

Параллельные реализации алгоритмов нахождения зеркальной симметрии бинарных растровых изображений

С.А. Федотова^а, О.С. Середин^а, О.А. Кушнир^а, В.В. Сулимова^а

^а Тульский государственный университет, 300012, пр. Ленина, 92, Тула, Россия

Аннотация

Оценка симметричности фигур является важным этапом анализа бинарных изображений и может применяться для решения многих прикладных задач компьютерного зрения, таких, например, как анализ условий произрастания растений, билатеральной симметрии насекомых. Известные алгоритмы поиска оси зеркальной симметрии позволяют найти лишь приближенное решение данной задачи, как правило, не предоставляя возможности оценить качество полученного решения. Естественным способом оценки качества в данном случае является сравнение с точным решением - эталонной осью симметрии, мера симметричности изображения относительно которой имеет максимальное значение. В данной работе исследуется точный метод поиска такой эталонной оси симметрии, основанный на полном переборе всех потенциально возможных осей и оценке симметричности фигуры с использованием теоретико-множественного подобия Жаккарда, применяемого к подмножествам пикселей фигуры при делении ее осью. Алгоритм полного перебора гарантированно находит эталонную ось симметрии, но требует весьма значительного времени на обработку каждого изображения. Для достижения скорости, позволяющей работать с большими базами изображений, была разработана параллельная версия данного алгоритма, которая была реализована на языке C++ с применением технологии параллельного программирования MPI и протестирована с использованием ресурсов суперкомпьютерного комплекса МГУ имени М.В. Ломоносова. Экспериментальные исследования на базе изображений «Бабочки» показали, что предложенный алгоритм позволяет найти эталонную ось симметрии за время, приемлемое для обработки баз, состоящих из сотен и тысяч изображений, что сделало возможным его применение для автоматической разметки баз изображений, а также отладки и тестирования на них предложенных ранее авторами приближенных процедур поиска оси симметрии. Разработанная параллельная версия одного из приближенных алгоритмов обеспечивает возможность решения прикладных задач анализа изображений в условиях, близких к режиму реального времени, позволяя достичь времени обработки, исчисляемого в долях секунды даже на обычных многоядерных персональных компьютерах, сохраняя при этом максимальное, либо близкое к максимальному качество решения.

Ключевые слова: высокопроизводительные вычисления; зеркальная симметрия; бинарные растровые изображения; технологии распараллеливания; технология MPI; технология OpenMP; суперкомпьютерный комплекс МГУ имени М.В. Ломоносова

1. Введение

Анализируя бинарные изображения можно заметить, что формам ряда объектов как искусственного, так и природного происхождения присуща зеркальная (осевая) симметрия. Очевидно, что изображения реальных объектов редко бывают идеально симметричны. Поэтому представляет интерес задача поиска приближенной симметрии и оценки степени симметричности изображения (см. рис. 1). Оценка симметричности фигур может применяться для решения многих задач компьютерного зрения, таких как анализ условий произрастания растений, билатеральной симметрии насекомых.

Задача определения симметрии и нахождения меры симметричности для бинарных растровых изображений известна, но существует достаточно небольшое число эффективных методов для ее решения, основывающихся на: 1) параметрическом представлении контура фигуры и разложении его в ряд Фурье [1], 2) представлении контура фигуры функцией поворота [2], 3) представлении контура фигуры критическими точками и вычислении сходства двух подконтуров при помощи векторов геодезических расстояний фигуры [3], 4) парном сравнении подцепочек скелетных примитивов [4,5]. Перечисленные методы базируются на известных алгоритмах вычисления различия (или сходства) форм фигур.

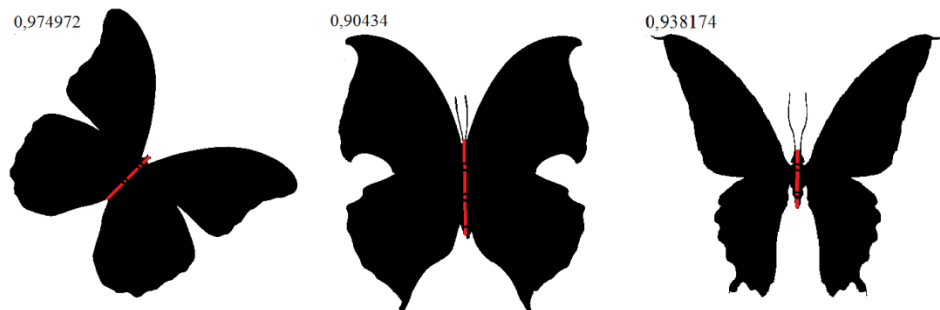


Рис. 1. Примеры изображений с осью, соответствующей максимальному значению меры симметричности, вычисленной на основании подобия Жаккарда.

Для оценки и сравнения результатов работы алгоритмов определения зеркальной симметрии бинарных изображений в работе [5] был предложен точный алгоритм, который позволяет вычислить эталонную ось приближенной симметрии, а также меру симметричности изображения.

