

УДК 004.932

В. Н. Котенко, И. Н. Полищук
Донецкий национальный университет, г. Донецк
83001, г. Донецк, ул. Университетская, 24

МЕТОД ПОВЫШЕНИЯ КАЧЕСТВА ВЫДЕЛЕНИЯ ГРАНИЦ ИЗОБРАЖЕНИЙ С ИСПОЛЬЗОВАНИЕМ АЛГОРИТМА PARTICLE SWARM OPTIMIZATION

V. N. Kotenko, I. N. Polishchuk
Donetsk National University, c. Donetsk
83001, c. Donetsk, Universitetskaya str., 24

METHOD TO INCREASE EDGE DETECTION QUALITY USING PARTICLE SWARM OPTIMIZATION ALGORITHM

В. М. Котенко, І. М. Поліщук
Донецький національний університет, м. Донецьк
83001, м. Донецьк, вул. Університетська, 24

МЕТОД ПІДВИЩЕННЯ ЯКОСТІ ВИДІЛЕННЯ КОРДОНІВ ЗОБРАЖЕНЬ З ВИКОРИСТАННЯМ АЛГОРИТМУ PARTICLE SWARM OPTIMIZATION

На примере конкретной задачи разработан алгоритм для повышения качества выделения границ изображений. Проведено практическое исследование производительности алгоритмов для выделения границ изображений. Предложен метод использования роевого алгоритма для высокой точности локализации на сильно зашумлённых изображениях.

Ключевые слова: обработка изображений, выделение границ, гомогенность, алгоритм Кенни, роевые алгоритмы, производительность.

New algorithm was designed on the specific objective to improve edge detection performance on images. Existing edge detection algorithms performance was researched on practice. New approach of edge detection using particle swarm optimization was proposed to increase localization accuracy on heavy noised images.

Keywords: image processing, edge detection, homogeneity, Kenny algorithm, Particle swarm optimization, performance.

На прикладі конкретного завдання розроблений алгоритм для підвищення якості виділення кордонів зображень. Проведено практичне дослідження продуктивності алгоритмів для виділення меж зображень. Запропоновано метод використання ройового алгоритму для високої точності локалізації на сильно зашумлених зображеннях.

Ключові слова: обробка зображень, виділення меж, гомогенність, алгоритм Кенні, ройові алгоритми, продуктивність.

Введение

Технологический прогресс поставил перед разработчиками программного обеспечения проблему распознавания изображений. Выделение границ объектов играет важную роль в алгоритмах обработки изображений и находит применение в различных сферах, начиная от распознавания лиц и заканчивая поиском раковых опухолей [1]. Особенно сложной данная задача является в случае зашумлённости изображения.

Обозначенная проблема остаётся актуальной в сфере искусственного интеллекта: проводятся ежегодные конференции IEEE, создаются интерфейсы программирования приложений (API) для работы с изображениями в реальном времени.

В статье предложен метод выделения границ на изображениях, построенный на основе алгоритма Partical Swarm Optimisation (PSO) [1-3]. Данный алгоритм представляет собой метод численной оптимизации и изначально использовался для моделирования социального поведения. Метод роя частиц доказал свою пригодность для выполнения широкого спектра задач: минимизация функций, оптимизация, обучение систем искусственного интеллекта и др. В данной работе будет рассмотрена его адаптация к задаче выделения границ с целью улучшения качества результатов на зашумлённых изображениях.

Большинство существующих алгоритмов детектирования границ используют информацию небольшого участка изображения для определения принадлежности пикселя к границе [1]. Таким образом, теряется часть данных, которые могут помочь в определении принадлежности пикселя к кривой. Площадь покрытия в задаче определения границ имеет большое значение для точности локализации границы.

Целью данной работы является создание метода, в котором используется оценочная функция, построенная на анализе полного изображения, что позволяет минимизировать влияние шума. Эта функция базируется на построении кривых, повторяющих возможную границу, и оценки этих кривых при помощи двух критериев – гомогенности и похожести. В отличие от традиционных алгоритмов, этот метод позволит находить границы объектов на очень сильно зашумлённых изображениях. В алгоритме будет использоваться метод округления вещественных значений частиц в PSO для их большей дифференциации.

