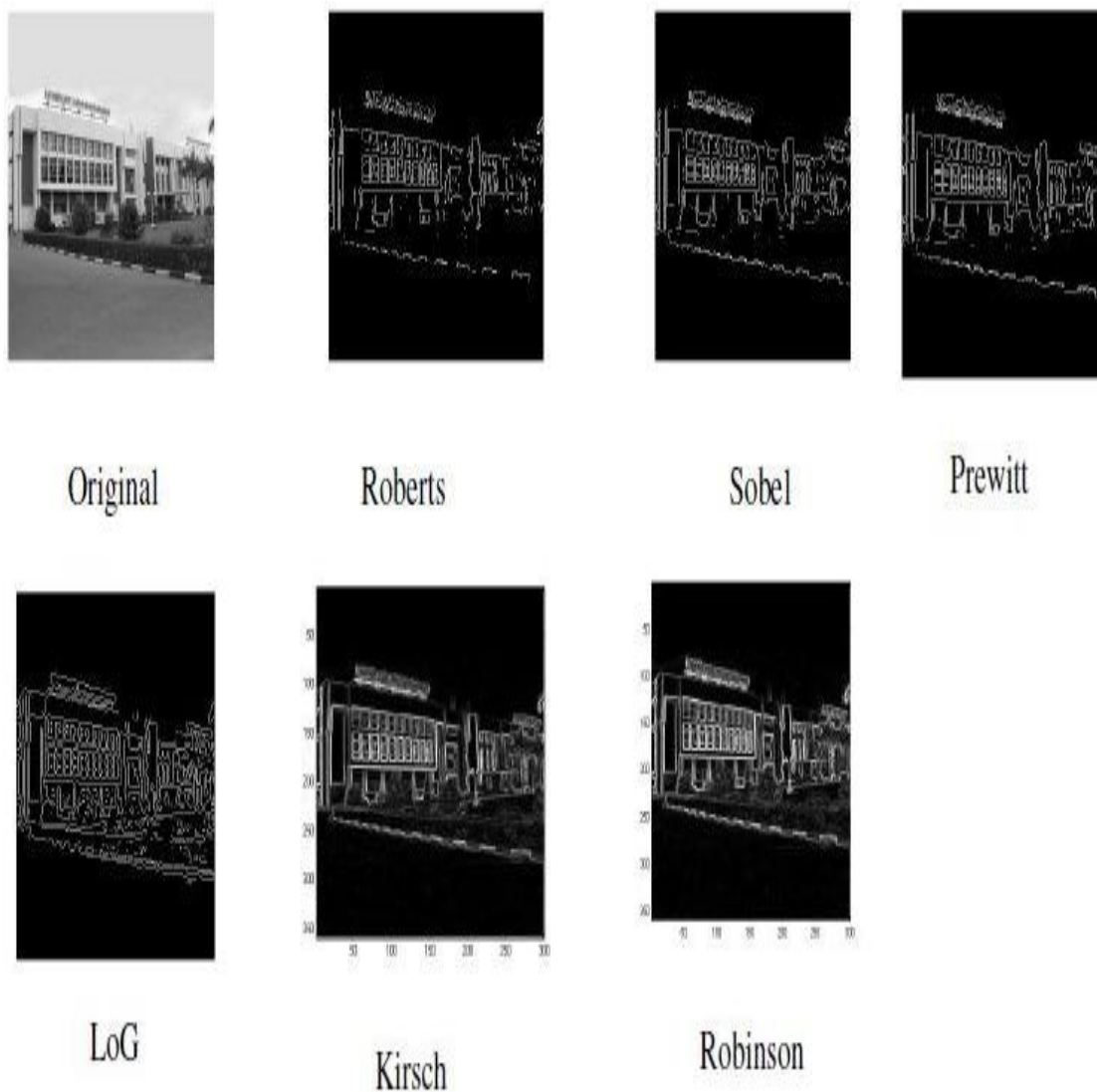


Рис 3. Оригинальное изображение и результаты работы выделения контуров с помощью различных математических операторов

Авторами был проведен дополнительный детальный анализ изображений. Результаты анализа показали, что: операторы Прюитта, Робертса и Собеля дают очень различные результаты, которые могут дополнять друг друга. Оператор Лапласа выделил больше контуров, чем операторы перечисленные выше. Операторы Робинсона и Кирша дали лучший и практически одинаковый результат

по основным контурам, но при этом появились ложные контуры на однородных объектах (стены и газон перед зданием).