Точный алгоритм оценки симметрии основан на полном попарном переборе точек внешнего контура, суть которого заключается в следующем: через пару точек проводится прямая, которая рассматривается как возможная ось симметрии фигуры. Ось делит фигуру на две части, они представляются как два множества точек-пикселей, сходство между которыми вычисляется по мере Жаккарда:

$$\mu_T(B) = \frac{|S(B) \cap S(B_r)|}{|S(B) \cup S(B_r)|} \quad (1)$$

где B – бинарное изображение, яркость черных пикселей которого обозначим 1, белых – 0; B_r – изображение, полученное отражением бинарного изображения B относительно прямой, $S(B)$ – множество пикселей изображения B , яркость которых равна 1.

Прямая, которая делит фигуру на два наиболее схожих множества (мера Жаккарда на них максимальна), считается искомой осью симметрии фигуры. К сожалению, значения меры симметрии, построенные для всех пар точек контура фигуры, представляют собой крайне сложную поверхность в пространстве поиска (см. рис. 2); приходится применять алгоритм полного перебора возможных осей симметрии, который имеет очень большую трудоемкость – десятки и даже сотни минут работы на обычном персональном компьютере.

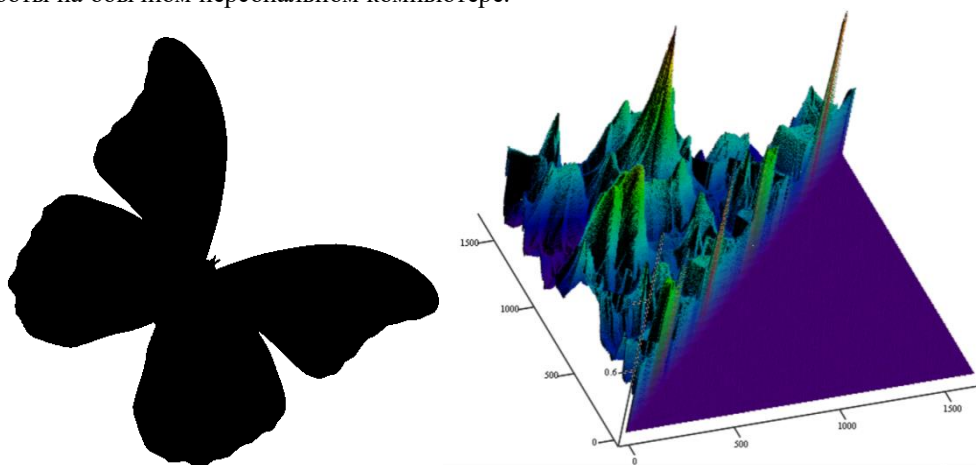


Рис. 2. Пример изображения и значения функции симметрии для вариантов осей, проходящих через все $N \cdot (N - 1) / 2$ пары точек контура.

Ранее были предложены две ускоренные версии – оптимизация с учетом полупериметра фигуры и оптимизация с учетом центра масс фигуры [5,12]. Несмотря на то, что подобные оптимизации точного алгоритма существенно сокращают время работы, их скорость все равно недостаточна для обработки больших баз изображений. Также для достижения максимальной точности, т.е. получения эталонной оси симметрии данными алгоритмами, необходим корректный подбор параметров алгоритмов для каждого изображения, что делает процесс обработки больших баз изображений очень трудоемким.

В этой работе описываются две параллельные реализации предложенных ранее алгоритмов [4,5,7]. Во-первых, будет рассмотрена версия алгоритма полного перебора для запуска на суперкомпьютере, что позволит получать точные оценки симметричности фигуры за время, приемлемое для обработки больших баз изображений. Полученная разметка (знание о положении оси симметрии на фигуре) позволит отлаживать варианты неточных, но быстрых методов поиска симметрии. Во-вторых, предложена параллельная версия одного из таких приближенных методов, а именно метода доуточнения пробной оси симметрии [7], полученной на основе скелета бинарного растрового изображения [4,5]. Это обеспечит возможность решения прикладных задач анализа изображений в условиях, близких к режиму реального времени, позволяя достичь времени обработки, исчисляемого в долях секунды, даже на обычных многоядерных персональных компьютерах, сохраняя при этом максимальное, либо близкое к максимальному качество решения.

2. Параллельная версия алгоритма поиска оси зеркальной симметрии полным перебором для суперкомпьютера

Алгоритм полного перебора гарантированно находит эталонную ось симметрии, но имеет квадратичную вычислительную сложность относительно количества точек контура фигуры. Среднее время обработки одного изображения с разрешением 800 на 600 пикселей составляет порядка двух часов на обычном ноутбуке. Для достижения приемлемого времени работы точного алгоритма на больших базах изображений было принято решение применить технологии распараллеливания, позволяющие использовать значительные вычислительные мощности. В частности, мы использовали ресурсы суперкомпьютерного комплекса МГУ имени М.В. Ломоносова.