Формулировка задачи

Качество работы систем, связанных с обработкой изображений зависит как от аппаратного, так и от программного обеспечения. В контексте данной статьи будет рассматриваться программная обработка статического изображения, где уровень зашумлённости будет заранее определён (от 0 до 100%). Таким образом, на вход системы детектирования границ получаем изображение в виде двумерного массива пикселей размером $W \times H$. В пределах изображения имеется граница, которую можно представить кривой линией – она разделяет между собой объект и фон. Алгоритму необходимо определить эту границу, возможно, при наличии шумов.

Для решения задачи будут использоваться:

- принципы работы существующих алгоритмов детектирования границ;
- концептуальная модель PSO.

Необходимо:

- адаптировать алгоритм PSO к задаче выделения границ;

- создать программное обеспечение на основе этого алгоритма для проведения экспериментов;
- провести эксперименты для оценки качества полученного метода.

Понятие выделения границ изображения

В идеальном случае, результатом выделения границ является набор связанных кривых, обозначающих границы объектов, граней и оттисков на поверхности, а также кривых, которые отображают изменения положения поверхностей. Таким образом, применение фильтров выделения границ, таких как фильтр Робертса, Собеля, Трюерта или Кенни, к изображению может существенно уменьшить количество обрабатываемых данных вследствие того, что отфильтрованная часть изображения считается менее значимой для решаемой задачи, а наиболее важные структурные свойства изображения сохраняются. Однако не всегда возможно выделить границы в зашумлённых или размытых изображениях даже с использованием фильтра [1].

При выделении границ есть два основных подхода: поиск максимумов и поиск нулей [4].

Существуют четыре большие группы детекторов границ изображения:

- детекторы, использующие компоненты градиента яркости;
- детекторы, использующие коэффициент совпадения с шаблоном;
- детекторы, использующие преобразования пространства яркости;
- детекторы, использующие статистические методы.

Основные критерии оптимизации [4]:

- высокая вероятность детектирования;
- высокая точность локализации;
- единственность отклика на один контрастный перепад.

Проблемы, представляющие интерес:

- улучшение точности алгоритмов;
- снижение порога ошибок;
- снижение количества «ложных» границ.

Концептуальная модель PSO

Формальное определение алгоритма PSO, представленного в данной работе, было предложено Кеннеди и Эберхартом [2], [3].

Каждая отдельная частица i в поколениях роя имеет следующие свойства: текущее положение в пространстве решений x_i , текущую скорость v_i , и персональное лучшее положение в пространстве решений y_i . Персональное лучшее положение – это такая позиция в пространстве решений, при которой оценочная функция $f(y_i)$ принимает максимальное значение. Позиция, в которой значение данной функции было наилучшим среди всех y_i , называется глобально лучшей позицией, и будет обозначаться как Y . Уравнения (1) и (2) задают то, как будут изменяться величины y_i и Y .

$$y_i(t+1) = \begin{cases} y_i(t), & \text{если } f(x_i(t+1)) \leq f(y_i(t)) \\ x_i(t+1), & \text{если } f(x_i(t+1)) > f(y_i(t)) \end{cases}, \quad (1)$$

$$Y(t) \in \{y_0(t), y_1(t), \dots, y_s(t)\},$$

$$f(Y(t)) = \max\{f(y_0(t)), f(y_1(t)), \dots, f(y_s(t))\}, \quad (2)$$

где в рое всего s частиц, т.е. $i \in \{1, \dots, s\}$.

