

Оптимальное совмещение конечного множества точек с непрерывной поверхностью в трёхмерном пространстве

Александр Самойлов, Евгений Попов

Кафедра инженерной геометрии, компьютерной графики и автоматизированного проектирования
Нижегородский Государственный Архитектурно-Строительный Университет, Нижний Новгород, Россия
{greatlimit@rambler.ru, popov@sandy.ru}

Аннотация

Работа посвящена проблеме совмещения в трёхмерном пространстве двух множеств друг с другом, первое из которых представляет собой конечный набор точек (заданный в общем случае на нерегулярной сетке), а второе является фрагментом непрерывной поверхности.

Новый алгоритм решения данной проблемы основан на геометрических построениях и имеет преимущества по скорости и простоте программной реализации по сравнению с решением на основе методов оптимизации.

Алгоритм может найти эффективное применение в области бесконтактного определения геометрии крупногабаритных изделий, а так же для совмещения фрагментов цветных 2D изображений.

Ключевые слова: совмещение геометрических фигур, совмещение изображений, преобразование движения.

1. ВВЕДЕНИЕ

Постановка задачи. Дано два множества – непрерывная поверхность Σ в трёхмерном пространстве, заданная на компактной области $D \subset R^2$, и набор точек $P: P_i(x_i, y_i, z_i)$, $i = \overline{1, N}$. Требуется из всевозможных трёхмерных преобразований движения Ω найти такое преобразование Ω_{opt} , чтобы множество $\Omega_{opt}(P)$ являлось ближайшим к поверхности Σ согласно заданной функции расстояния ρ , то есть чтобы было выполнено условие минимума:

$$\sum_{i=1}^N \rho(\Sigma, \Omega_{opt}(P_i)) = \min_{\Omega} \sum_{i=1}^N \rho(\Sigma, \Omega(P_i)) \quad (1)$$

где $\rho(\Sigma, X)$ - функция расстояния от точки X до поверхности Σ . Преобразование движения в трёхмерном пространстве задается 6 параметрами: координатами вектора сдвига $\Delta_1, \Delta_2, \Delta_3$ и тремя углами поворота $\varphi_x, \varphi_y, \varphi_z$. Численное решение задачи методами оптимизации представляется затруднительным по следующим причинам:

- В общем случае между множествами трудно установить даже приблизительное соответствие, что не позволяет оценить интервалы изменения параметров $\Delta_1, \Delta_2, \Delta_3, \varphi_x, \varphi_y, \varphi_z$, вынуждая брать их максимальными, что в свою очередь существенно замедляет вычислительный процесс
- Поверхность может не быть односвязной, что увеличивает число ограничений в задаче оптимизации, тем самым усложняя её
- Поверхность не всегда задается аналитически (может вычисляться через интерполяционные формулы), поэтому её производная в общем случае не известна, что препятствует использованию эффективных численных алгоритмов, требующих знания производной

- Время вычисления функции расстояния зависит от N - числа точек множества P , что существенно замедляет вычислительный процесс в случае густого точечного множества

Сформулированная задача является актуальной, как минимум, по двум направлениям.

Во-первых, совмещение трёхмерных фигур – проблема, востребованная в контроле геометрии изделий на предприятиях, который предполагает сопоставление модели изготовленного изделия с моделью эталонного образца (первая, как правило, строится с помощью бесконтактных измерений, а вторая - на основе таблицы проектных величин эталонного образца). Для вычисления отклонений в конкретных точках изделия необходимо привязать обе модели к одной системе координат, для чего используется ряд опорных точек (базис), относительно которых проводятся бесконтактные измерения. Но в ряде случаев (деформируемые изделия; изделия, подвергающиеся транспортировке в разобранном виде и затем – сборке) задание базиса не имеет смысла. В подобных ситуациях нужен алгоритм совмещения двух подобных изделия в общей системе координат, не обладая информацией об изначальном базисе измерений.

Во-вторых, задача гипотетически может представлять более общую формулировку задачи о совмещении изображений, которая на данный момент является одной из фундаментальных задач компьютерного зрения. Если двумерному цветному изображению можно поставить в соответствие функцию двух переменных, значение которой выражает оттенок цвета в конкретной точке (x, y) , то по двум изображениям можно восстановить две поверхности и рассматривать меньшее по площади изображение как множество P . Задача о совмещении изображений является плохо обусловленной задачей [1]. Наиболее известные на сегодня методы её решения – корреляционные методы [6, 7] и методы, основанные на поиске инвариантов или особых точек [1, 2, 3].

2. РЕШЕНИЕ ДВУМЕРНОЙ ЗАДАЧИ

Основные положения разработанного подхода могут быть пояснены на примере двумерной задачи. В этом случае требуется найти двумерное преобразование движения, переводящее двумерный массив точек $P: P_i(x_i, y_i)$ в положение, соответствующее минимальному расстоянию от заданной непрерывной кривой l . Могут быть сформулированы следующие три допущения:

1. Множество P и фрагмент кривой l являются объектами с приблизительно схожей формой
2. Точки множества P идут друг за другом достаточно «густо», т. е. множество P формой похоже на непрерывную кривую
3. Множество P не имеет выбросов, либо выбросы были предварительно устранены.