Последовательная реализация точного алгоритма, основанного на полном переборе линий-кандидатов в искомую ось симметрии:

1. контур фигуры представляется как последовательность, состоящая из граничных точек фигуры;
2. из последовательности выбирается одна точка;

3. перебираются все оставшиеся точки, каждая из которых вместе с точкой, выбранной на шаге 1, определяет некоторую линию. Вычисляется и запоминается мера симметрии (1) относительно каждой полученной линии. Так как будут рассмотрены все возможные линии, проходящие через точку, выбранную на шаге 1, данную точку можно исключить из дальнейшего перебора;

4. шаги 2 и 3 повторяются для уменьшенной на единицу последовательности точек контура.

В результате работы алгоритма будут рассмотрены все возможные линии, пересекающие фигуру. Осью приближенной симметрии фигуры будет та из них, относительно которой получается максимальная мера симметричности.

Следует отметить, что вычисление меры симметрии относительно каждой линии выполняется независимо, что означает наличие параллелизма по данным, который может быть естественным образом использован для разработки параллельной версии данного алгоритма путем распределения обрабатываемых линий между процессами приложения, реализующего точный алгоритм.

Для распределения $L = C_N^2 = \frac{N!}{(N-2)!2!} = \frac{(N-1)N}{2}$ линий между P процессами в данной работе используется

следующая схема: первые $L \bmod P$ процессов получают $l = (L - L \bmod P) / P + 1$ линий, а оставшиеся $L - L \bmod P$ процессов получают $l = (L - L \bmod P) / P$ линий, где \bmod - операция взятия остатка от целочисленного деления.

Параллельная реализация алгоритма на основе полного перебора для P процессов:

1. каждый процесс определяет согласно описанной выше схеме набор из l линий, которые ему необходимо обработать;

2. каждый процесс последовательно перебирает линии из своего набора, вычисляет для каждой линии меру симметрии и выбирает среди своих линий ту, для которой мера симметрии имеет максимальное значение;

3. каждый процесс осуществляет передачу информации о найденной им линии и соответствующее ей значение меры симметрии на нулевой процесс.

4. нулевой процесс осуществляет прием информации о найденных линиях от всех остальных $P-1$ процессов и среди P полученных линий (включая найденную им самим) выбирает ту, для которой значение меры симметрии максимально.

Очевидно, что такое построение алгоритма гарантирует нахождение линии, значение меры симметрии для которой является максимальным среди полного множества из L линий, и которая, таким образом, является эталонной осью симметрии.

Данный алгоритм был реализован на языке C++ с применением технологии параллельного программирования MPI [10] для организации обмена данными между параллельно работающими процессами. Выбор технологии программирования MPI в данном случае обусловлен двумя факторами: 1) особенностью приведенного алгоритма, который содержит множество независимых, но требует продолжительных вычислений фрагментов и 2) особенностью архитектуры используемой вычислительной системы – суперкомпьютерного комплекса МГУ имени М.В. Ломоносова, имеющего несколько тысяч восьмипроцессорных узлов. Разработанная параллельная реализация предложенного параллельного алгоритма позволяет максимально эффективно использовать ресурсы вычислительной системы.

В разделе 4 приведены результаты тестирования производительности разработанной параллельной версии программы с использованием ресурсов суперкомпьютерного комплекса МГУ имени М.В. Ломоносова. Полученные результаты показывают эффективность предложенного метода. Для изображений из базы «Бабочки» удается снизить время поиска оси симметрии до одной-пяти минут. Стало возможным рассчитать оси симметрии для больших баз изображений. Так, вся база Flavia, состоящая из 1907 бинарных изображений листьев растений [8] разрешением 800 на 600 пикселей была обчислена менее чем за 9 часов с использованием 1024 процессов.

Однако, для работы в условиях производства необходимы скорости, сопоставимые с реальным временем (десятки миллисекунд при обработке потокового видео), или близкие к реальному (сотни миллисекунд для работы промышленных роботов), причем достижимые на персональных компьютерах. Ниже описывается параллельная версия быстрого алгоритма поиска оси симметрии, близкой к эталонной.

3. Параллельная версия алгоритма уточнения оси зеркальной симметрии для персонального компьютера

Для вычисления меры симметричности в задачах, требующих принятия решений в условиях, приближенных к реальным, параллельная версия точного алгоритма не применима: невозможно привлечь суперкомпьютер при решении повседневных прикладных задач. Поэтому для обеспечения возможности обработки изображений в режиме, близком к режиму реального времени даже на обычном персональном многоядерном компьютере, была разработана еще одна параллельная программа, в основу которой положен предложенный ранее в работе [7] алгоритм уточнения оси зеркальной симметрии, найденной скелетным методом. Этот метод является приближенным методом поиска, но имеет чрезвычайно высокую скорость работы [5].

