

# Сравнение некоторых алгоритмов определения центра тяжести изображения

*Емельянов Эдуард Владимирович*

30 сентября 2009 г.

## Введение

В задачах автоматической стабилизации изображения (возникающих, в частности, при астрономических наблюдениях) важной деталью является точность и помехоустойчивость алгоритма определения центра тяжести изображения. При низком уровне сигнал/шум на изображении значительные погрешности в определении координат центроида вносятся помимо собственно шумов дефектами изображения («горячими» и «битыми» пикселями) и космическими частицами.

В данной работе рассмотрена зависимость точности некоторых методов определения координат центра тяжести изображения от отношения сигнал/шум, величины смещения изображения (данный параметр особенно важен для корреляционных методов) и внутренних характеристик конкретного метода. Кроме того, для каждого метода приводится среднее быстродействие. Некоторые из приведенных алгоритмов рассматривались в [1, 3]. Отличием данной статьи является рассмотрение более широкого спектра алгоритмов и проведение экспериментального, а не теоретического анализа ошибок. В качестве быстродействия алгоритма выбрано значение временного интервала, необходимого для определения относительного смещения очередного кадра изображения без учета времени, затрачиваемого на формирование данного кадра. Вычисления производились на персональном компьютере со следующими характеристиками (чипсет — Intel 82801G):

```
> uname -srmp
Linux 2.6.27-desktop-0.rc8.2mnb i686 Intel(R) Core(TM)2 Duo CPU E4500 @ 2.20GHz
> cat /proc/meminfo | grep MemTotal
MemTotal:          449640 kB
```

Ошибки алгоритмов для каждого значения характеристик определялись путем усреднения результатов тридцати измерений. Для каждого уровня S/N рассчитывались также усредненные значения ошибок по девяти сериям из тридцати измерений с различными значениями сдвига изображения (от  $r = 0.80$  до  $r = 5.88$ , где  $r = \sqrt{(\Delta x)^2 + (\Delta y)^2}$ ). В каждом измерении генерировалось новое изображение с заданным сдвигом и уровнем шума, опорное изображение с заданным уровнем шума генерировалось отдельно для каждой серии. Каждый кадр содержал пять имитирующих изображения звезд гауссиан с различной амплитудой (от 0.59 до 1.53 единиц), распределенных по полю заданного размера. Полуширина всех гауссиан составляла  $1/\sqrt{2}$ .

# 1 Алгоритмы с пороговой величиной

## 1.1 Пороговое нахождение центра тяжести без фильтрации изображения

В данном алгоритме центр тяжести изображения вычисляется по формуле [1]

$$x_c = \frac{\sum_{I>I_T} x(I - I_T)}{\sum_{I>I_T} (I - I_T)}, \quad y_c = \frac{\sum_{I>I_T} y(I - I_T)}{\sum_{I>I_T} (I - I_T)}.$$

Здесь  $I$  – уровень интенсивности в данном пикселе. В качестве порога  $I_T$  выбирается какая-либо статистическая характеристика изображения. В данном случае  $I_T = \langle I \rangle + \sigma_I$  (сумма среднего значения интенсивности по изображению и ее среднеквадратического отклонения).

В исследовании использовался модифицированный алгоритм, использующийся в САО РАН: для коррекции на «горячие» пиксели и шумы в сумму включаются лишь пиксели, имеющие не менее одного соседа с заданным уровнем сигнала.