В двумерном случае на кривой возможно выделить начальную и конечную точки и пронумеровать точки от первой до последней. Тогда приближённое решение может быть найдено путем совмещения кривой l с треугольником максимального периметра, построенного на точках множества P , что представляет собой существенно более простую задачу, т. к. число точек равно трем. Решения данной подзадачи методами оптимизации имеет те же недостатки, что и перечисленные выше для исходной задачи, за исключением того, что неизвестных параметров в данном случае три, и число точек множества P равно трём. Использовать его нет смысла еще и потому, что при обобщении результатов решения двумерной задачи на трехмерный случай задача поиска минимума вновь требует определения 6 параметров. В связи с этим, найдём соответствие между треугольником максимального периметра и кривой l путем геометрических построений (назовем этот подход «методом окружностей»). Для удобства обозначим $\rho(P_a, P_b) \stackrel{def}{=} r_{ab} \forall a, b = \overline{1, N}$ и далее везде будем использовать данное обозначение. Пусть кривая l параметризована $l: \{x = x(t), y = y(t)\}, t \in [0, 1]$. Организуем перебор по параметру t с шагом Δt (см. рис. 1), отмечая на фрагменте кривой l точку P_a^* (первая точка треугольника) и проводя окружность с центром в этой точке радиуса, равного r_{ab} .

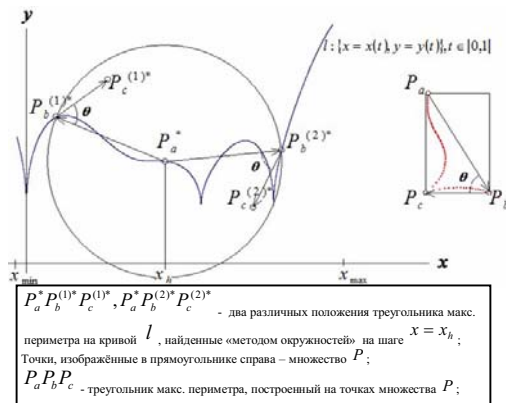


Рис 1. Поиск решений 1-го приближения двумерной задачи «методом окружностей»

Точку пересечения P_b^* (вторая точка треугольника) окружности с кривой l найдем методом бисекции, разбив окружность на дуги такого малого размера $\Delta l^{(1)}$, чтобы на каждой из дуг было не более одной точки пересечения. В общем случае, точек P_b^* может быть несколько. Для каждой из найденных точек пересечения P_b^* найдём точку P_c^* (третья точка треугольника), исходя из условия равенства треугольников $P_a^* P_b^* P_c^*$ и $P_a P_b P_c$. Если третья точка отстоит от кривой l на расстояние, не превышающее заданную величину δ_1 , ищем параметры двумерного преобразования движения и пополняем ими множество решений 1-го приближения $G^{(1)}$. Преобразование движения в двумерном пространстве однозначно определяется тремя параметрами, при условии, что задан центр поворота. Для повышения точности будем брать в качестве центра поворота точку пересечения медиан треугольника максимального периметра. Найдя множество $G^{(1)}$, организуем процедуру уточнения, которая будет основана на следующих положениях.

1) На каждом шаге итерационного процесса согласно заданному правилу помимо точек P_a, P_b, P_c в рассмотрение вводятся добавочные точки из множества P . Пусть Ω_k - множество точек, введённых в рассмотрение на k -м шаге приближения. Тогда $\Omega_1: \{P_a, P_b, P_c\}$. По определённому правилу каждому из двух отрезков $P_a P_b, P_b P_c$ ставится в соответствие по одной новой точке из множества P , и на 2-м шаге приближения рассматривается уже 4 отрезка: $P_a P_\mu, P_\mu P_b, P_b P_\nu, P_\nu P_c$. Далее эта процедура применяется ко всем отрезкам рекурсивно. То есть на k -м шаге приближения имеется $N_k = (2^k + 1)$ точек и 2^k отрезков, соединяющих эти точки. В качестве правила может быть выбрано условие: каждому двум точкам P_i, P_j ставится в соответствие треугольник наибольшего периметра среди всех треугольников $P_i P_s P_j$, где $i < s < j$.

2) Для каждого из множества решений k -го приближения $G^{(k)}$ выполняется варьирование (см. рис. 2) каждой из трёх точек-образов P_a^*, P_b^*, P_c^* в своих δ_k -окрестностях. Вариации P_{am}^* первой точки определяются сдвигом первой точки P_a^* с шагами $\Delta\varphi$ и $(\Delta r)_k = \delta_k / N_{div}$ ($\Delta\varphi, N_{div}$ - заданы) соответственно по углу φ и радиус-вектору r в полярных координатах ($0 \leq \varphi < 2\pi, 0 \leq r \leq \delta_k$). Вариации P_{bm}^* второй точки определяются как точки, взятые на дуге, полученной пересечением δ_k -окрестности точки P_b^* и окружности радиуса r_{ab} с центром в P_a^* . Итерирование по дуге происходит по параметру $\kappa = \overline{0, L-1}$, где $L = \Delta l / \Delta l_{окр}^{(2)}$ - число точек, взятых на дуге через заданное расстояние $\Delta l_{окр}^{(2)}$, Δl - длина дуги. Вариации P_{cm}^* третьей точки определяются из условия равенства треугольников $P_a^* P_b^* P_c^*$ и $P_{am}^* P_{bm}^* P_{cm}^*$.

3) Для каждой из вариаций $P_{am}^*, P_{bm}^*, P_{cm}^*$ вычисляются параметры двумерного преобразования движения, а затем - положения, в которые отобразятся этим преобразованием остальные точки из множества Ω_k . Если образы всех точек из Ω_k удовлетворяют условию

$$\forall s = \overline{1, N_k} : \rho(l, P_{sm}^*) < \delta_k \quad (2),$$

тогда преобразование запоминается в список элементов множества $G^{(k+1)}$. Это шаг итерационного процесса.