Предварительные исследования показали, что ось, найденная скелетным методом, обычно дает меньшую меру симметричности по сравнению с осью, полученной полным перебором и доставляющей максимальную меру симметричности данному изображению. Тем не менее, скелетная ось располагается таким образом, что пересекает контур фигуры в некоторой ε -окрестности каждой из точек пересечения эталонной оси симметрии с контуром этой фигуры. Для повышения точности скелетного метода найденную им ось симметрии мы уточняем, т.е. находим в

некоторой ее окрестности прямую, которая будет иметь большую меру симметричности. Уточняемая скелетная ось называется затравочной, а любая ось-кандидат в процессе поиска уточненной оси – пробной. Так как ось симметрии хотя бы дважды пересекает контур объекта, для построения пробных осей рассматриваются только граничные точки фигуры. Контур изображения представлен последовательностью точек, пронумерованных от 0 до $N-1$.

Из приведенных в работе [7] результатов видно, что наилучшими показателями по времени и точности обладает второй из вариантов алгоритма уточнения оси симметрии, основанный на учете расположения центра масс фигуры. Алгоритм строится на том факте, что ось идеальной симметрии обязательно проходит через центр масс фигуры. Как правило, ось приближенной симметрии проходит не точно через центр масс, а в некоторой его окрестности R , которую условно будем рассматривать как окружность с центром, совпадающим с центром масс фигуры. Радиус R такой окружности рассчитывается как $k_R \cdot D$, где D – расстояние от центра масс до самой удаленной точки контура, k_R – коэффициент близости к центру масс. Если имеются априорные знания о качестве затравочной оси, то будет достаточно перебрать только те пробные оси, которые находятся не далее, чем на расстоянии R от центра масс, задаваемом через параметр k_R . Только они пересекают окружность с радиусом R , центром которой является центр масс.

Приведенный алгоритм также имеет некоторый ресурс внутреннего параллелизма, связанный с перебором всех пробных линий из найденной окрестности и вычислением соответствующих им значений меры симметричности. Данные операции, аналогично предыдущему случаю, являются независимыми и могут быть выполнены параллельно для разных пробных линий. Применение параллельных вычислений в данном случае может позволить осуществить дополнительное повышение производительности, что и является целью данной работы.

Поскольку разрабатываемую параллельную версию программы предполагается использовать на обычном персональном многоядерном компьютере, то в качестве технологии параллельного программирования была выбрана технология OpenMP [9,11], предназначенная для систем с общей памятью и предоставляющая удобные инструменты манипулирования потоками в рамках одного приложения.

Алгоритм уточнения оси симметрии, найденной скелетным методом, учитывающий центр масс фигуры, для T потоков:

1. Затравочная ось определяется двумя точками пересечения оси, найденной скелетным методом, с контуром: p_1 и p_2 .

2. Определить точки, находящиеся в некоторой заданной окрестности ε от первой точки: $a = p_1 - \varepsilon$, $b = p_1 + \varepsilon$. Эти две точки (a и b) ограничивают конечный набор некоторых точек контура $[a;b]$. Определить точки, находящиеся в некоторой заданной окрестности ε от второй точки: $c = p_2 - \varepsilon$, $d = p_2 + \varepsilon$. Эти две точки (c и d) ограничивают конечный набор некоторых точек контура $[c;d]$.

3. На отрезках $[a;b]$ и $[c;d]$ выбрать множества равноудаленных точек $Q = \{q_i = a + h \cdot i, i = 0, \dots, n\}$, $S = \{s_i = c + h \cdot i, i = 0, \dots, n\}$, где n – число участков разбиения отрезков, h – шаг разбиения, который рассчитывается как $2\varepsilon / n$.

4. Парно перебрать точки множеств Q и S , полученные в п.3. Этот перебор задает $l = |Q| \cdot |S|$ пробных линий, рассмотрение которых статически распределяется между T потоками. Вычислить меру симметрии относительно каждой полученной пробной оси, находящейся в окрестности R центра масс, запомнить ту из них, для которой получена максимальная мера симметрии.

5. Из T линий, представленных каждым потоком, необходимо выбрать одну линию с максимальной мерой симметричности.

6. Если шаг h больше 1, то две точки p'_1 и p'_2 , относительно которых получен максимум меры симметрии, передаются в п. 2: $p_1 := p'_1$, $p_2 := p'_2$, $\varepsilon := h$, иначе переход на п. 7.

7. Если найденная прямая лежит на границе поиска (любая ее точка совпадает с одной из точек a, b, c, d), то она объявляется затравочной, и описывающие её точки p'_1 и p'_2 передаются в п. 1, иначе найдена прямая, доставляющая максимум мере симметрии.

В результате работы алгоритма будет найдена ось, мера симметричности относительно которой будет не меньше меры, полученной скелетным методом.

Предложенная параллельная версия алгоритма уточнения оси симметрии реализована языке C++ с применением технологии параллельного программирования OpenMP [9,11].

4. Экспериментальные исследования

Предложенные методы поиска оси зеркальной симметрии экспериментально исследованы на базе бинарных растровых изображений «Бабочки», в которой представлено 30 изображений размером примерно 400 на 600 пикселей (см. рис. 3).