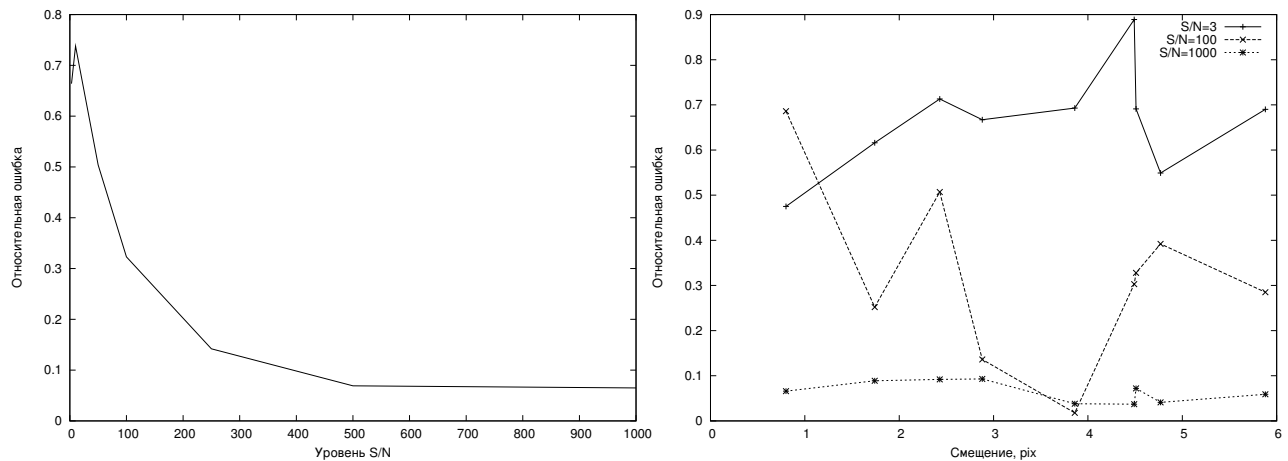


Рис. 1: Пороговое вычисление центра тяжести.  $S/N = (\max I^2)/\sigma_{noise}^2$ .

На рис. 1 (слева) приведена зависимость относительной ошибки определения изменения положения центра тяжести изображения от уровня сигнала/шум. Точность данного метода достаточно высока лишь при малом уровне шума. Однако, уже на уровне  $S/N=50$  (17 дБ) ошибка определения сдвига изображения возрастает до 50%. При еще меньших значениях  $S/N$  использовать рассматриваемый алгоритм нецелесообразно ввиду высокого уровня ошибок.

Несомненным достоинством данного алгоритма является скорость: относительное смещение кадра  $128 \times 128$  пикселей вычисляется в течение 0.3 мс. Благодаря такой скорости данный метод является наилучшим выбором при работе с незашумленными изображениями. Но учитывая то, что для незашумленных изображений центр тяжести можно вычислить по классической формуле

$$x_c = \frac{\sum_{x=1}^{size_x} x \cdot \sum_{y=1}^{size_y} I_{x,y}}{\sum I_{x,y}}, \quad x_c = \frac{\sum_{y=1}^{size_y} y \cdot \sum_{x=1}^{size_x} I_{x,y}}{\sum I_{x,y}},$$

сфера применения рассматриваемого алгоритма сильно ограничена.

Справа на рис. 1 показана зависимость относительной ошибки рассматриваемого метода от величины смещения исходного изображения для разных значений  $S/N$ . Несмотря на сильный разброс значений заметна тенденция снижения ошибки с увеличением  $S/N$ , как того и следовало ожидать.

## 1.2 Пороговый поиск объектов оператором лапласиана гауссианы

Оператор лапласиана гауссианы (**LoG**) представляет собой дифференциальный фильтр второго порядка. Применение к дискретному изображению только фильтра Лапласа некорректно, т.к. в данном случае будет наблюдаться значительная чувствительность к помехам. Поэтому до дифференцирования изображение необходимо сгладить фильтром Гаусса

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

Сглаженное изображение,  $I'$ , представляет собой свертку исходного,  $I$ , и фильтра Гаусса:  $I' = I \otimes \mathbf{G}$ . Локальные минимумы лапласиана сглаженного изображения будут соответствовать положению опорных элементов исходного изображения. Комбинация фильтров Лапласа и Гаусса и есть оператор лапласиана гауссианы:

$$\mathbf{LoG}(\sigma; x, y) = \nabla^2 \mathbf{G} = \frac{1}{\sigma^3\sqrt{2\pi}} \left(2 - \frac{x^2 + y^2}{\sigma^2}\right) e^{-\frac{x^2 + y^2}{2\sigma^2}}.$$

Первый множитель не влияет на результат определения положения объектов на изображении, поэтому его можно не учитывать.