4) Выбор критериев останова. Процесс должен остановиться, если выполнено одно из условий:

- Достигнута заданная точность δ_∞ , т. е. условие (2) выполнено при $\delta_k = \delta_\infty$.
- Дальнейшие итерации не приводят к уменьшению числа элементов множества $G^{(k)}$.
- Выполнено более N_{fin} итераций (N_{fin} - задано)

Формулы для вычисления координат трёх варьируемых точек приведены в [5]. Таким образом, имеем всего 5 параметров, влияющих на точность и скорость алгоритма: $\Delta t, \Delta\varphi, N_{div}, \Delta l_{окр}^{(1)}, \Delta l_{окр}^{(2)}$. Описанный выше алгоритм был реализован в программном модуле «2DAnalyzer» [5] и протестирован на кривых различного вида.

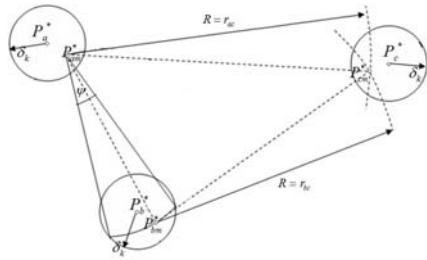


Рис 2: Процедура уточнения для двумерной задачи.

Тест №1 (точное соответствие между множествами нет). Кривая l задана уравнением $f(x) = x \cos(0.04x)$ на отрезке $[-50;30]$. Значения параметров: $\delta_1 = 3$, $\Delta t = 0.00625$, $\Delta \varphi = 0.2\pi$, $N_{div} = 10$, $\Delta l_{окр}^{(1,2)} = 0.001$, $n_{fin} = 3$. В качестве множества P были взяты точки кривой $g(x) = 0.0005x^3$ на отрезке $[-20;40]$ через расстояние 0.9, сдвинутые затем на вектор $\{100;50\}$ и повернутые на угол $-\frac{\pi}{3}$ относительно начала координат.

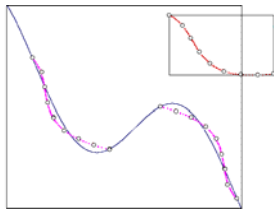


Рис 3: Совмещение кривых $f(x) = x \cos(0.04x)$ и $g(x) = 0.0005x^3$ с помощью модуля «2DAnalyzer»

На рис. 3 изображена последняя (третья) итерация алгоритма. В большем по размеру прямоугольнике изображена кривая l и точечное множество, совмещённое с ней. В меньшем по размеру прямоугольнике изображено множество P . Малыми кругами отмечены точки множества Ω_3 . Время работы программы – около 5 с.

Тест №2 (точное соответствие между множествами) Исходные данные для теста такие же, как для теста №1, кроме того, что: а) Кривая l задана уравнением $f(x) = x \ln(2.5 + \sin(0.1x))$ на отрезке $[-30;0]$; б) $\delta_1 = 0.2$, в) $g(x) = f(x)$. На рис. 4 изображена последняя (третья) итерация алгоритма. Время работы программы – около 5 с.

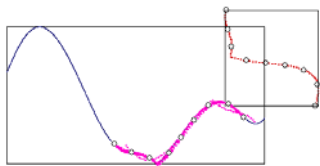


Рис 4: Совмещение двух фрагментов кривой $f(x) = x \ln(2.5 + \sin(0.1x))$ с помощью модуля «2DAnalyzer»

Эти и другие (не приведённые здесь) тесты показали, что алгоритм для двумерного случая работает эффективно.

3. РЕШЕНИЕ ТРЕХМЕРНОЙ ЗАДАЧИ

Попытаемся обобщить алгоритм на трехмерный случай. Обобщению будут подвергнуты принцип треугольника максимального периметра и процедура уточнения. Три допущения, сделанные для двумерной задачи, остаются в силе. Пусть $P_a P_b P_c P_d$ - тетраэдр с максимальной суммой

длин сторон, $P_a^* P_b^* P_c^* P_d^*$ - образ тетраэдра $P_a P_b P_c P_d$, соответствующий одному из искомым положений последнего на поверхности Σ . Пусть поверхность Σ задана параметрически в ограниченной двумерной компактной области D , то есть $\Sigma : \{x = x(u,v), y = y(u,v), z = z(u,v)\}; (u,v) \in [0,1] \times [0,1], (x(u,v), y(u,v)) \in D \forall (u,v) \in [0,1] \times [0,1]$. По аналогии с двумерной задачей, организуем перебор по каждому из параметров u, v с заданными шагами $\Delta u, \Delta v$ соответственно. Из каждой итерированной точки $(x(u,v), y(u,v), z(u,v))$ (назовем ее **первой точкой** P_a^* тетраэдра) на поверхности Σ очертим сферу радиуса r_{ab} . r_{ab} -сфера имеет в общем случае несколько линий пересечения с поверхностью Σ , которые могут быть как замкнутые, так и разомкнутые (в случае, если r_{ab} -сфера пересекает поверхность Σ на её границе, либо если поверхность Σ не односвязна). Поставим задачу определить число Λ линий пересечения и отметить на каждой из линий пересечения точки через приблизительно равные расстояния, измеренные вдоль линии (далее будем называть линии пересечения r_{ab} -сферы с поверхностью Σ l -линиями). Введём для этого понятие «матрица сферы». В сферических координатах разделим r_{ab} -сферу на параллели и меридианы с заданными шагами $\Delta \zeta, \Delta \eta$ по горизонтали и по вертикали соответственно:

$$\zeta \in [0, 2\pi], \eta \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right], m = \left\lceil \frac{2\pi}{\Delta \zeta} \right\rceil + 1, n = \left\lceil \frac{\pi}{\Delta \eta} \right\rceil + 1 \quad (3)$$

r_{ab} -сфера разделится на nm участков, при этом каждому i, j -участку ($i = \overline{0, n-1}; j = \overline{0, m-1}$) будут поставлены в соответствие значения

$$\zeta_j = j\Delta \zeta, \eta_i = -\frac{\pi}{2} + i\Delta \eta, i = \overline{0, n-1}; j = \overline{0, m-1} \quad (4)$$

