



Ссылка на статью:

// Ученые записки УлГУ. Сер. Математика и информационные технологии. 2023, № 1, с.19-27.

Поступила: 19.12.2022

Окончательный вариант: 07.02.2023

© УлГУ

УДК 519.87 + 004.9 + 573.2

Фильтрация изображений на основе дискретного косинусного преобразования с использованием робастных индикаторов негауссовских распределений

Дементьев Н. А.^{*}, Бутов А. А.

*spelucker@gmail.com

УлГУ, Ульяновск, Россия

Рассматривается фильтр для изображений на основе дискретного косинусного преобразования с использованием робастного индикатора негауссовских распределений. Представлен алгоритм, который варьирует свои параметры в зависимости от исследуемой области, уменьшая пороговое значение в окрестностях границ объектов. Предложенный метод позволяет эффективно бороться с шумами на изображении и сохранять границы объектов.

Ключевые слова: дискретное косинусное преобразование, робастная оценка эксцесса, фильтрация изображений

Введение

В работе рассмотрен локально-адаптивный фильтр на основе дискретного косинусного преобразования (далее ДКП) (см. [2]) с использованием робастного индикатора негауссовских распределений, который эффективно убирает шумы на изображении, распределения и параметры которых заранее неизвестны.

ДКП применяется в двух основных целях. Первая – сглаживание и устранение шумов. Вторая – сжатие изображений, как раз на таком преобразовании и основан распространённый формат изображений JPEG. Изображения такого формата хранятся как координаты точек в многомерном пространстве ДКП. В работе основное внимание уделено первому варианту использования данного метода (см. [1, 3, 7, 8]).

Методы локальной фильтрации связаны общей идеей фильтрации со скользящим окном (или апертурой). В алгоритме такого фильтра берётся небольшое, по сравнению с

размерами самого изображения, окно, обычно квадратной формы, и на нём задаётся некоторая функция.

Робастность – это нечувствительность к малым отклонениям от предположений (см. [9]). К примеру, имеется некоторая выборка, и в этой выборке какое-либо одно значение резко выделяется от остальных наблюдений («выброс»). Такой случай может возникнуть при помехах аппаратуры или же при банальной ошибке ввода данных, поэтому необходимо оценивать и проверять гипотезы в подобных условиях, когда имеются некоторые малые отклонения от предположений.

1. Математическая модель

Рассмотрим систему ортогональных векторов $\{\vec{g}_1, \vec{g}_2, \dots, \vec{g}_N\}$ в многомерном пространстве \mathbb{R}^N . На плоскости можно представить такой вектор в виде N точек. Например, единичный вектор $\vec{g}_1 = \{\frac{1}{\sqrt{N}}; \frac{1}{\sqrt{N}}; \dots; \frac{1}{\sqrt{N}}\}$ представлен на рис. 1.

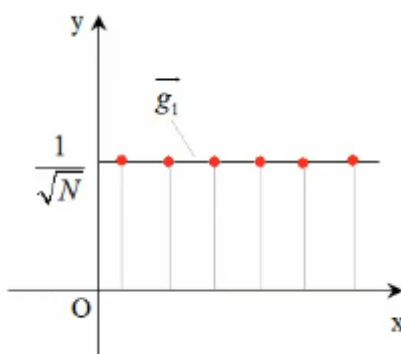


Рис. 1. Графическое представление вектора \vec{g}_1 .

Получается, что каждый из единичных векторов в многомерном пространстве превращается в некое подобие дискретной функции. Тогда нам ничего не мешает взять набор из каких-либо ортогональных функций, к примеру функций косинусов, и на основе этих функций составить новый ортогональный базис. Для простоты положим, что число функций равно восьми, по аналогии с алгоритмом JPEG. Размер выбран так, чтобы не сильно увеличивать вычислительную сложность алгоритма и при этом иметь минимум визуальной потери качества изображения. Зададим ортогональный базис в восьмимерном пространстве:

$$\begin{cases} f_1(w) = \cos(w * 0) \\ f_2(w) = \cos(w * 1) \\ \dots \\ f_8(w) = \cos(w * 7) \end{cases} \quad (1)$$

Очевидно, что первая функция системы (1) равна единице, то есть для первого базисного вектора мы можем взять восемь точек и получить его координаты: $\vec{g}_1 = \{\frac{1}{\sqrt{8}}; \frac{1}{\sqrt{8}}; \dots; \frac{1}{\sqrt{8}}\}$. Коэффициент $\frac{1}{\sqrt{8}}$ введён для того, чтобы нормировать вектор.