Для ускорения процедуры расчета свертки изображения и фильтра используются преобразования Фурье. Пусть  $\mathcal{F}(I)$  — оператор прямого преобразования Фурье, а  $\mathcal{F}^{-1}(\mathfrak{F})$  — обратного. Тогда процедура обнаружения положения ключевых объектов на изображении сведется к поиску пикселей с  $I < I_T$  на преобразованном изображении:

$$I'' = \mathcal{F}^{-1}(\mathcal{F}(I) \cdot \mathcal{F}(\mathbf{LoG})), \quad \{x, y\}_{obj} = \{x, y\}_{I'' < I_T}, \quad I_T = \tau \cdot \min I''.$$

Коэффициент  $\tau$  выбирается экспериментально. В данном случае  $\tau = 0.25$  — такое значение коэффициента позволяет обнаружить и довольно слабые объекты. Для повышения точности метода он комбинировался с 1.1.

Так как преобразование Фурье фильтра  $\mathfrak{F} = \mathcal{F}(\mathbf{LoG})$  для изображений фиксированного размера вычисляется лишь при первом проходе, наиболее ресурсоемкой операцией при вычислении относительного сдвига каждого последующего кадра является вычисление двух преобразований Фурье. Для тестового изображения  $128 \times 128$  пикселей среднее время работы алгоритма составило 1.6 мс.

Для ускорения работы алгоритма в предфокальных системах автоматической коррекции в случае, когда относительное смещение изображения между соседними кадрами не превышает  $2 \div 3$  пикселей, возможна его адаптивная модификация. Для опорного изображения при помощи фильтра **LoG** вычисляется маска положения объектов (звезд)  $\{x, y\}$ . По этой маске вычисляются приближенные координаты центра тяжести следующего изображения:

$$x_c = \frac{\sum_{\{x,y\}} x \cdot I_{x,y}}{\sum_{\{x,y\}} I_{x,y}}, \quad y_c = \frac{\sum_{\{x,y\}} y \cdot I_{x,y}}{\sum_{\{x,y\}} I_{x,y}}. \quad (1.1)$$

Новая маска вычисляется лишь в том случае, когда сдвиг изображения по любой из осей превышает  $0.3\sigma$ .

На рис. 2 (слева) приведена зависимость относительной ошибки метода от уровня S/N для разных значений параметра сглаживания  $\sigma$ . На рис. 2 (справа) представлены для сравнения ошибки вычисления смещения кадров этим и описанным выше алгоритмами для сильно зашумленного сигнала. Выигрыш в погрешности при использовании метода LoG очень невелик по сравнению с пороговым методом. Однако, при использовании адаптивного LoG-метода для вычисления относительных смещений крупных изображений достигается значительный выигрыш по времени работы алгоритма.