Пусть для каждого из измерений пространства решений $j \in \{1, \dots, n\}$, $x_{i,j}$, $u_{i,j}$ и $v_{i,j}$ будут текущей позицией, текущей лучшей позицией и скоростью j -го измерения i -й частицы. Также введем некоторую меру инерционности частицы w и константы c_1 и c_2 . Тогда изменение скорости будет проходить по такому закону:

$$v_{i,j}(t+1) = wv_{i,j}(t) + c_1r_{1,j}(t)[y_{i,j}(t) - x_{i,j}(t)] + c_2r_{2,j}(t)[Y_j(t) - x_{i,j}(t)]. \quad (3)$$

Позиция частицы будет изменяться по закону:

$$x_i(t+1) = x_i(t) + v_i(t+1). \quad (4)$$

Значение каждого измерения любого вектора скорости v_i принадлежит диапазону $[-v_{\max}, v_{\max}]$ для того чтобы уменьшить тенденцию частиц покидать пространство решений. Значение v_{\max} считается исходя из значения $k \times x_{\max}$, где x_{\max} определяет границы пространства решений.

Инерционность частицы w используется для контроля сходимости частиц. При малых значениях w алгоритм быстро сходится к субоптимальным решениям, в то время большие предотвращают сходимость и значительно увеличивают время работы алгоритма. Обычно эта величина вычисляется экспериментально.

Коэффициенты c_1 и c_2 также вычисляются экспериментально и используются для контроля составляющих ускорения. Обычно они выбираются так, чтобы: $c_1 + c_2 = 2$.

Адаптация алгоритма PSO к задаче выделения границ

Пусть частица роя будет представлять собой случайную кривую, проходящую через заданный пиксель, а лучшим положением частицы (1) будет являться кривая, являющаяся потенциальной границей. Тогда алгоритм будет искать лучшие решения, исходя из оценочной функции, которую необходимо получить. Применяя данную функцию к каждому пикселю, в идеальном случае, мы получим набор кривых, каждая из которых представляет собой границу какого-либо объекта.

Пусть \max – это максимальное количество пикселей в кривой C , которая задаётся набором значений $d_1, d_2, \dots, d_{\max}$, где $d_i \in [0; 8]$. Числа от 1 до 8 обозначают направление перехода от предыдущего пикселя в кривой к следующему, а 0 обозначает отсутствие перехода и конец кривой.

Формальное представление частицы в виде кривой предложено в работе М. Сетайеш [1] и показано на рисунках 1 – 3:

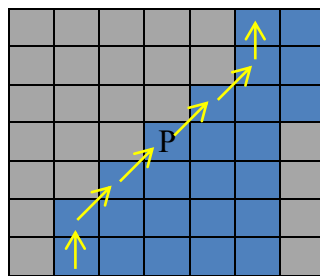


Рисунок 1 – Пример кривой C

7	8	1
6	P	2
5	4	3

Рисунок 2 – Направление перехода в пикселе Р на следующий пиксель в кривой С

8	1	1	1	1	8	0
---	---	---	---	---	---	---

Рисунок 3 – Пример представления кривой С в качестве частицы

Так как большая часть пикселей на границе имеет одинаковую или близкую интенсивность, введём два новых параметра:

1. Гомогенность кривой.
2. Похожесть пикселей кривой.

Чтобы подсчитать среднюю гомогенность кривой H_C , используем оператор гомогенности (общности) H_P пикселя Р и соседних восьми пикселей. Формально оператор записывается следующим образом:

$$H_P = \max \left(\left| I_P - I_{N_i} \right|, i=1..8 \right), \quad (5)$$

где Р – текущий пиксель, для которого считается H_P ;
 N_i – это i -й сосед пикселя Р;
 I_P – интенсивность пикселя Р.

Пусть порог ε – число от 0 до 255 включительно, вычисляемое экспериментально перед началом работы алгоритма, путём проверки участков на среднюю гомогенность пикселей. Этот порог показывает на сколько, в среднем, пиксели внутри ограничиваемой области похожи между собой.