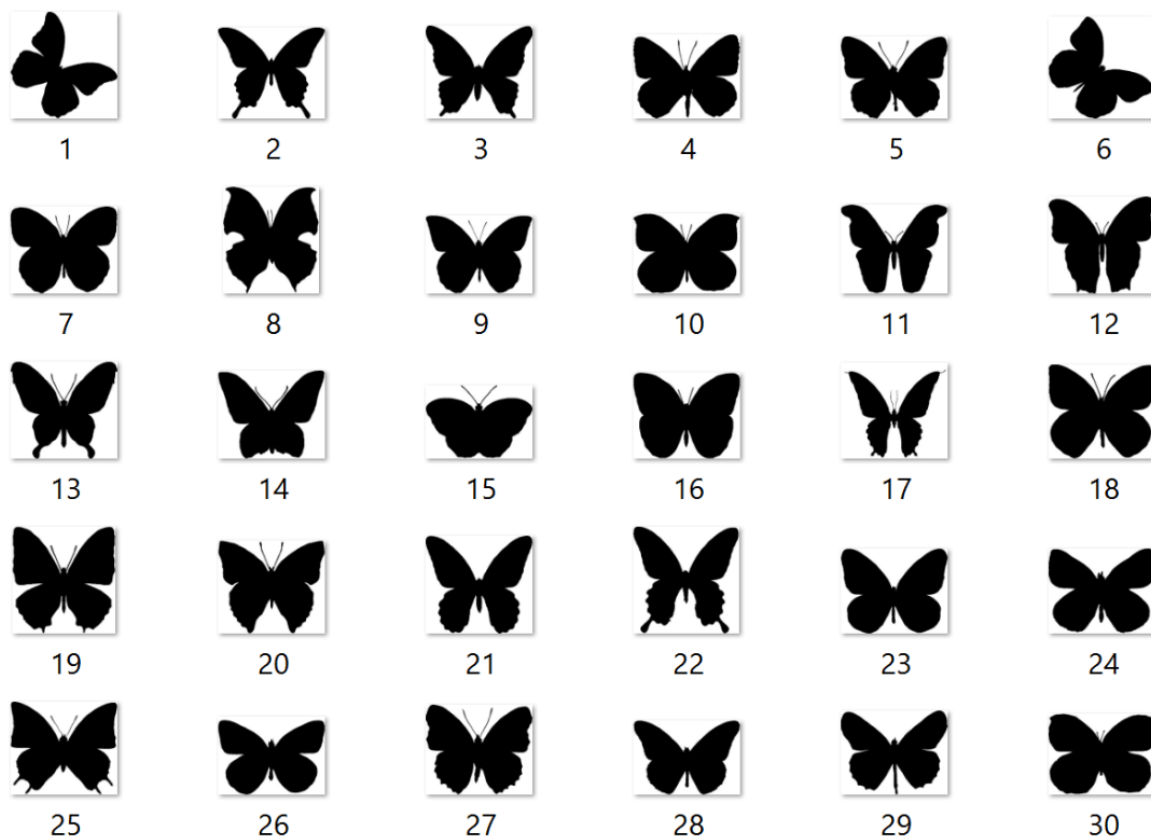


Рис. 3. Изображения из базы «Бабочки».

Для алгоритма на основе полного перебора эксперименты выполнены с использованием ресурсов суперкомпьютерного комплекса МГУ имени М.В. Ломоносова [6] как для последовательной, так и для параллельной реализации. В таблице 1 приведены результаты для 30 изображений базы бабочек, в частности, значение меры симметрии, время обработки каждого изображения, затраченное последовательным и параллельным алгоритмами, и достигнутое ускорение. Параллельный алгоритм был запущен на 64 процессорах. Реализация алгоритма на основе полного перебора такова, что доля $\alpha = 0,07\%$ от общего объёма вычислений может быть получена только последовательными расчётами, а, соответственно, доля $1 - \alpha = 99,93\%$ может быть распараллелена идеально. Тогда по закону Амдала (2) ускорение, которое может быть получено на вычислительной системе из 64 процессоров (по сравнению с однопроцессорным решением), не будет превышать величины:

$$S_p = \frac{1}{\alpha + \frac{1-\alpha}{p}}, \quad (2)$$

$$S_{64} = \frac{1}{0,0007 + \frac{0,9993}{64}} \approx 61,3. \quad (3)$$

В экспериментальных исследованиях среднее время работы параллельного алгоритма полного перебора для базы бабочек при вычислении на 64-х процессорах суперкомпьютера уменьшилось в 58 раз по сравнению с последовательным алгоритмом, что достаточно близко к теоретической оценке, рассчитанной по закону Амдала.