Тогда определим матрицу сферы для r_{ab} -сферы как матрицу $n \times m$, значение i, j -элемента которой равно 1, если через i, j -участок сферы проходит хотя бы одна из l -линий, и равно 0, если ни одна из l -линий не проходит через i, j -участок. Для того, чтобы определить, проходит ли одна из l -линий через конкретный i, j -участок, достаточно определить, пересекает ли один из 4-х (или 3-х) криволинейных отрезков i, j -участка поверхность Σ . Будем полагать, что размеры $\Delta \zeta, \Delta \eta$ достаточно малы для того, чтобы судить о пересечении/не пересечении криволинейным отрезком поверхности Σ через условие расположения концов этого отрезка по разные/по одну сторону от поверхности Σ соответственно. Таким образом, матрица сферы может быть вычислена.

3.1 Поиск второй точки тетраэдра (метод «мини-сфер»)

Далее матрица сферы будет использована для поиска l -линий. Будем в цикле итерировать по i, j -участкам, в которых замечено пересечение r_{ab} -сферы с поверхностью Σ (они соответствуют элементам матрицы со значением 1). Критерий выхода из цикла – все элементы матрицы сферы стали равны 0. На каждом шаге цикла будем пытаться выстроить ряд точек на l -линии, проходящей через текущий i, j -участок, считая в качестве первой точки λ_1^i на l -линии ту точку, в которой одно из рёбер i, j -участка пересекает поверхность Σ (точка может быть найдена

методом бисекции). Зная первую точку λ_1^l на l -линии и матрицу сферы, выполним «следование» по l -линии методом «мини-сфер», который заключается в следующем. Очертим сферу достаточно малого радиуса τ с центром в λ_1^l и определим все точки пересечения τ -сферы с l -линией. Поскольку l -линия есть результат пересечения поверхности Σ с r_{ab} -сферой, то искомые точки являются результатом пересечения трёх объектов: r_{ab} -сферы, поверхности Σ и τ -сферы. А т. к. пересечение есть операция коммутативная, то будет выгоднее сначала найти пересечение двух сфер: r_{ab} -сферы и τ -сферы (которое есть окружность), а затем найти искомые точки $\xi_{2,s}^l$ как пересечение поверхности Σ окружностью. Такой подход выгоден, так как окружность можно параметризовать и использовать для поиска пересечения метод бисекции (как и в случае двумерной задачи), разбив окружность на дуги достаточно малого заданного размера $\Delta_{окп}^{(1)}$. После того, как найдены точки пересечения $\xi_{2,s}^l$ τ -сферы с l -линией, выберем среди них ту, которая максимально отстоит от

$$\text{точки } \lambda_1^l, \text{ т. е. } \rho(\lambda_2^l, \lambda_1^l) = \max_s \rho(\xi_{2,s}^l, \lambda_1^l) \quad (5)$$

Так определяется вторая точка λ_2^l . На k -м шаге «следования» из точки λ_k^l вновь проводим τ -сферу и ищем её точки пересечения с l -линией. Но в отличие от первого шага «следования», среди найденных точек пересечения $\xi_{k,s}^l$ выберем одну, которая дальше всех отстоит от λ_{k-1}^l , найденной на $(k-1)$ -м шаге «следования», или строго:

$$\rho(\lambda_k^l, \lambda_{k-2}^l) = \max_s \rho(\xi_{k,s}^l, \lambda_{k-2}^l), k \geq 3 \quad (6)$$

На каждом шаге «следования» вычисляется, в какой i, j -участок r_{ab} -сферы попадает точка λ_k^l , и значение соответствующего i, j -элемента матрицы сферы устанавливается равным 0. Это значит, что данный i, j -участок r_{ab} -сферы уже был затронут одной из l -линий, проходящих через него, и на следующем шаге внешнего цикла (по i, j -участкам) данный i, j -участок уже не будет рассматриваться как возможный для того, чтобы выстраивать из него l -линию методом мини-сфер.

«Следование» по l -линии должно остановиться, если либо а) на k -м шаге «следования» не было найдено ни одной точки пересечения $\xi_{k,s}^l$, либо б) на k -м шаге «следования» обнаружилось, что точки обхода начали повторяться (l -линия уже пройдена целиком, и мы вернулись в её начало):

$$\rho(\lambda_k^l, \lambda_1^l) < \tau \quad (7)$$

Таким образом l -линия проходит в положительном направлении. Таким же способом l -линия должна быть пройдена и в отрицательном направлении от точки λ_1^l (это имеет смысл для не односвязных поверхностей). Каждая из точек λ_k^l рассматривается как одна из возможных **вторых точек** P_b^* тетраэдра. Метод мини-сфер учитывает случаи, когда несколько l -линий пересекаются или касаются друг друга в одной точке (несмотря на то, что из одного i, j -участка может быть выстроена лишь одна l -линия, другие l -линии, пересекающиеся с ней или касающиеся её в этом же i, j -участке будут выстроены «следованием» из других i, j -участков, т. к. элементы матрицы сферы для смежных l -линий не «зануляются» при обходе данной l -линии).