Так как кривая – это потенциальная граница, то пиксели, принадлежащие кривой, должны отличаться от пикселей фона более чем на это пороговое значение ε . Поэтому справедлива следующая формула:

$$H_P = \begin{cases} H_P, & \text{если } H_P > \varepsilon \\ 0, & \text{если } H_P \leq \varepsilon \end{cases} \quad (6)$$

Гомогенность кривой – параметр, показывающий среднюю гомогенность пикселей, принадлежащих текущей кривой. Формальное описание:

$$H_C = \frac{1}{L_C} \sum_{i=1}^s H_{P_i}, \quad (7)$$

где s – количество пикселей в кривой;
 P_i – i -й пиксель кривой С;
 H_{P_i} – оператор гомогенности i -го пикселя.
 L_C – длина кривой С, которая вычисляется по формуле:

$$L_C = \sum_{i=1}^{s-1} \sqrt{\left(x_{P_i} - x_{P_{i+1}} \right)^2 + \left(y_{P_i} - y_{P_{i+1}} \right)^2}, \quad (8)$$

где (x_{P_i}, y_{P_i}) – координата пикселя P_i .

Второй параметр – похожесть пикселей кривой обозначим через U_C . Этот параметр показывает какая, в среднем, разница между интенсивностью начального пикселя кривой и остальными пикселями. Формально похожесть пикселей кривой определяется по формуле:

$$U_C = \frac{1}{L_C} \sum_{i=2}^s |I_{P_i} - I_{P_1}|, \quad (9)$$

Исходя из описанных параметров, можем задать функцию оценки кривой в качестве границы:

$$f_C = \begin{cases} -\infty, & \text{если } H_C \leq \varepsilon \\ (H_C - U_C) \times L_C, & \text{если } H_C > \varepsilon \end{cases}. \quad (10)$$

Формула показывает, что если средняя гомогенность кривой меньше порога (неотличима от фона), то функция принимает бесконечно малое значение. Если же средняя гомогенность кривой больше заданного порога, то результатом функции будет являться произведение длины кривой на разность средней гомогенности кривой и похожести пикселей кривой.

Подставляя найденную оценочную функцию f_C в формулы 1 и 2 для нахождения оптимальной позиции, получим глобально лучшую позицию Y , которая является искомой границей. Повторяя данную операцию для каждого пикселя, получаем набор кривых, которые представляют собой выделенные границы изображения.

Экспериментальное исследование

Так как алгоритм Кенни доказал свою эффективность в задаче выделения границ изображений, то с целью выявления преимуществ предложенного метода с использованием алгоритма PSO был разработан программный продукт на языке C# и проведена серия тестирований для сравнения полученных результатов с результатами алгоритма Кенни с использованием этого программного продукта.

Для тестирования были выбраны: изображение в высоком качестве, размытое по Гауссу изображение (с $\sigma = 15$, $\sigma = 20$ и $\sigma = 40$); размытое по Гауссу (с $\sigma = 15$) изображение с аддитивным Гауссовским белым и цветным шумом; изображение со случайным перемешиванием пикселей в матрице 4×4 .

Результаты приведены в табл. 1.

Таблица 1 – Сравнение времени обработки изображения

Качество изображения	T_{Canny} , сек	T_{PSO} , сек
<i>Изображение малого размера (131×316)</i>		
Высокое	0.03548512	0.307220011949539
Размыто по Гауссу ($\sigma = 40$)	0.03512357	0.323231786489487
Размыто по Гауссу ($\sigma = 20$)	0.03767823	0.304217398166656
Размыто по Гауссу ($\sigma = 15$)	0.03549946	0.299211412668228
Аддитивный цветной шум 25%	0.03559544	0.305219411849976
Аддитивный белый шум 25%	0.03648521	0.316227287054062
Аддитивный белый шум 50%	0.03548125	0.347244709730148
Перемешивание в матрице 4×4	0.03484567	0.314725995063782

Продолж. табл. 1

Изображение среднего размера (500×451)		
Высокое	0.1811322	10.8126821517944
Размыто по Гауссу ($\sigma = 40$)	0.1771204	9.7369270324707
Размыто по Гауссу ($\sigma = 20$)	0.1871326	11.5281524658203
Размыто по Гауссу ($\sigma = 15$)	0.1881236	10.0041036605835
Аддитивный цветной шум 25%	0.1861339	11.6022329330444
Аддитивный белый шум 25%	0.1911532	9.53176689147949
Аддитивный белый шум 50%	0.1921446	9.7297420501709
Перемешивание в матрице 4x4	0.2031572	8.10878276824951

Анализ результатов позволяет сделать следующие выводы:

- качество изображения не оказывает влияния на время работы обоих алгоритмов;
- увеличение размера изображения уменьшает время работы алгоритма PSO в 2 раза.