Вторая функция в системе (1) – это обычная косинусоида с периодом 2π . Если равномерно разбить период 2π и получить 8 точек, то некоторые из базисных векторов окажутся коллинеарными друг другу, то есть они будут лежать на одной прямой, но направлены в противоположные стороны. Такие векторы будут описывать одну и ту же координатную ось, а значит получится не полная система координат в восьмимерном пространстве, некоторые оси просто будут отсутствовать. Чтобы избежать таких векторов, следует взять интервал $[0; \pi]$ и уже его равномерно разбивать точками на восемь равных интервалов, то есть с шагом $\Delta = \frac{\pi}{8}$. Но если взять эти восемь точек на графике косинуса, то они распределяться не совсем равномерно (рис. 2), на конце будет не хватать одной граничной точки, из-за чего ортогональности не будет.

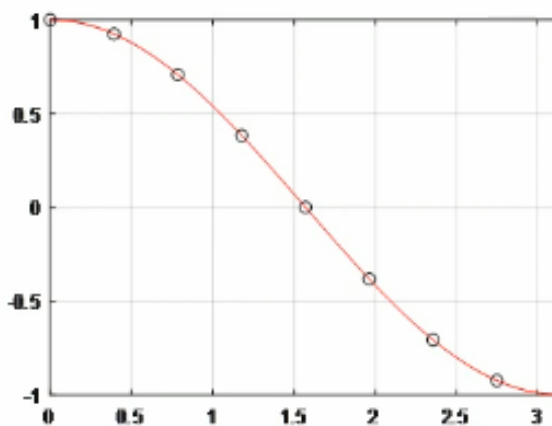


Рис. 2. График косинуса на интервале $[0; \pi]$

Чтобы точки расположились равномерно, необходимо их сдвинуть на половину шага $\theta = \frac{\Delta}{2} = \frac{\pi}{16}$. В итоге получается следующая формула для координат вектора:

$$g_n = \cos\left(\left(\frac{\pi}{N} * n + \frac{\pi}{2N}\right)\right) = \cos\left(\frac{\pi}{N} * \left(n + \frac{1}{2}\right)\right). \quad (2)$$

Подставляя приближенные значения косинуса, получим следующий вектор:

$$\vec{g}_2^* = \{0.9808; 0.8315; 0.5556; 0.1951; -0.1951; -0.5556; -0.8315; -0.9808\}. \quad (3)$$

Но квадратичная норма вектора (3), то есть квадрат длины, равен четырём, а нам необходимо, чтобы векторы были равны единице. Для того, чтобы исправить это, введём нормирующий коэффициент $\frac{1}{\sqrt{N/2}} = \frac{1}{\sqrt{4}}$, тогда второй базисный вектор будет равен $\vec{g}_2 = \frac{1}{\sqrt{4}} * \vec{g}_2^*$. И вот уже полученный вектор будет иметь норму, равную единице. По аналогии получают все остальные векторы (4) – (9), а общая формула имеет вид (10):

$$\vec{g}_3 = \frac{1}{\sqrt{4}} * \{0.9239; 0.3827; -0.3827; -0.9239; -0.9239; -0.3827; 0.3827; 0.9239\}. \quad (4)$$

$$\vec{g}_4 = \frac{1}{\sqrt{4}} * \{0.8315; -0.1951; -0.9808; -0.5556; 0.5556; 0.9808; 0.1951; -0.8315\}. \quad (5)$$

$$\vec{g}_5 = \frac{1}{\sqrt{4}} * \{0.7071; -0.7071; -0.7071; 0.7071; 0.7071; -0.7071; -0.7071; 0.7071\}. \quad (6)$$

$$\vec{g}_6 = \frac{1}{\sqrt{4}} * \{0.5556; -0.9808; 0.1951; 0.8315; -0.8315; -0.1951; 0.9808; -0.5556\}. \quad (7)$$