Испытания алгоритма Канни

Проведём серию испытаний собственной программной реализации алгоритма Канни.

Для расчёта градиента применялся оператор Робинсона.

На рисунках 4-10 показаны обработки изображений модифицированным алгоритмом

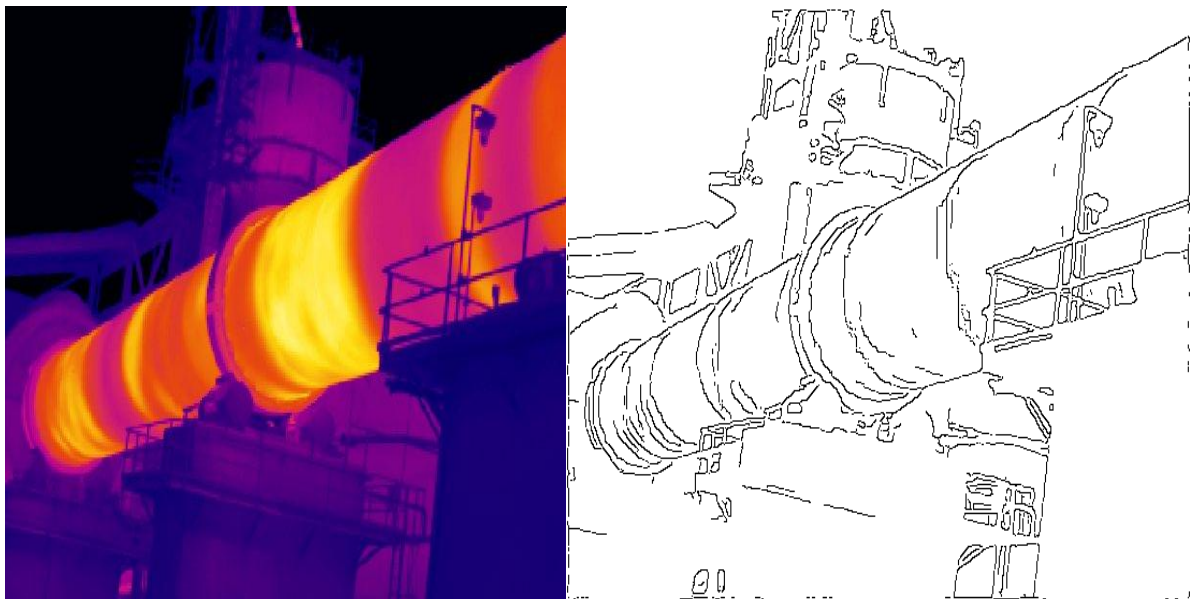


Рис.4 Обработка изображения теплотрассы в ИК диапазоне

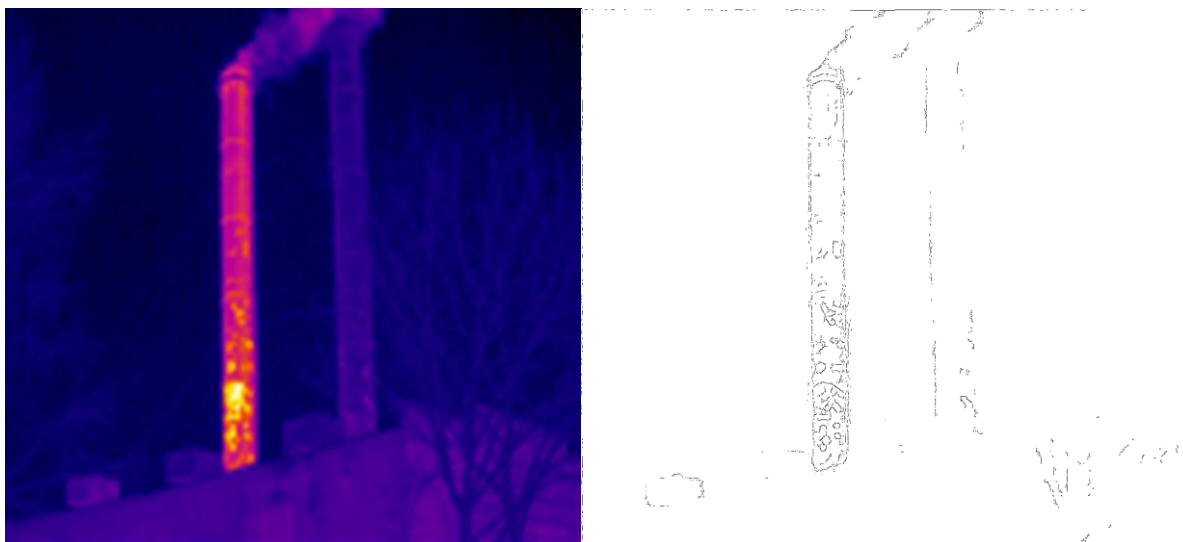


Рис.5 Обработка изображения заводской трубы в ИК диапазоне

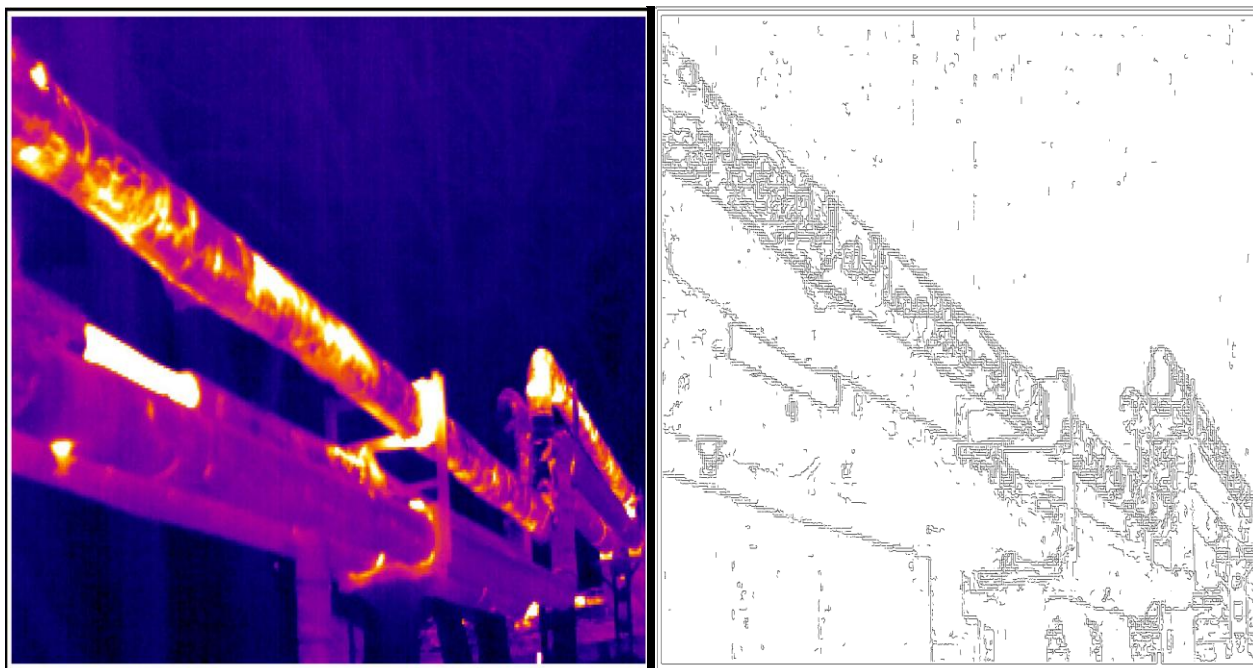


Рис.6 Обработка изображения теплопровода в ИК диапазоне

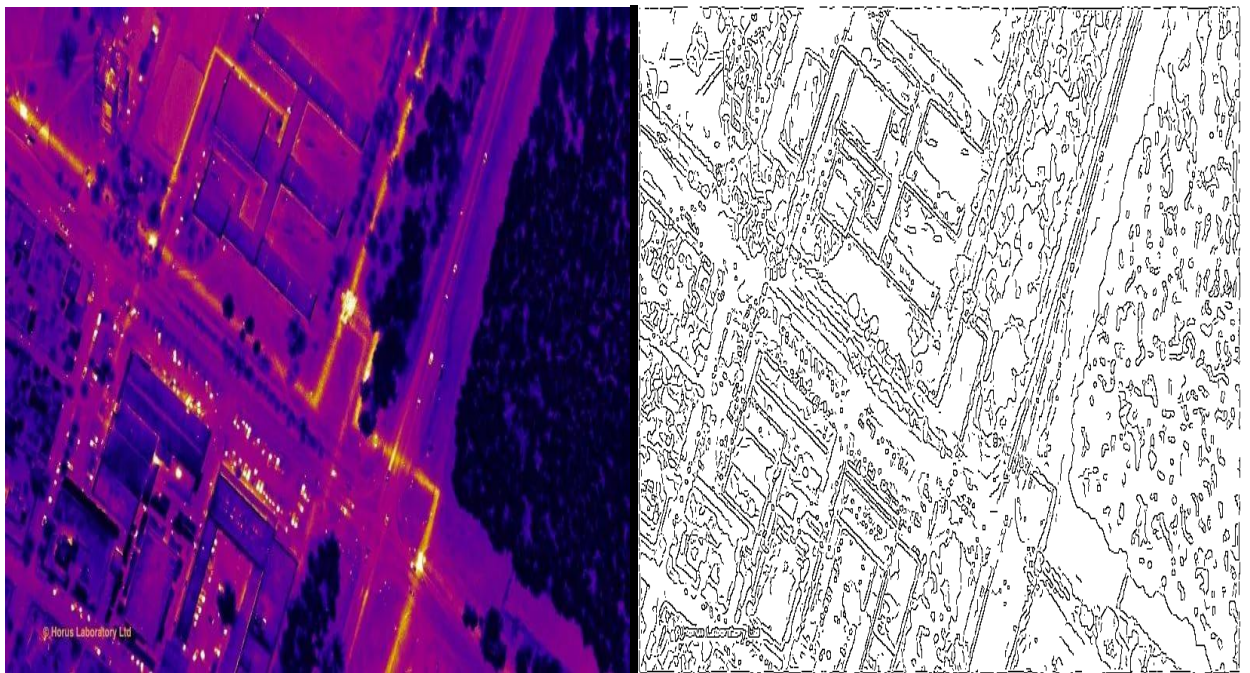


Рис.7 Обработка изображения города в ИК диапазоне

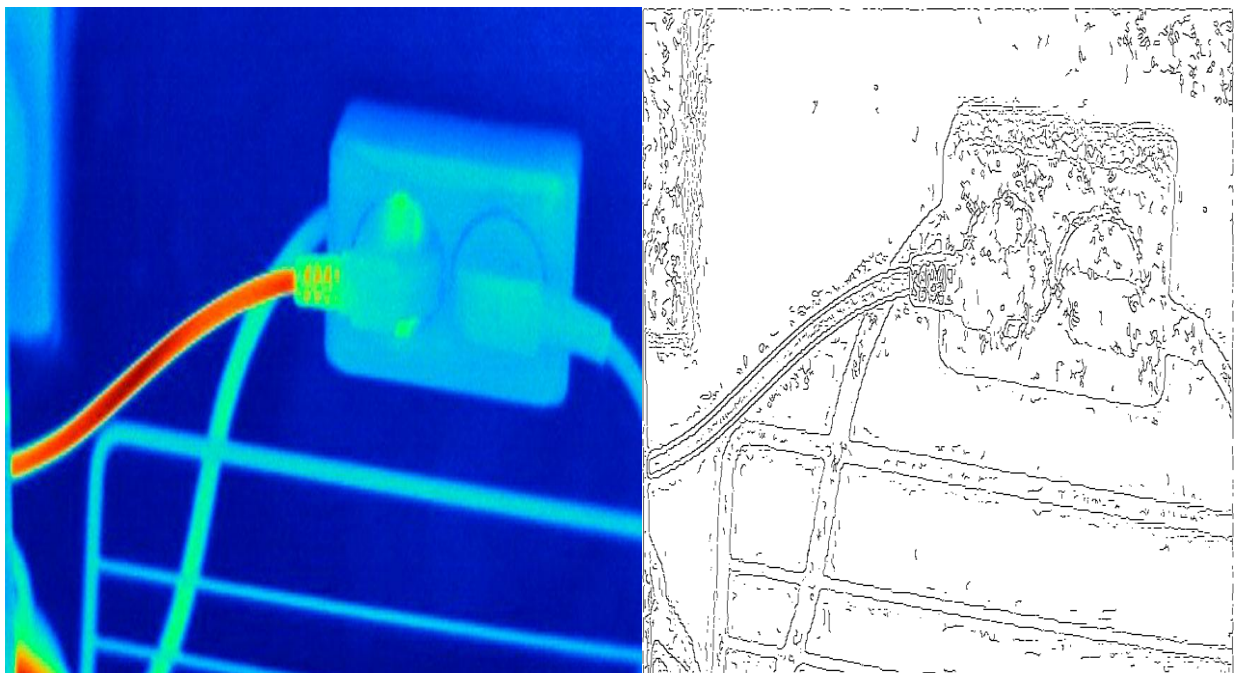


Рис.8 Обработка изображения электрокабеля и розетки в ИК диапазоне