На рис. 4 представлены исходные изображения, использованные в работе.



Рисунок 4 – Исходные изображения

На рис. 5 показаны результаты распознавания границ зашумлённых изображений алгоритмами PSO и Кенни.

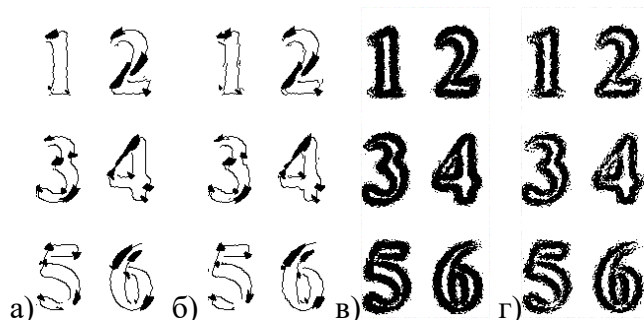


Рисунок 5 – Сравнение работы алгоритмов на цветном и белом шумах:

а – Кенни на 25% цветного шума; б – Кенни на 25% белого шума;

в – PSO на 25% цветного шума; г – PSO на 25% белого шума

Анализ изображений свидетельствует о том, что цветной шум оказывает значительно меньшее влияние на оба алгоритма в сравнении с белым.

На рис. 6 продемонстрированы результаты выделения границ на изображении с перемешанными пикселями в матрице 4×4.

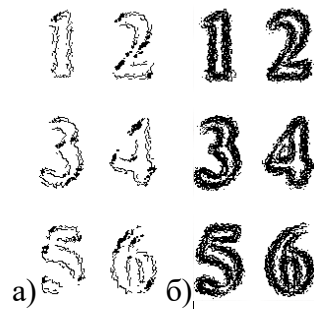


Рисунок 6 – Сравнение работы алгоритмов на перемешанном изображении:
а – Кенни; б – PSO.

Анализ изображений показывает, что метод на основе PSO более устойчив к перемешиванию пикселей.

На рис. 7 продемонстрирована работа алгоритмов на цветном изображении с 25% аддитивного шума.

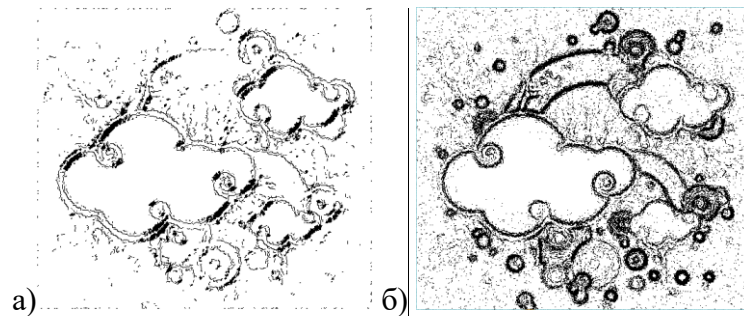


Рисунок 7 – Сравнение работы алгоритмов на цветном изображении с 25% белого аддитивного шума: а – Кенни; б – PSO.

Анализа изображений свидетельствуют о том, что и в этом случае PSO демонстрирует лучшие результаты.

Из рисунка 8 видно, что на цветном изображении с 50% шума работа PSO ухудшается вследствие отсутствия фильтра в алгоритме.