Для алгоритма уточнения оси симметрии, найденной скелетным методом, эксперименты проводились на персональном компьютере со следующими характеристиками: Intel® Core™ i7-2670QM CPU @ 2.2GHz, 16 Гб ОЗУ. В таблице 2 приведены результаты для 30 изображений базы бабочек: приведена эталонная мера симметрии, полученная алгоритмом на основе полного перебора, а также мера, полученная параллельной и последовательной версиями уточняющего алгоритма, усредненное время обработки каждого изображения и достигнутое ускорение. Параллельный алгоритм был запущен на 2 и 4 потоках. Для алгоритма уточнения заданы следующие параметры: $k_R = 0,03$, $\varepsilon = 0,25$, $n = 20$.

Используя закон Амдала (2), можно рассчитать величину предельного ускорения, которое может быть достигнуто параллельной реализацией уточняющего алгоритма с долей последовательных вычислений $\alpha = 15,3\%$ на 2 и 4 потоках соответственно:

$$S_2 = \frac{1}{0,153 + \frac{0,847}{2}} \approx 1,73, \quad (4)$$

$$S_4 = \frac{1}{0,153 + \frac{0,847}{4}} \approx 2,74. \quad (5)$$

Среднее время работы параллельного уточняющего алгоритма для базы бабочек при вычислении на 2-х потоках уменьшилось в 1,158 раз по сравнению с последовательным алгоритмом, что достаточно близко к теоретической оценке (4), рассчитанной по закону Амдала.

Таблица 1. Результаты работы последовательной и параллельной версии алгоритма поиска оси симметрии на основе полного перебора на суперкомпьютере

№ изобр.	Значение меры симметрии	Изображения из базы «Бабочки»		Ускорение при увеличении числа процессов, раз
		Число процессов		
		1	64	
		Время, сек	Время, сек	
1	0,974972	3469,73	59,68	58,14
2	0,940896	8640,51	147,61	58,53
3	0,925644	9121,3	156,62	58,24
4	0,873039	11743,7	208,46	56,34
5	0,852842	14633,6	240,91	60,74
6	0,91712	4696,73	81,48	57,64
7	0,90349	10410,6	182,03	57,19
8	0,90434	5456,33	92,16	59,21
9	0,921313	11460,4	189,07	60,61
10	0,847191	11529,1	204,06	56,50
11	0,935819	6504,34	110,16	59,04
12	0,931439	5741	104,44	54,97
13	0,93892	5922,68	99,52	59,51
14	0,956936	5745,89	102,85	55,87
15	0,962781	4847,06	84,80	57,16
16	0,936332	5781,57	96,05	60,20
17	0,938174	6590,04	116,24	56,69
18	0,85629	8570,97	141,06	60,76
19	0,903032	7802,34	129,74	60,14
20	0,928616	7021,68	117,79	59,61
21	0,94528	5541,84	97,51	56,83
22	0,935842	5181,28	89,52	57,88
23	0,917868	6739,13	118,61	56,82
24	0,895583	9412,85	166,89	56,40
25	0,925221	14363,6	235,49	61,00
26	0,900524	9320,4	158,42	58,83
27	0,881277	13511,5	225,11	60,02
28	0,940923	3217,98	56,82	56,63
29	0,921864	3554,41	62,60	56,78
30	0,794933	8561,1	146,89	58,28

Таблица 2. Результаты работы последовательной и параллельной версии алгоритма уточнения оси зеркальной симметрии на персональном компьютере

Изображения из базы «Бабочки»								
№ изображения	Полный перебор на суперкомпьютере с 64 процессорами			Алгоритм уточнения оси, выполняемый на персональном компьютере Intel® Core™ i7-2670QM CPU @ 2.2GHz, 16 Гб ОЗУ				
	Мера	Время, сек	Мера	Число потоков			Ускорение на 2-х потоках	Ускорение на 4-х потоках
				1	2	4		
				Время, сек	Время, сек	Время, сек		
1	0,974972	3469,73	0,974972	0,553	0,487	0,429	1,136	1,289
2	0,940896	8640,51	0,940896	0,652	0,552	0,553	1,180	1,178
3	0,925644	9121,3	0,925644	0,566	0,489	0,488	1,157	1,161
4	0,873039	11743,7	0,79014	0,522	0,472	0,448	1,107	1,166
5	0,852842	14633,6	0,852842	0,637	0,543	0,541	1,173	1,179
6	0,91712	4696,73	0,91712	0,674	0,550	0,533	1,225	1,263
7	0,90349	10410,6	0,90349	0,656	0,561	0,556	1,170	1,181
8	0,90434	5456,33	0,89266	0,584	0,498	0,437	1,173	1,337
9	0,921313	11460,4	0,921313	0,684	0,681	0,645	1,004	1,061
10	0,847191	11529,1	0,847191	0,685	0,644	0,632	1,065	1,083
11	0,935819	6504,34	0,935819	0,523	0,476	0,442	1,097	1,182
12	0,931439	5741	0,931439	0,538	0,493	0,448	1,093	1,201
13	0,93892	5922,68	0,850256	0,257	0,236	0,216	1,092	1,189
14	0,956936	5745,89	0,819866	0,383	0,342	0,316	1,121	1,215
15	0,962781	4847,06	0,962781	0,525	0,522	0,489	1,005	1,073
16	0,936332	5781,57	0,936332	0,507	0,491	0,450	1,032	1,128
17	0,938174	6590,04	0,938174	0,678	0,618	0,582	1,098	1,166
18	0,85629	8570,97	0,703983	0,459	0,411	0,399	1,118	1,150
19	0,903032	7802,34	0,903032	0,440	0,395	0,372	1,114	1,181
20	0,928616	7021,68	0,737619	0,378	0,340	0,334	1,110	1,131
21	0,94528	5541,84	0,94528	0,460	0,420	0,408	1,096	1,127
22	0,935842	5181,28	0,935842	0,437	0,391	0,367	1,120	1,193
23	0,917868	6739,13	0,749053	0,595	0,570	0,539	1,045	1,105
24	0,895583	9412,85	0,895583	0,643	0,679	0,637	0,948	1,009
25	0,925221	14363,6	0,925221	0,669	0,596	0,617	1,122	1,083
26	0,900524	9320,4	0,900524	0,703	0,651	0,653	1,080	1,076
27	0,881277	13511,5	0,880764	0,595	0,583	0,562	1,020	1,058
28	0,940923	3217,98	0,940923	0,491	0,437	0,401	1,124	1,224
29	0,921864	3554,41	0,921864	0,478	0,376	0,346	1,272	1,384
30	0,794933	8561,1	0,794714	0,665	0,610	0,621	1,089	1,070