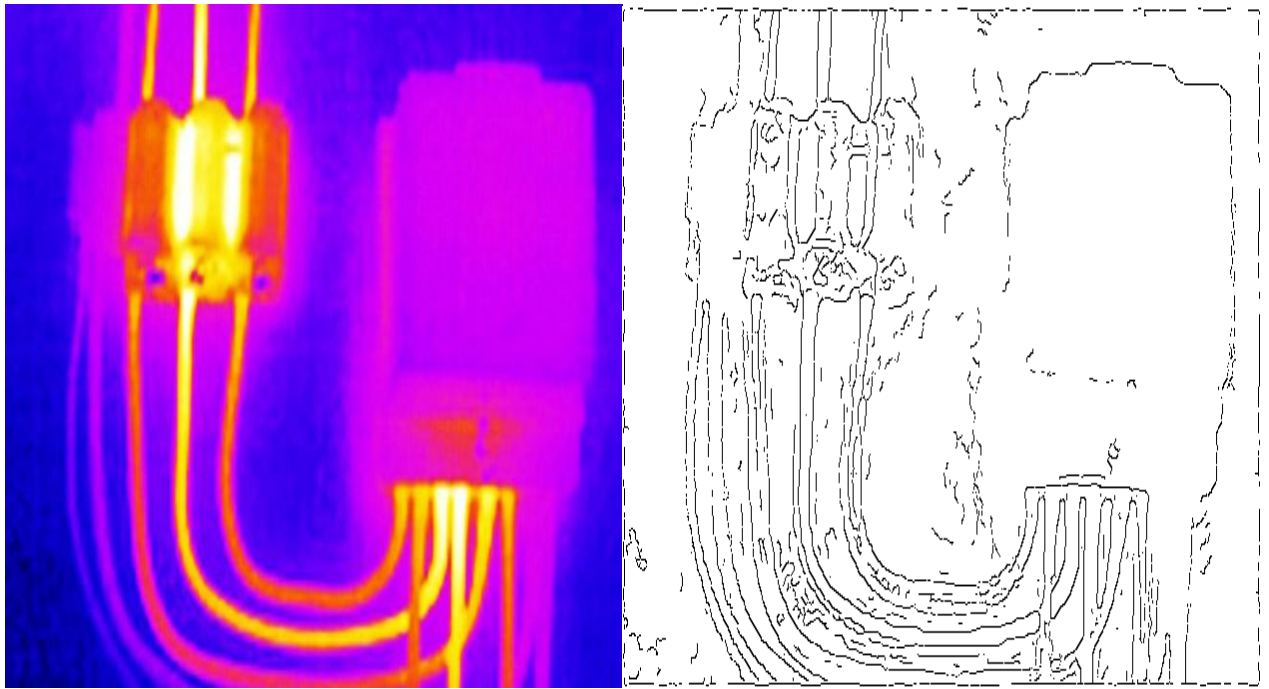


Рис.9 Обработка изображения линий связи в ИК диапазоне

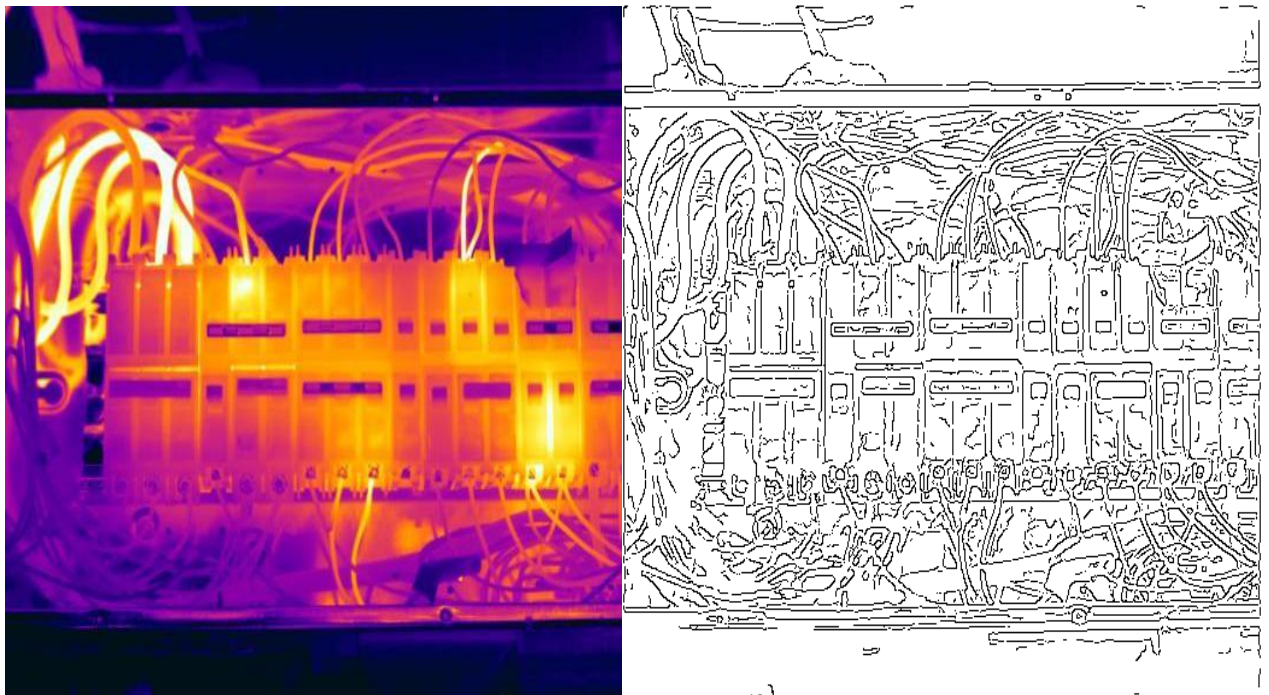


Рис.10 Обработка изображения распределительного щита в ИК диапазоне

Заключение

Достоинствами алгоритма Канни являются:

- минимизация множественного отклика на фильтр;
- увеличение значения отношения сигнал/шум;
- улучшенные показатели распознавания на зашумленных изображениях, за счёт сглаживания.

К недостаткам алгоритма можно отнести:

- отсутствие четких критериев выбора порогов, что приводит к искажению и потере части реальных контуров и к появлению ложных контуров.
- округление углов границ объектов, что приводит к повреждению или разрушению границ в точках соединения.