$$\vec{g}_7 = \frac{1}{\sqrt{4}} * \{0.3827; -0.9239; 0.9239; -0.3827; -0.3827; 0.9239; -0.9239; 0.3827\}. \quad (8)$$

$$\vec{g}_8 = \frac{1}{\sqrt{4}} * \{0.1951; -0.5556; 0.8315; -0.9808; 0.9808; -0.8315; 0.5556; -0.1951\}. \quad (9)$$

$$g_{k,n} = C_k \cos\left(\frac{\pi}{N}\left(n + \frac{1}{2}\right) * k\right), \quad C_k = \begin{cases} \frac{1}{\sqrt{N}}, & k = 0 \\ \frac{1}{\sqrt{\frac{N}{2}}}, & k \neq 0 \end{cases} \quad (10)$$

$$k = 0, 1, 2, \dots, N - 1, \quad n = 0, 1, 2, \dots, N - 1.$$

В формуле (10) k – номер базисного вектора, n – номер точки в k -м базисном векторе.

В итоге получается восьмимерная ортонормированная система координат на базе функций косинуса. Преобразование, построенное на таких векторах, называется дискретным косинусным преобразованием.

Теперь можно взять произвольный вектор $X = [x_1, x_2, \dots, x_N]^T$ и представить его в восьмимерном пространстве $A = \{\vec{g}_1, \vec{g}_2, \dots, \vec{g}_8\}^T$. Преобразование $Y = A * X$ называется прямым ДКП, а $X = A^T * Y$ – обратным ДКП или восстановлением.

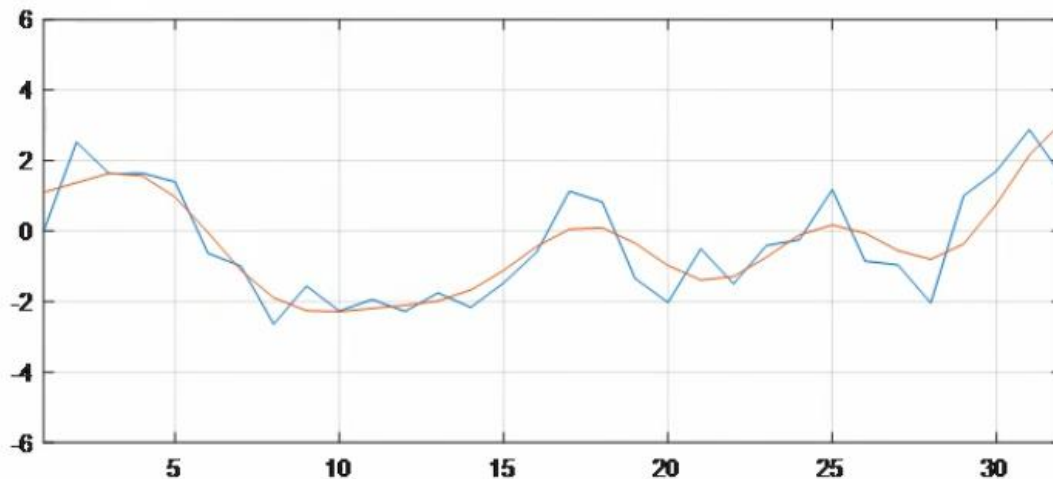


Рис. 3. Сглаживание с помощью ДКП

На рис. 3 был взят сигнал длиной в 32 отсчёта (голубой график), соответственно применялась 32-х мерная система координат ДКП. А затем этот вектор был восстановлен по первым десяти коэффициентам (коричневый график). То есть мы представили вектор как точку в 32-х мерном пространстве, оставили только значимые первые десять коэффициентов разложения, и при обратном переходе в 32-х мерную декартовую систему координат сигнал оказался сглаженным.

Алгоритм фильтрации на основе ДКП следующий. Для каждого положения скользящего окна рассчитываются коэффициенты ДКП, затем эти коэффициенты сравниваются с рассчитанными порогами и наименьшие из них обнуляются, такие значения и соответствуют помехам. Затем применяется обратное ДКП и восстанавливается изображение. Пороговое значение выбирают постоянным, равным $W = k\sigma$, где k рекомендуют брать равным 2.6. (см. [10]).