3.2 Поиск третьей точки тетраэдра

Проведём в треугольнике $P_a P_b P_c$ высоту $P_c H$. Для поиска образа H^* точки H можно воспользоваться пропорцией:

$$\frac{P_a^* H^*}{P_a^* P_b^*} = \frac{P_a H}{P_a P_b} \Rightarrow P_a^* H^* = \frac{|P_a^* P_b^*| |P_a H|}{|P_a P_b|^2} P_a^* P_b^* = \frac{|P_a H|}{r_{ab}} P_a^* P_b^* \quad (8)$$

Проведём в точке H^* плоскость, перпендикулярную отрезку $P_a^* P_b^*$, а в этой плоскости очертим окружность с центром в точке H^* радиуса $|P_c H| = h$. Тогда точка P_c^* может быть получена как пересечение этой окружности с поверхностью Σ . Для поиска точек пересечения вновь используем метод бисекции, разбив окружность на дуги достаточно малого заданного размера $\Delta_{окп}^{(2)}$. Координаты точек, разделяющих окружность на равные дуги, могут быть вычислены по формулам:

$$\begin{aligned} x_{i0} &= x_a^* + (x_b^* - x_a^*) \mu + h \frac{(-z_b^* - z_a^*)(x_b^* - x_a^*) \cos \theta_i - r_{ab}(y_b^* - y_a^*) \sin \theta_i}{r_{ab} \sqrt{(x_b^* - x_a^*)^2 + (y_b^* - y_a^*)^2}} \quad (9), \\ y_{i0} &= y_a^* + (y_b^* - y_a^*) \mu + h \frac{(-z_b^* - z_a^*)(y_b^* - y_a^*) \cos \theta_i + r_{ab}(x_b^* - x_a^*) \sin \theta_i}{r_{ab} \sqrt{(x_b^* - x_a^*)^2 + (y_b^* - y_a^*)^2}} \\ z_{i0} &= z_a^* + (z_b^* - z_a^*) \mu + h \frac{\sqrt{(x_b^* - x_a^*)^2 + (y_b^* - y_a^*)^2}}{r_{ab}} \cos \theta_i \end{aligned}$$

где
$$\mu = \frac{1}{2} \left((r_{ac}^2 - r_{bc}^2) / r_{ab}^2 + 1 \right) \theta_i = \frac{i \Delta_{окп}^{(2)}}{h}, i = 0, \left(\frac{2\pi h}{\Delta_{окп}^{(2)}} \right) - 1 \quad (10)$$

Каждая из найденных точек пересечения окружности с поверхностью Σ рассматривается как одна из возможных **третьих точек** P_c^* тетраэдра.

3.3 Поиск четвёртой точки тетраэдра

Четвёртая точка тетраэдра может быть восстановлена по высоте тетраэдра, опущенной из точки P_d^* , поскольку тетраэдры $P_a P_b P_c P_d$ и $P_a^* P_b^* P_c^* P_d^*$ равны. Опуская промежуточные выкладки, приведём лишь результат:

$$\begin{cases} x_d^* = x_a^* + \chi_{ab}(x_b^* - x_a^*) + \chi_{ac}(x_c^* - x_a^*) \pm A^1 |p| \\ y_d^* = y_a^* + \chi_{ab}(y_b^* - y_a^*) + \chi_{ac}(y_c^* - y_a^*) \pm B^1 |p| \\ z_d^* = z_a^* + \chi_{ab}(z_b^* - z_a^*) + \chi_{ac}(z_c^* - z_a^*) \pm C^1 |p| \end{cases} \quad (11),$$

где
$$p = -\frac{1}{A^2 + B^2 + C^2} \begin{vmatrix} x_a^* & y_a^* & z_a^* & 1 \\ x_b^* & y_b^* & z_b^* & 1 \\ x_c^* & y_c^* & z_c^* & 1 \\ x_d^* & y_d^* & z_d^* & 1 \end{vmatrix} \quad (12),$$

$$A = \begin{vmatrix} y_a^* & z_a^* & 1 \\ y_b^* & z_b^* & 1 \\ y_c^* & z_c^* & 1 \end{vmatrix} \quad B = \begin{vmatrix} z_a^* & x_a^* & 1 \\ z_b^* & x_b^* & 1 \\ z_c^* & x_c^* & 1 \end{vmatrix} \quad C = \begin{vmatrix} x_a^* & y_a^* & 1 \\ x_b^* & y_b^* & 1 \\ x_c^* & y_c^* & 1 \end{vmatrix} \quad (13),$$

$$A^1 = \begin{vmatrix} y_a^* & z_a^* & 1 \\ y_b^* & z_b^* & 1 \\ y_c^* & z_c^* & 1 \end{vmatrix} \quad B^1 = \begin{vmatrix} z_a^* & x_a^* & 1 \\ z_b^* & x_b^* & 1 \\ z_c^* & x_c^* & 1 \end{vmatrix} \quad C^1 = \begin{vmatrix} x_a^* & y_a^* & 1 \\ x_b^* & y_b^* & 1 \\ x_c^* & y_c^* & 1 \end{vmatrix} \quad (14),$$

а значения χ_{ab}, χ_{ac} могут быть найдены из системы:

$$\begin{cases} \chi_{ab}(x_b - x_a) + \chi_{ac}(x_c - x_a) = x_d - x_a - Ap \\ \chi_{ab}(y_b - y_a) + \chi_{ac}(y_c - y_a) = y_d - y_a - Bp \\ \chi_{ab}(z_b - z_a) + \chi_{ac}(z_c - z_a) = z_d - z_a - Cp \end{cases} \quad (15)$$