Результаты испытания программной реализации алгоритма Канни:

1) Программная реализация алгоритма Канни позволяет выделять технологические объекты, как на чётких, так и на размытых изображениях, а также распознаёт наличие повреждений объектов на изображениях. Результаты исследования возможно применять в системах жизнеобеспечения города.

2) Основными помехами при анализе изображений являются физические неровности фоновой поверхности, а также отражения источников света на данной поверхности.

3) Рекомендуемые параметры работы алгоритма для чётких изображений: $T_{\min} = 20$, $T_{\max} = 45$, $\sigma = 1$. Рекомендуемые параметры работы алгоритма для размытых изображений: $T_{\min} = 5$, $T_{\max} = 15$, $\sigma = 0,01$.

Библиографический список

1. Алексеев В.В., Иванова О.Г., Лакомов Д.В. Анализ применимости алгоритма Кэнни для распознавания изображений в условиях неопределённости // XVI Международная конференция «Информатика: проблемы, методология, технологии». Сборник трудов. (Воронеж, 11-12 февраля 2016) - Воронеж: Изд-во Научно-технические публикации, 2016. С. 42 - 45.
2. Местецкий Л.М. Математические методы распознавания образов. – М.: МГУ, 2004. – 144 с.
3. Дидрих В.Е., Мадроньеро П.Р. Применение контурного анализа для идентификации объектов в изображениях // XIII Международная научно-методическая конференция «Информатика: проблемы, методология, технологии». Сборник трудов. (Воронеж, 7–8 февраля 2013). – Воронеж: Воронежский государственный университет, 2013. Т. 2. С. 290 - 293.
4. Алексеев В.В., Громов Ю.Ю., Губсков Ю.А., Ищук И.Н. Методология дистанционной оценки пространственных распределений оптико-теплофизических параметров объектов, замаскированных под поверхностью грунта. - М.: Научтехлитиздат, 2014. - 248 с.
5. Карасев П.И., Губсков Ю.А. Процедура обработки графических изображений в системах видеонаблюдения // Вестник Воронежского института ФСИН России. 2015. № 2. С. 35 – 37.
6. Ким Н.В., Кузнецов А.Г., Крылов И.Г. Применение систем технического зрения на беспилотных летательных аппаратах в задачах ориентации на местности // Вестник Московского авиационного института. 2010. Т.17. № 3. С. 42 - 49.