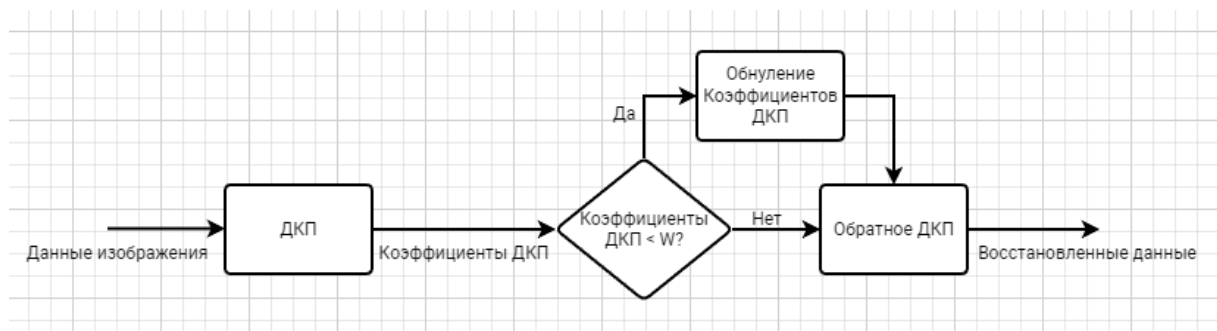


Рис. 4. Схема алгоритма ДКП.

Такой подход даёт хорошее качество изображения и неплохо справляется с помехами, но сильно сглаживает границы. Поэтому стоит использовать адаптивный подход. Локально-адаптивный подход заключается в уменьшении порога в окрестностях границ объектов. Обнаружить такие области можно с помощью параметра D_{ij} по следующей формуле (см. [10]):

$$D_{ij} = \frac{I_{ij}^{(p)} - I_{ij}^{(q)}}{Z_{ij}}, \quad (11)$$

где $Z_{ij} = \min_s (I_{ij}^{(t)} - I_{ij}^{(s)}), t = s + K, s = 1, \dots, N - K, K < N$.

При параметрах $p \approx 0.9N, q \approx 0.1N$, параметр (11) может быть рассчитан как робастная оценка эксцесса, которая определяется следующей формулой (см. [11]):

$$P_{ij} = \frac{I_{ij}^{(r)} - I_{ij}^{(c)}}{2(I_{ij}^{(p)} - I_{ij}^{(q)})}, \quad (12)$$

где $r \approx 0.75N, c \approx 0.25N$.

Локально-адаптивный метод можно описать системой (13), где параметр D_{thr} для выборок данных, соответствующих гауссовскому распределению, то есть однородным участкам изображения, примерно равен двум:

$$\begin{cases} W = k * \sigma, k < 2.6, & \text{если } D_{ij} > D_{thr} \\ W = 2.6 * \sigma, & \text{если } D_{ij} < D_{thr} \end{cases} \quad (13)$$

2. Применение алгоритма и анализ результатов

Для сравнения с предложенным в данной работе алгоритмом, применим к исходному зашумлённому изображению (рис. 5) простой сглаживающий фильтр с

апертурой 3×3 (см. [6]). В результате сглаживания интенсивность шума уменьшается, становится легче определить содержимое фотографии (рис. 6). Но изображение по-прежнему остаётся сильно зашумлённым, а границы объектов размываются тем сильнее, чем больше размер скользящего окна.



Рис. 5. Исходное изображение (иллюстративный материал заимствован из общедоступных ресурсов интернета, не содержащих указаний на авторов этих материалов и каких-либо ограничений для их заимствования, URL: https://kartinkin.net/pics/uploads/posts/2022-09/1662877336_22-kartinkin-net-p-kot-prokaznik-vkontakte-25.png)



Рис. 6. Результат сглаживающего фильтра с апертурой 3×3

Для представленного ДКП фильтра рассмотрим три случая: с апертурой 3×3 , 5×5 и 7×7 (рис. 7–9 соответственно). Предложенный алгоритм хорошо справляется с шумами, сохраняя границы объектов и мелкие детали. При увеличении размеров скользящего окна удаётся добиться полного исчезновения шумов на изображении, однако при слишком больших значениях апертуры с изображения начинают исчезать мелкие детали, что заметно на рис. 9. Также возрастают вычислительные затраты, что влияет на скорость работы алгоритма.