(следует взять такие два из уравнений системы 15, чтобы выбранная подсистема имела наибольший определитель). Для определения знака в (11) следует воспользоваться

критерием:
$$\text{sign} \begin{pmatrix} x_a^* & y_a^* & z_a^* & 1 \\ x_b^* & y_b^* & z_b^* & 1 \\ x_c^* & y_c^* & z_c^* & 1 \\ x_d^* & y_d^* & z_d^* & 1 \end{pmatrix} = \text{sign} \begin{pmatrix} x_a & y_a & z_a & 1 \\ x_b & y_b & z_b & 1 \\ x_c & y_c & z_c & 1 \\ x_d & y_d & z_d & 1 \end{pmatrix} \quad (16)$$

Далее найденная точка (x_d^*, y_d^*, z_d^*) также должна быть проверена на близость к поверхности Σ . Если выполнено условие $\rho(\Sigma, (x_d^*, y_d^*, z_d^*)) < \delta_k$ (17),

тогда считаем, что (x_d^*, y_d^*, z_d^*) - четвёртая точка P_i^* тетраэдра, ищем преобразование движения, переводящее тетраэдр $P_a^*P_b^*P_c^*P_d^*$ в $P_a^*P_b^*P_c^*P_d^*$ [4], и заносим параметры преобразования в память. После проверки условия близости, независимо от результата проверки переходим к рассмотрению следующего кандидата на **третью точку**.

Таким образом, имеем всего 13 параметров, влияющих на точность и скорость алгоритма - $\Delta u, \Delta v, \Delta \zeta, \Delta \eta, \tau, \Delta I_{окр}^{(1)}, \Delta I_{окр}^{(2)}, N_{div}, \Delta \phi, \Delta \psi, L_1, L_2, L_3$, где $N_{div}, \Delta \phi, \Delta \psi, L_1, L_2, L_3$ - параметры, используемые в процедуре уточнения. Процедура уточнения для трехмерного случая отличается от последней для двумерного случая тем, что окрестности точек являются не кругами, а сферами. Соотношения для координат варьируемых точек будут выглядеть следующим образом:

1-я варьируемая точка тетраэдра $P_{am}^* (x_{am}^*, y_{am}^*, z_{am}^*)$

$$x_{am}^* = x_a^* + r \cos \psi \cos \varphi; y_{am}^* = y_a^* + r \cos \psi \sin \varphi; z_{am}^* = z_1^* + r \sin \psi \quad (18)$$

2-я варьируемая точка тетраэдра $P_{bm}^* (x_{bm}^*, y_{bm}^*, z_{bm}^*)$

$$\begin{cases} x_{bm}^* = x_{am}^* + \frac{\Delta x}{d} r_{ab} \cos\left(\frac{i\alpha}{L_1}\right) + \frac{\sin\left(\frac{i\alpha}{L_1}\right)}{\sqrt{\Delta x^2 + \Delta y^2}} \left(-\Delta x \Delta z \cos\frac{2\pi j}{L_2} + R \Delta y \sin\frac{2\pi j}{L_2} \right) \\ y_{bm}^* = y_{am}^* + \frac{\Delta y}{d} r_{ab} \cos\left(\frac{i\alpha}{L_1}\right) + \frac{\sin\left(\frac{i\alpha}{L_1}\right)}{\sqrt{\Delta x^2 + \Delta y^2}} \left(-\Delta y \Delta z \cos\frac{2\pi j}{L_2} - R \Delta x \sin\frac{2\pi j}{L_2} \right) \\ z_{bm}^* = z_{am}^* + \frac{\Delta z}{d} r_{ab} \cos\left(\frac{i\alpha}{L_1}\right) + \sin\left(\frac{i\alpha}{L_1}\right) \sqrt{\Delta x^2 + \Delta y^2} \cos\frac{2\pi j}{L_2}; i = 0, L_1, j = 0, L_2 - 1 \end{cases} \quad (19)$$

где $d = \sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2}; \Delta x = x_b^* - x_{am}^*; \Delta y = y_b^* - y_{am}^*; \Delta z = z_b^* - z_{am}^*$ (20),

$$\cos \alpha = \frac{1}{2} \left(\frac{d^2 + r_{ab}^2 - \delta_k^2}{r_{ab} d} \right)^2 - 1 \quad (21)$$

3-я варьируемая точка тетраэдра $P_{cm}^* (x_{cm}^*, y_{cm}^*, z_{cm}^*)$

$$\begin{cases} x_{cm}^* = \Delta x + h \left(-\frac{(z_{bm}^* - z_{am}^*)}{r_{ab}} (x_{bm}^* - x_{am}^*) \cos \chi_i - r_{ab} (y_{bm}^* - y_{am}^*) \sin \chi_i \right) \\ y_{cm}^* = \Delta y + h \left(-\frac{(z_{bm}^* - z_{am}^*)}{r_{ab}} (y_{bm}^* - y_{am}^*) \cos \chi_i + r_{ab} (x_{bm}^* - x_{am}^*) \sin \chi_i \right) \\ z_{cm}^* = \Delta z + h \frac{\sqrt{(x_{bm}^* - x_{am}^*)^2 + (y_{bm}^* - y_{am}^*)^2}}{r_{ab}} \cos \chi_i \end{cases} \quad (22)$$