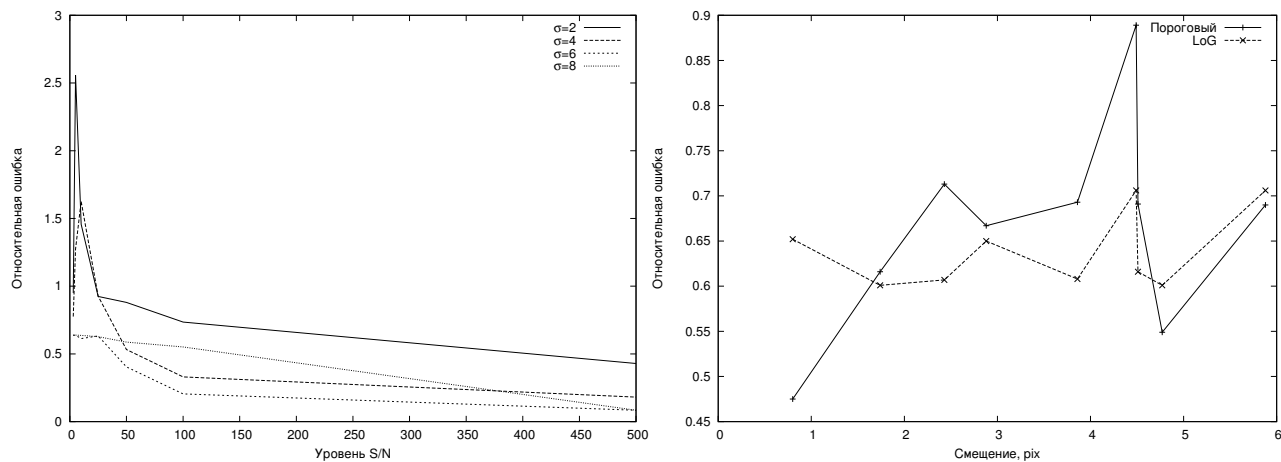


Рис. 2: Вычисление центра тяжести при помощи оператора **LoG** (рис. слева). Сравнение порогового метода и **LoG** при  $S/N=3$  (рис. справа).

## 2 Корреляционные алгоритмы

Данные алгоритмы основаны на поиске положения максимума корреляционной функции опорного и смещенного изображений. Основными ресурсоемкими операциями «чистого» корреляционного алгоритма являются преобразования Фурье: для определения смещения изображения необходимо выполнить прямое преобразование Фурье изображения, умножить полученный образ на образ предыдущего изображения и выполнить обратное преобразование Фурье для получения корреляционной функции.

Для нахождения координат максимума корреляционной функции можно использовать классический алгоритм поиска центра тяжести либо пороговый алгоритм. В данном случае использовался классический алгоритм в следующей модификации:

1. нахождение приближенных координат максимума корреляционной функции (перебор значений функции во всех пикселях изображения с сохранением координат пикселя с наибольшим значением);
2. уточнение координат максимума по формуле (1.1) в окрестности  $7 \times 7$  пикселя с максимальным значением корреляционной функции.

Так как процедура корреляции сильно зависит от уровня шумов (см. рис. 3), в чистом виде корреляционный алгоритм обычно не используется. Для очистки изображения от шумов можно использовать различные методы. Мы рассмотрим два из них.

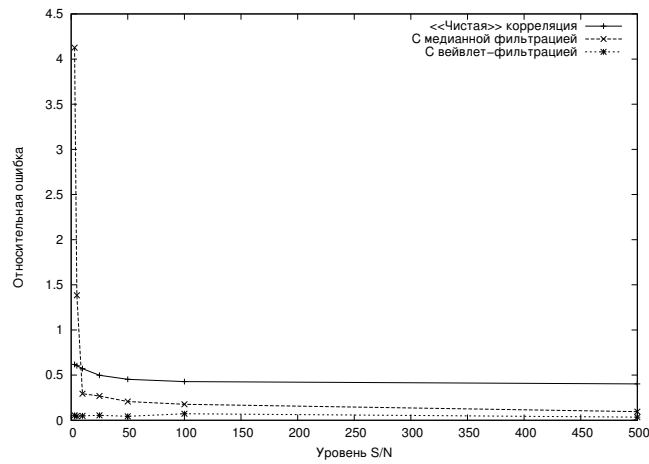


Рис. 3: Корреляционные методы вычисления центра тяжести изображения.

## 2.1 Корреляция с предварительной медианной фильтрацией

Одним из методов очистки изображения от шумов является медианная фильтрация. Ее достоинство заключается в отсечении шумов вида «соль/перец» (т.е. «плохих» и «горячих» пикселей), а недостаток — в сглаживании изображения, что приводит к уничтожению объектов с площадью меньше площади маски фильтра.

В нашем случае площадь изображений по полуширине меньше площади фильтра  $3 \times 3$  пикселя. Это привело к резкому возрастанию ошибки на малых  $S/N$  (см. рис. 3). На более чистых изображениях, как и следовало этого ожидать, медианная фильтрация в  $2 \div 4$  раза уменьшила ошибку определения смещения изображения корреляционным методом.