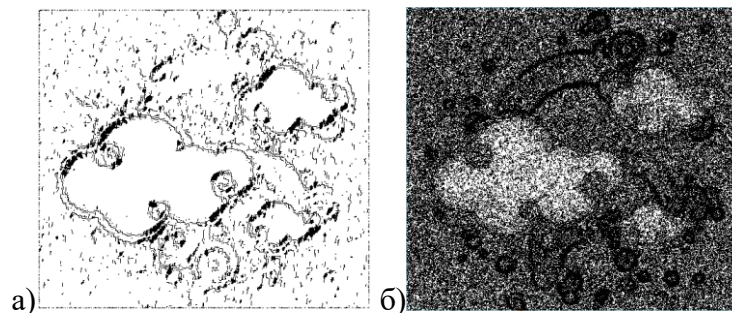


Рисунок 8 – Сравнение работы алгоритмов на цветном изображении с 50% белого аддитивного шума: а – Кенни; б – PSO.

Дополнительные результаты исследований представлены на рисунках 9 – 10.

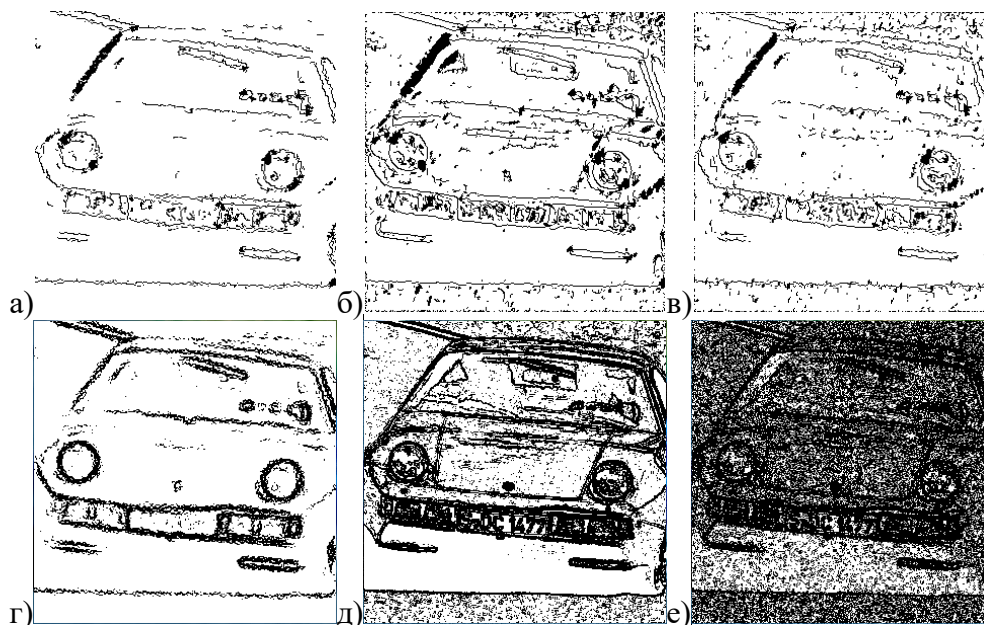


Рисунок 9 – Дополнительные результаты работы алгоритмов:
 а – Кенни на перемешанном изображении; б – Кенни на 25% белого шума;
 в – Кенни на 50% белого шума; г – PSO на перемешанном изображении;
 д – PSO на 25% белого шума; е – PSO на 50% белого шума;