5. Заключение

Предложенный параллельный алгоритм полного перебора позволяет находить эталонную ось симметрии за приемлемое время при наличии большой вычислительной мощности (такой как суперкомпьютер «Ломоносов»). Экспериментальные исследования на базах изображений «Бабочки» и Flavia показали, что алгоритм позволяет найти эталонную ось симметрии за время, приемлемое для обработки баз, состоящих из сотен и тысяч изображений, что сделало возможным его применение для автоматической разметки баз изображений. Полученные результаты могут рассматриваться как эталонные оси симметрии, которые необходимы для дальнейшей отладки приближенных методов поиска осей симметрии, а также других реализаций точного алгоритма.

Разработанная параллельная версия предложенного приближенного алгоритма уточнения обеспечивает возможность решения прикладных задач анализа изображений в условиях, близких к режиму реального времени, позволяя достичь времени обработки, исчисляемого в долях секунды, даже на обычных многоядерных персональных компьютерах,

сохраняя при этом максимальное, либо близкое к максимальному качество решения. Как видно из табл. 2, для 21-го изображения из 30-и быстрый алгоритм нашел ось симметрии, полностью совпадающую с эталонной (зеленая заливка в ячейках таблицы). Что касается оставшихся девяти изображений, то для трех из них (номера 8, 27 и 30) различие в мере симметрии составило менее 1%. Впрочем, из рис. 4 видно, что и для изображений под номерами 4, 13, 14, 18, 20 и 23 решение, найденное быстрым алгоритмом уточнения, лежит достаточно близко к эталонному. Тот факт, что приближенный алгоритм уточнения оси симметрии не смог найти эталонную ось, объясняется либо очень неудачным затравочным приближением, выполненным скелетным алгоритмом (изображения 4, 20, 23), либо очень сложной конфигурацией контура в области поиска симметрии (усики бабочки).