7. Гонсалес Р., Вудс Р. Цифровая обработка изображений. – М.: Техносфера, 2005. – 1072 с.
8. Пытьев Ю.П. Методы морфологического анализа изображений. - М.: ФИЗМАТЛИТ, 2010. – 336 с.
9. Потапов А.А. Новейшие методы обработки изображений. – М.: ФИЗМАТЛИТ, 2008. – 494 с.
10. Сакович И.О., Белов Ю.С. Обзор основных методов контурного анализа для выделения контуров движущихся объектов // Инженерный журнал: Наука и инновации. 2014. № 12. С. 21 - 25.
11. Фурман Я.А. Введение в контурный анализ. Приложения к обработке изображений и сигналов. – М.: ФИЗМАТЛИТ, 2003. – 592 с.
12. Кравченко В.Ф. Цифровая обработка сигналов и изображений в радиофизических приложениях. – М.: ФИЗМАТЛИТ, 2007. – 553 с.
13. John Canny. A computational approach to edge detection // IEEE Transactions on Pattern Analysis and Machine Intelligence, 1986, vol. PAMI-8(6), pp. 679 – 698.
14. Ким Н.В., Крылов И.Г. Групповое применение беспилотного летательного аппарата в задачах наблюдения // Труды МАИ. 2012. № 62. URL: <http://trudymai.ru/published.php?ID=35507>
15. Алексеев В.В., Карасев П.И., Лакомов Д.В. Анализ методов обработки изображений, применимых в условиях неопределённости // XVI Международная конференция «Информатика: проблемы, методология, технологии». Сборник трудов. (Воронеж, 11-12 февраля 2016) - Воронеж: Изд-во Научно-технические публикации, 2016. С. 37 - 41.

16. Алексеев В.В., Лакомов Д.В. Анализ применимости размытия при распознавании изображений в условиях неопределенности // III Международная научно-практическая конференция «Виртуальное моделирование, прототипирование и промышленный дизайн». Тезисы докладов. (Тамбов, 15-17 ноября 2016). - Тамбов: ТГТУ, 2016, Т.2. С. 138 - 141.
17. Алексеев В.В., Лакомов Д.В. Оператор Робинсона и его применение в алгоритме Кэнни для распознавания изображений в условиях неопределённости // XV Всероссийская научная конференция «Нейрокомпьютеры и их применение». Тезисы докладов. (Москва, 14 марта 2017). – М: ГППУ, 2017. С. 89 - 90.
18. Грузман И.С., Киричук В.С. и др. Цифровая обработка изображений в информационных системах – Новосибирск: Изд-во НГТУ, 2002. – 352 с.
19. Ковригин А.В. Применение принципов построения систем машинного зрения в задаче анализа изображений клеточных структур // Научный журнал КубГАУ. 2007. № 29 (5). URL: <http://ej.kubagro.ru/2007/05/pdf/03.pdf>
20. Смирнов А.В., Пескин А.Е. Цифровое телевидение: от теории к практике. – М.: Горячая линия - Телеком, 2005.- 349 с.
21. Джакония В.Е., Гоголь А.А., Друзин Я.В. и др. Телевидение. – М.: Горячая линия - Телеком, 2004. - 228 с.
22. Быков Р.Е., Фрайер Р., Иванов К.В., Манцветов А.А. Цифровое преобразование изображений – М.: Горячая линия-Телеком, 2003. - 229 с.
23. Muthukrishnan R., Radha M. Edge Detection Techniques for Image Segmentation // International Journal of Computer Science & Information Technology (IJCSIT), 2012, no. 3(6), pp. 259 – 267.