Рис. 7. Результат адаптивного ДКП фильтра с апертурой 3×3



Рис. 8. Результат адаптивного ДКП фильтра с апертурой 5×5



Рис. 9. Результат адаптивного ДКП фильтра с апертурой 7×7

3. Выводы

Нередко на изображениях присутствуют помехи сложного типа. Под этим термином подразумевается одновременное присутствие на изображении помех разной природы. Параметры такого изображения изменяются от точки к точке, характер помехи и её характеристики неизвестны, из-за чего применение простых неадаптивных линейных или нелинейных фильтров становится неэффективным. Таким образом, возникает задача разработки адаптивных алгоритмов, которые изменяли бы свои свойства в соответствии с изменяющимися характеристиками изображения, при отсутствии сведений о характере помех и их характеристик.

Представленный алгоритм может применяться для фильтрации изображений с импульсными или комбинированными помехами, где простое сглаживание демонстрирует плохие результаты. Данный алгоритм хорошо сохраняет границы объектов на изображениях, поэтому его можно использовать для распознавания объектов или для медицинских снимков, а также для решения множества других задач.

Список литературы

1. Аллен Б. *Think DSP. Цифровая обработка сигналов на Python*. М.: ДМК-пресс, 2017. 160 с.
2. Андреев В. Г., Нгуен Т. Ф. Адаптивная обработка сигналов на фоне комбинированных помех // *Изв. вузов. Радиоэлектроника*. 2015. Т. 58, № 2, с. 48-53.
3. Будаков Б. М., Фомин С. В. *Кратные интегралы и ряды*. М.: Наука, 1965. 607 с.
4. Бутов А. А. *Теория случайных процессов и ее дополнительные главы: учебное пособие*. Ч.1: Введение в стохастическое исчисление Ульяновск: УлГУ, 2016.
5. Бутов А. А., Раводин К. О. *Теория случайных процессов: учебно-методическое пособие*. Ульяновск: УлГУ, 2009. 62 с.
6. Гонсалес Р., Вудс Р. *Цифровая обработка изображений*. М.: Техносфера, 2005.
7. Ефимов А.В. *Математический анализ (специальные разделы). Ч. 1. Общие функциональные ряды и их приложение*. М.: Высшая школа, 1980. 279 с.
8. Степанов М. М., Миков С. Н. Оптимизация быстрого преобразования Фурье для различных процессов // *Ученые записки УлГУ. Сер. Физическая*. Ульяновск: УлГУ, 2005, вып. 1 (17), с. 82-90.
9. Смагин А. А., Валишин М. Ф. Устойчивые к атакам на контейнер стеганографические алгоритмы // *Инфокоммуникационные технологии*. 2015. Т. 13, № 1, с. 82-88.
10. Tsymbal O. V., Lukin V. V., Ponomarenko N. N., Zelensky A. A., Egiazarian K. O., Astola J. T. Three-state Locally Adaptive Texture Preserving Filter for Radar and Optical

Image Processing // *EURASIP Journal on Applied Signal Processing*. 2005. i. 8, p. 1185-1204.

11. Suoranta R. Amplitude domain approach to digital filtering. Theory and applications: Thesis for the degree of Doctor of Technology, Tampere, Finland, Tampere University of Technology, 199 (1995).

Image filtering based on discrete cosine transform using robust indicators of non-Gaussian distributions

Dementiev, N. A. ^{*}, Butov, A. A.

^{*}spelucker@gmail.com

Ulyanovsk State University, Russia

The paper considers a filter for images based on a discrete cosine transform using a robust indicator of non-Gaussian distributions. An algorithm is presented that varies its parameters depending on the area under study, decreasing the threshold value in the vicinity of the boundaries of objects. The proposed method allows you to effectively deal with noise in the image and preserve the boundaries of objects.

Keywords: *discrete cosine transform, robust kurtosis estimation, image filtering*