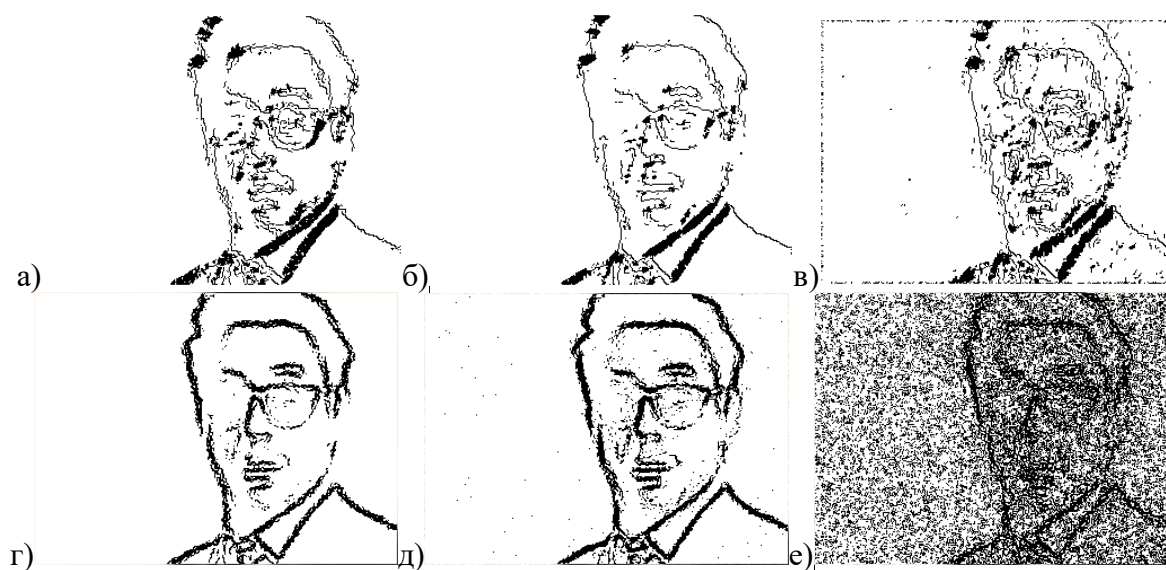


Рисунок 10 – Дополнительные результаты работы алгоритмов:
 а – Кенни на перемешанном изображении; б – Кенни на 25% белого шума;
 в – Кенни на 50% белого шума; г – PSO на перемешанном изображении;
 д – PSO на 25% белого шума; е – PSO на 50% белого шума;

Оценим качество полученных изображений путём сравнения с двумя эталонами, где эталонами будут являться границы изображений, полученные на изображении высокого качества алгоритмами Кенни и PSO (табл. 2).

Пусть цвет пикселя $P \in \{0,1\}$, где 0 обозначает белый цвет, а 1 – чёрный.

Введём функцию сравнения, которая может быть вычислена по формуле (11):

$$F = \frac{N_m}{N_p} \times 100\%, \quad (11)$$

где N_m – количество совпадающих пикселей (одного цвета) в эталоне и тестовом изображении;

N_p – количество пикселей на изображении.

Меняя эталон и тестовые изображения в функции, получаем 5 факторов сравнения:

- А – сравнение по пикселям результатов Кенни с эталоном Кенни;
- В – сравнение по пикселям результатов PSO с эталоном PSO;
- С – сравнение по пикселям результатов PSO с эталоном Кенни;
- D – сравнение по пикселям совпадений вдоль выделенной границы результатов PSO и эталона Кенни;
- E – сравнение по пикселям совпадений вдоль выделенной границы результатов Кенни и эталона Кенни.

Таблица 2 – Сравнение качественных характеристик

Описание изображения	A	B	C	D	E
Перемешанное монохромное изображение № 1	92,69%	82,48%	78,56%	83,9%	42,2%
25% белого шума, монохромное изображение № 1	92,96%	83,38%	78,33%	91,43%	55,23%
50% белого шума, монохромное изображение № 1	92,46%	80,36%	75,48%	90%	55,23%
Перемешанное цветное изображение № 2	84,53%	78,57%	81,78%	30,1%	32,2%
25% белого шума, цветное изображение № 2	86,13%	78,27%	77,98%	41,88%	24,81%
50% белого шума, цветное изображение № 2	82,07%	45,28%	36,56%	80,87%	26,09%
Перемешанное цветное изображение № 3	84,67%	78,40%	79,25%	30,33%	20,42%
25% белого шума, цветное изображение № 3	83,31%	87,01%	73,58%	85,25%	39,15%
50% белого шума, цветное изображение № 3	83,21%	52,80%	41,42%	90,47%	26,42%
Перемешанное цветное изображение № 4	92,80%	91,47%	90,06%	46,23%	55,23%
25% белого шума, цветное изображение № 4	94,04%	89,84%	89,47%	52,83%	50,50%
50% белого шума, цветное изображение № 4	90,39%	61,05%	59,36%	70,27%	53,32%

Анализ таблицы позволяет сделать вывод о том, что:

Алгоритм PSO в большей степени сохраняет правильные границы изображения после зашумления или перемешивания. Это видно, поскольку параметр $D > E$, что означает сохранность границы (отсутствие разрывов или ошибки локализации).