Итак, как и метод 1.2, данный метод не позволяет реализовать автоматическое гидирование объектов при малых  $S/N$ , а при больших значениях сигнал/шум его успешно заменяет пороговый или классический метод со значительным выигрышем в производительности.

## 2.2 Корреляция с предварительной вейвлет-фильтрацией

Вейвлет-преобразования нашли широкое применение в обработке изображений: как для их сжатия, так и для фильтрации. В очистке изображений от артефактов типа «соль-перец», а также гауссова размывания, хорошие результаты показывает фильтрация вейвлетами Дюбеши [2]. Наиболее подходящими для астрономических изображений являются вейвлеты Дюбеши четвертого порядка.

Алгоритм вейвлет-фильтрации заключается в нахождении вейвлет-базиса изображения, обнулении коэффициентов с порядком выше некоторого порогового значения  $T_{res}$  и выполнении обратного вейвлет-преобразования полученного базиса. Величина  $T_{res}$  равна количеству деталей, присутствующих на восстановленном изображении. От выбора значения этого порога зависит качество восстановленного изображения: при слишком малых значениях  $T_{res}$  часть объектов и их деталей будет отфильтрована, при слишком высоких — на изображении появится шум. При размере тестового изображения  $128 \times 128$  пикселей максимальное значение величины  $T_{res}$  составит 16384. Минимальным значением этой величины в данном случае будет 5 (т.к. тестовое изображение содержит пять объектов). Эта зависимость наглядно иллюстрируется рисунком 4 (слева). При значении  $T_{res} = 2$  на малых  $S/N$  ошибка очень велика, т.к. количество объектов больше количества аппроксимирующих вейвлетов, в результате чего на двух соседних изображениях возможен выбор разных объектов, по которым вычис-

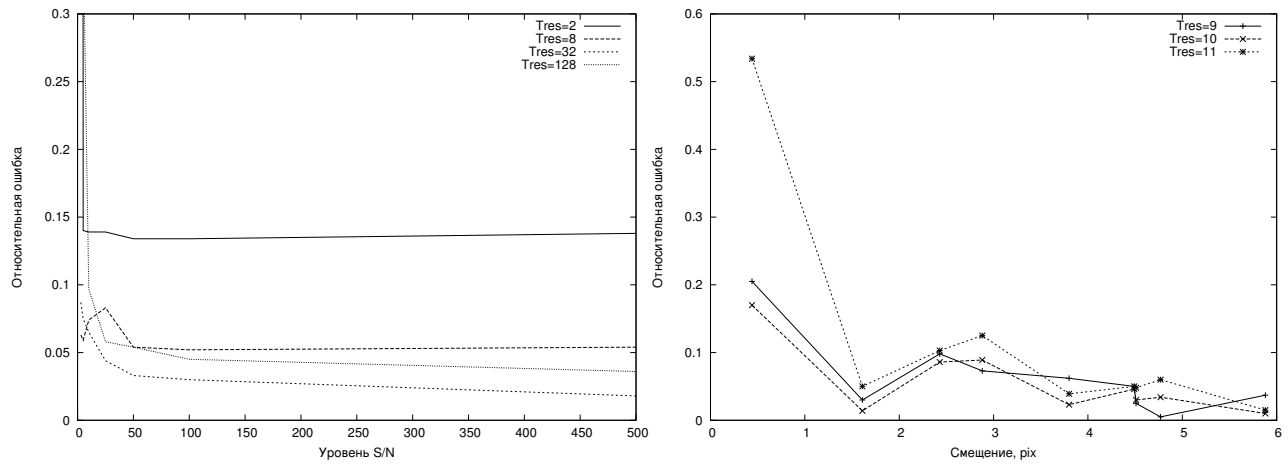


Рис. 4: Зависимость погрешности корреляционного метода с вейвлет-фильтрацией от порогового значения фильтра,  $T_{res}$ .

ляется общий центр тяжести. Кроме того, ошибка велика и при больших значениях  $S/N$  за счет недостаточного количества деталей в восстановленных изображениях.