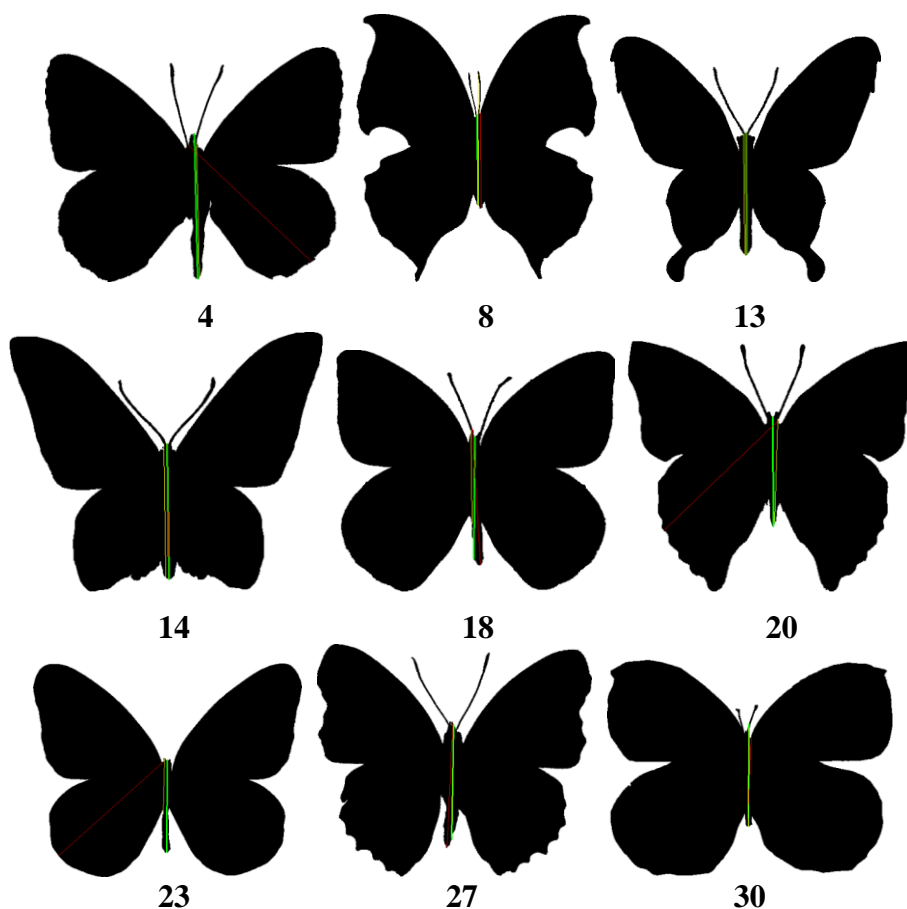


Рис. 4. Изображения из базы «Бабочки», для которых не удалось найти эталонную ось симметрии быстрым методом. Эталонная ось - зеленая, затравочная (построенная на основе скелета) - красная, уточненная – желтая.

Благодарности

Работа выполнена при поддержке гранта РФФИ 16-57-52042.

Авторы выражают благодарность ректору МГУ Садовничему Виктору Антоновичу, а также НИВЦ МГУ за предоставление возможности проведения вычислительных экспериментов на суперкомпьютерном комплексе МГУ имени М.В. Ломоносова.

Литература

- [1] van Otterloo, P. J. A Contour-Oriented Approach to Digital Shape Analysis // PhD thesis, Delft University of Technology, Delft, The Netherlands. – 1988.
- [2] Sheynin, S. Computation of Symmetry Measures for Polygonal Shapes / S. Sheynin, A. Tuzikov, D. Volgin // Computer Analysis of Images and Patterns. Springer Berlin Heidelberg. – 1999. – P. 183–190.
- [3] Yang, X., Symmetry of shapes via self-similarity / X. Yang, N. Adluru, L. J. Latecki, X. Bai, Z. Pizlo // Advances in Visual Computing, Springer Berlin Heidelberg. – 2008. – P. 561–570.
- [4] Кушнир, О.А. Определение зеркальной симметрии фигур на основе цепочек скелетных примитивов / О.А. Кушнир, О.С. Середин // Математические методы распознавания образов: Тезисы докладов 17-й Всероссийской конференции с международным участием, г. Светлогорск, 2015 г. – М.: Торус Пресс, 2015. – С.180–181.
- [5] Kushnir, O. Reflection Symmetry of Shapes Based on Skeleton Primitive Chains / O. Kushnir, S. Fedotova, O. Seredin, A. Karkishchenko // Fifth International Conference, AIST 2016, Yekaterinburg, Russia, April 7–9, 2016, Revised Selected Papers, CCIS, Springer International Publishing Switzerland. – 2016. – Vol. 661. – P. 281–292.
- [6] Воеводин, Вл.В. Практика суперкомпьютера «Ломоносов» / Вл.В. Воеводин, С.А. Жуматий, С.И. Соболев, А.С. Антонов, П.А. Брызгалов, Д.А. Никитенко, К.С. Стефанов, Вад.В. Воеводин // Открытые системы. – Москва: Издательский дом «Открытые системы», 2012. – № 7. – С. 36–39.

- [7] Федотова, С.А. Алгоритмы уточнения оси зеркальной симметрии, найденной методом сравнения подцепочек скелетных примитивов / С.А. Федотова, О.С. Середин, О.А. Кушнир. // Известия Тульского государственного университета. Технические науки. –Тула: Изд-во ТулГУ, 2016. – Вып.11. – Ч.1. – С.99–111.
- [8] Wu, S. G. A leaf recognition algorithm for plant classification using probabilistic neural network // 2007 IEEE international symposium on signal processing and information technology. – IEEE, 2007. – P. 11–16.
- [9] Michael, J. Q. Parallel Programming in C with MPI and OpenMP // Dubuque, IA: McGraw-Hill. – 2004.
- [10] Антонов, А.С. Параллельное программирование с использованием технологии MPI: Учебное пособие. – М.: Изд-во МГУ, 2004. – 71 с.
- [11] Антонов, А.С. Параллельное программирование с использованием технологии OpenMP: Учебное пособие.-М.: Изд-во МГУ, 2009. – 77 с.
- [12] Федотова, С.А. Оптимизация точного алгоритма определения зеркальной симметрии бинарных растровых изображений // Ломоносов – 2016: Материалы XXIII Международной научной конференции студентов, аспирантов и молодых ученых: секция «Вычислительная математика и кибернетика»; 11 – 15 апреля; Москва, МГУ им. М.В. Ломоносова, факультет ВМК: Сб. тезисов. – М.: Издательский отдел факультета ВМК МГУ, 2016. – С. 32–35.