При перемешивании изображения и 25% зашумлённости алгоритм PSO даёт результат совпадений с эталоном Кенни близкий к совпадению результатов Кенни со своим эталоном ($C \approx A$), что говорит об отсутствии дефектов (шумовые зёрна, ложные границы).

При 25% белого шума параметр В принимает высокое значение, что подразумевает низкую степень отклонения от эталона полученных алгоритмом PSO границ.

Выводы

В ходе выполнения работы был разработан метод детектирования границ изображения с использованием алгоритма PSO. Применение предложенного метода позволяет получить границы изображения за приемлемый промежуток времени с высокой точностью локализации и отсутствием разрывов в кривых, даже при наличии значительных шумов.

При проведении экспериментов были получены оптимальные параметры запуска для алгоритма, которые можно исследовать для выявления зависимости между производительностью алгоритма и изменчивыми признаками внешней среды (освещенность, шумы и др.)

Созданная модель и её программная реализация могут быть платформой для решения более сложных задач, связанных с обработкой информации в сфере компьютерного зрения.

Список литературы

1. Setayesh M. Particle Swarm Optimisation for Edge Detection in Noisy Images /M. Setayesh // Research Archive [Электронный ресурс] URL: <http://researcharchive.vuw.ac.nz/xmlui/bitstream/handle/10063/2698/thesis.pdf?sequence=2> – 2013.
2. An efficient hybrid approach based on PSO, ACO and k-means for cluster analysis / Applied Soft Computing – 2010. – № 10 (1). – P. 183-197
3. Clerc M. Standard Particle Swarm Optimization / HAL open access archive. – 2012.
4. Canny J. A Computational Approach To Edge Detection / IEEE Trans. Pattern Analysis and Machine Intelligence – 1986. – 8(6). – P. 679-698.

References

1. Setayesh M. Particle Swarm Optimisation for Edge Detection in Noisy Images /M. Setayesh // ResearchArchive [URL] : <http://researcharchive.vuw.ac.nz/xmlui/bitstream/handle/10063/2698/thesis.pdf?sequence=2> – 2013.
2. An efficient hybrid approach based on PSO, ACO and k-means for cluster analysis / Applied Soft Computing 10 (1) – 2010. – 183-197.
3. Clerc, M. Standard Particle Swarm Optimization / HAL open access archive. – 2012.
4. Canny J., A Computational Approach To Edge Detection / IEEE Trans. Pattern Analysis and Machine Intelligence – 8(6):679–698. – 1986.

RESUME

V. N. Kotenko, I. N. Polishchuk

Method to Increase Edge Detection Quality Using Particle Swarm Optimization Algorithm

Background: technological progress in computer vision sphere led to a need for artificial intelligence systems being able to identify objects on images. Edge detection is a nontrivial task, and a lot of methods, which proved their advantage at particular sphere with a set of specific source parameters, were proposed to solve that task. Edge detection is used in different areas: from face detection to cancer diagnostics.

Materials and methods: references used in this paper: formal definition of PSO algorithm, proposed by J. Kennedy and R. Eberhart; Kenny edge detection algorithm; particle definition as an edge, proposed by M. Setayesh. Images were taken from open source storages. Image effects were achieved programmatically.

Results: experiments proved high localization accuracy, low level of errors and absence of broken edges for proposed PSO method compared with PSO.

Conclusion: a new method of edge detection was developed using PSO algorithm. Applying of this method allows getting image edges in decent time with high localization accuracy and without broken edges even at noisy images. During experiments optimal starting parameters were found. The use of filter is proposed to increase algorithms performance. Developed model and its program implementation can become a foundation for more complex computer vision problems.

Статья поступила в редакцию 25.02.2016.