При умеренном значении  $T_{res} \sim 10$  ошибка определения относительного смещения изображений имеет наименьшее значение благодаря четкому выделению объектов и их дополнительных деталей и практически полному отсутствию шумов на восстановленном изображении.

При высоких значениях  $T_{res} > 100$  погрешность возрастает и особенно сильно — для малых  $S/N$ , т.к., как и говорилось выше, в этом случае увеличивается количество артефактов на восстановленном изображении.

В нашем случае наилучшие результаты по определению относительного смещения изображений для малых смещений получились при  $T_{res} = 9 \div 10$  (см. рис. 4, справа). В среднем при  $T_{res} = 9 \div 32$  абсолютная ошибка держится на уровне 0.25 пикселей, что является наилучшим результатом из всех приведенных методов.

К недостаткам данного метода следует отнести зависимость его погрешности от выбора значения пороговой величины  $T_{res}$  и наименьшую скорость: для изображения  $64 \times 64$  пикселя среднее время вычисления относительного сдвига очередного изображения составляет 1 мс, для изображения  $128 \times 128$  пикселей — 4 мс, а для изображения  $256 \times 256$  пикселей — уже 56 мс.

## Заключение

Выстроим рассмотренные алгоритмы по возрастанию точности и по убыванию времени работы с пробным изображением  $128 \times 128$  пикселей при уровне  $S/N=3$ .

По возрастанию точности:

$$\text{CoG} \Rightarrow \text{TCoG} \Rightarrow \text{LCoG} \Rightarrow \text{MCorr} \Rightarrow \text{Corr} \Rightarrow \text{WCorr}$$

По возрастанию скорости:

$$\text{WCorr} \Rightarrow \text{MCorr} \Rightarrow \text{LCoG} \Rightarrow \text{Corr} \Rightarrow \text{TCoG} \Rightarrow \text{CoG}$$

Здесь  $\text{CoG}$  — классический алгоритм поиска центра тяжести изображения;  $\text{TCoG}$  — пороговый алгоритм;  $\text{LCoG}$  — пороговый алгоритм с построением маски лапласиана гауссианы

(скорость данного алгоритма значительно возрастает в случае малых относительных смещений изображений); *CoG* – определение центра тяжести по корреляционной функции (корреляционный алгоритм); *MCoG* – корреляционный алгоритм с предварительной медианной фильтрацией изображения; *WCoG* – корреляционный алгоритм с предварительной вейвлет-фильтрацией изображения.

Несомненным лидером по точности ( $\sim 0.35$  полуширин «звезды»-объекта) для сильно зашумленных изображений является метод *WCoG*. Естественным лидером в скорости (сотые доли микросекунды) является классический алгоритм. Однако, несмотря на значительную ресурсоемкость алгоритма *WCoG* (на два порядка выше, чем у *CoG*), он является наиболее подходящим для программного обеспечения автоматических гидрирующих устройств. При выборе опорного участка поля, по которому будет проводиться гидирование, можно ограничиться наиболее оптимальным его размером. Например, при размере участка  $64 \times 64$  пикселя, средняя скорость обработки информации может составлять до  $\sim 50$  кадров в секунду, при размере  $128 \times 128$  — до  $\sim 30$  кадров в секунду (с учетом времени получения изображения и выделения опорного участка).

## Список литературы

- [1] Comparison of centroid computation algorithms in a Shack-Hartmann sensor / S. Thomas, T. Fusco, A. Tokovinin et al. // *Mon. Not. of Royal Astron. Soc.* — 2006. — Vol. 371. — Pp. 323–336.
- [2] Junquan H., Daren H., Zhang Z. Wavelet technique recovering image blurred by mixed gaussian and salt-pepper noise // *Analisis in Theory and Applications.* — 2003. — Vol. 19, no. 2. — Pp. 185–192.
- [3] Study of optimal wavefront sensing with elongated laser guide stars / S. J. Thomas, S. Adkins, D. Gavel et al. // *Mon. Not. of Royal Astron. Soc.* — 2008. — Vol. 387. — Pp. 173–187.