где

$$\Delta x = x_{am}^* + \mu (x_{bm}^* - x_{am}^*); \Delta y = y_{am}^* + \mu (y_{bm}^* - y_{am}^*); \Delta z = z_{am}^* + \mu (z_{bm}^* - z_{am}^*) \quad (23),$$

$$\chi_i = -\arccos \frac{r_{ab} \Delta z - q (z_{bm}^* - z_{am}^*)}{r_{ab} \sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2 - q^2} \sqrt{(x_{bm}^* - x_{am}^*)^2 + (y_{bm}^* - y_{am}^*)^2}} + \left(\frac{2i}{L_3} - 1 \right) \arccos \frac{h^2 - \delta_k^2 + \sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2}}{2h \sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2 - q^2}}, i = 0, L_3 \quad (24)$$

$$q = (\Delta x (x_{bm}^* - x_{am}^*) + \Delta y (y_{bm}^* - y_{am}^*) + \Delta z (z_{bm}^* - z_{am}^*)) / r_{ab} \quad (25)$$

а μ определяется из (10).

Для выражения координат четвёртой точки следует использовать формулы, приведённые в разделе 3.3, при условии, что к нижнему индексу всех координат добавлена буква "m" (признак варьируемой точки).

Алгоритм для трехмерной задачи был реализован в программном модуле «3DAnalyzer». Ниже приведены два примера тестирования модуля на поверхностях.

Тест №3 (точного соответствия между множествами нет). Поверхность Σ задана уравнением $f(x) = 50 \sin(0.1 \sqrt{x^2 + y^2})$ на отрезке $[-60; 60]$. В качестве множества P были взяты точки поверхности $g(x) = 0.015(x^2 + y^2) - 150$ на отрезке $[-30; 30]$ через расстояние 0.9, сдвинутые затем на вектор $\{100; 320; 200\}$ и повернутые на углы $(\pi/2 - 0.2; -\pi/4; \pi/5)$ относительно начала координат. Параметры алгоритма: $\Delta u = 1/12, \Delta v = 1/12, \Delta \zeta = \Delta \eta = 0.1, \tau = 2, \Delta I_{окр}^{(1)} = \Delta I_{окр}^{(2)} = 0.04\pi h$ (такая связь задавалась внутри программы), $N_{div} = 3, \Delta \phi = \Delta \psi = \delta_k / (r N_{div})$ (такая связь задавалась внутри программы), $L_1 = L_2 = L_3 = 10$. Время работы программы - около 2 мин.

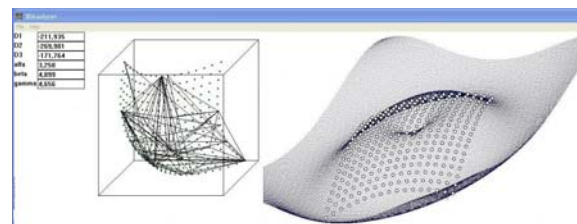


Рис 5: Совмещение поверхностей $f(x) = 50 \sin(0.1 \sqrt{x^2 + y^2})$ и $g(x) = 0.015(x^2 + y^2) - 150$ с помощью модуля «3DAnalyzer»

Слева на рис.5 (в кубе) изображены точки множества P . Справа изображена поверхность Σ и точки множества P , совмещённые с ней (обозначенные белым цветом).

3.4 Параметры алгоритма

данный момент вопрос о влиянии параметров $\Delta u, \Delta v, \Delta \zeta, \Delta \eta, \tau, \Delta I_{окр}^{(1)}, \Delta I_{окр}^{(2)}, N_{div}, \Delta \phi, \Delta \psi, L_1, L_2, L_3$ на эффективность алгоритма мало исследован. Но экспериментальным путём были установлены некоторые более предпочтительные соотношения между параметрами. 1) $\Delta u = \Delta v \approx 0.05$. Меньшие значения существенно снижают скорость алгоритма, но не сильно повышают точность. Тем более, что в ходе процедуры уточнения решения всё равно уточняются. 2) $\Delta \zeta, \Delta \eta$ следует выбирать такими, чтобы в пределах каждого i, j -участка r_{ab} -сферы поверхность имела малые перепады кривизны. 3) $\tau, \Delta I_{окр}^{(1)}, \Delta I_{окр}^{(2)}$ следует, как и два предыдущих параметра, выбирать, исходя из кривизны: между любыми двумя точками поверхности, отстоящими на $\tau, \Delta I_{окр}^{(1)}, \Delta I_{окр}^{(2)}$, кривизна не должна иметь сильных скачков. Излишнее уменьшение параметров $\Delta \zeta, \Delta \eta, \tau, \Delta I_{окр}^{(1)}, \Delta I_{окр}^{(2)}$ существенно снижает скорость алгоритма, но не сильно повышают точность. 4) N_{div} сильно влияет на скорость. Достаточно брать равным 3. То же относится и к L_1, L_2, L_3 - достаточно взять их равными 10. 5) $\Delta \phi, \Delta \psi$ лучше всего брать не константами, а функциями: $\Delta \phi = \Delta \psi = \delta_k / (r N_{div}), r \neq 0$ (при $r = 0 \Delta \phi = \Delta \psi = 0$). При таком выборе δ_k -окрестность «заполняется» точками более равномерно.

4. СОВМЕЩЕНИЕ ИЗОБРАЖЕНИЙ

В код модуля «3DAnalyzer» были добавлены функции для чтения/записи файлов формата «bmp» с преобразованием данных файла в поверхность и обратно (поверхность строится по регулярной сетке, а для получения координаты z значение, читаемое из файла, делится на нормирующий коэффициент ω). После совмещения полученных поверхностей в результирующий файл записывались обе поверхности, а на их границе – точки чёрного цвета (чтобы совмещённый фрагмент было проще увидеть).

Тест №4 (Поиск фрагмента изображения на нём же самом)

На рис. 6(в) приведён результат совмещения фрагмента изображения с ним же. Размеры изображения: 473x800, размеры фрагмента: 326x300. Нормирующий коэффициент $\omega = 100000$ (был подобран эмпирически). Параметры алгоритма: $\Delta u = 0.0528$, $\Delta v = 0.0313$, $\Delta \zeta = \Delta \eta = 0.1$, $\tau = 10$, $\Delta l_{\text{оп}}^{(1)} = \Delta l_{\text{оп}}^{(2)} = 0.04\pi h$, $N_{\text{div}} = 3$, $\Delta \varphi = \Delta \psi = \delta_k / (r N_{\text{div}})$, $L_1 = L_2 = L_3 = 10$.
Время работы программы – около 2 мин 25 с.



Рис 6 (а, б): Исходные данные теста №4 – изображение (слева) и его фрагмент (справа) **Рис 6 (в):** Результат теста №4 – совмещённое изображение

Тест №5 (Совмещение двух разных изображений)

На рис. 7(в) приведён результат совмещения двух изображений, представляющими вид на один объект с разных ракурсов. Размеры первого изображения: 473x800, размеры второго изображения: 397x670. Параметры алгоритма такие же, как для предыдущего теста. Время работы программы – около 2 мин 45 с.

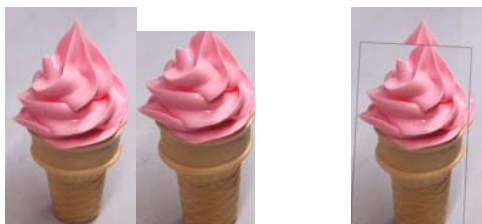


Рис 7 (а, б): Исходные данные теста №5 – два разных изображения **Рис 7 (в):** Результат теста №5 – совмещённое изображение

5. ЗАКЛЮЧЕНИЕ

Разработанный алгоритм показал удовлетворительные результаты как в применении для совмещения 3D геометрических фигур, так и для совмещения 2D изображений. Учитывая тот факт, что на данный момент разработка модуля «3DAnalyzer» не завершена, сделать окончательные выводы об эффективности его работы в плане скорости и точности не представляется возможным. Также остаётся не исследованным вопрос об оптимальных значениях параметров алгоритма в зависимости от типа поверхности или изображения (в случае с изображением добавляется ещё один параметр – нормирующий

коэффициент ω). В качестве недостатков алгоритма можно отметить следующие. Во-первых, на текущей стадии разработки алгоритм не учитывает преобразования масштабирования и симметрии. Во-вторых, алгоритм требует предварительной фильтрации шумов.

Между тем, алгоритм обладает серьёзными достоинствами, среди которых следует отметить, что во-первых, алгоритм способен работать на неодносвязных поверхностях (пример: тор) и поверхностях с самопересечением, во-вторых, рассматривает существование нескольких решений (пример: рис. 3). Задачи, которые предстоит решить в будущем в рамках затронутой проблемы – провести полное исследование влияния параметров алгоритма на скорость и точность, построив соответствующие графики, а также усовершенствовать алгоритм, устранив перечисленные выше недостатки.

6. ССЫЛКИ

- [1] Попов Ивашкин, Д. И. **Математические методы и комплекс программ частотно-пространственного анализа стереопар изображений** / Д. И. Ивашкин // Диссертация на соискание степени кандидата технических наук
- [2] Самарина, О. В. **Использование инвариантов при поиске соответствия изображений** / О. В. Самарина // Вестник Югорского государственного университета. - Ханты-Мансийск, 2008. выпуск 1(8). – С. 110-113.
- [3] Самарина, О. В. **Программный комплекс обработки одноканальных и трёхканальных изображений** / О. В. Самарина // Вестник ЮГУ.- Ханты-Мансийск, 2008. выпуск 3(10). – С. 67-71.
- [4] Самойлов, А. А. **Определение параметров трёхмерного преобразования движения по 4-м заданным точкам и их образам**/ А. А. Самойлов // Сборник трудов аспирантов, магистрантов и соискателей. Технические науки. – Н. Новгород: ННГАСУ, 2012. ISBN 978-5-87941-663-3
- [5] Самойлов, А. А. **Поиск оптимального совмещения кривой и точечного множества в двумерном пространстве**/ Е. В. Попов, С. И. Ротков, А. А. Самойлов // Приволжский научный журнал №4(20). Периодическое научное издание. Н. Новгород, ННГАСУ, 2011. ISSN 1995-2511
- [6] M. Thomas, Irene. **A Cross-Correlation Method for Merging Electron Crystallographic Image Data**/ Irene M. Thomas, Michael F. Schmid // Verna and Marrs McLean Department of Biochemistry and W. M. Keck Center for Computational Biology. Baylor College of Medicine, Houston, TX 77030
- [7] Tao XU. **Research of Real-time Wide Field Image Merging Based on Multi-Cameras**/ XU Tao, CEN Zhao-feng, LI Xiao-tong // State Key Lab of Modern Optical Instrumentation, Zhejiang University, Hang Zhou 310027, China

Об авторах

Александр Александрович Самойлов – аспирант ННГАСУ. E-mail: greatlimit@rambler.ru.
Евгений Владимирович Попов – д.т.н., профессор ННГАСУ. E-mail: popov@sandy.ru.