

Министерство образования Российской Федерации
Московский Физико-Технический Институт
Факультет Физической и квантовой электроники
Центр открытых систем и высоких технологий МФТИ

Дипломная работа
на соискание ученой степени бакалавра
**Система трехмерного зрения и анализ ее погрешностей
применительно к задачам робототехники и спутниковой
съемке земного рельефа**

студентки 954 группы Свешниковой Натальи Владимировны.

Научный руководитель, к.ф.м.н.: _____ Д. В. Юрин

Зам. Заведующего кафедрой академик: _____ А. С. Бугаев

Долгопрудный
2003

Содержание.

1. ВВЕДЕНИЕ	3
1.1. ОБЗОР МЕТОДОВ ВОССТАНОВЛЕНИЯ ТРЕХМЕРНЫХ СЦЕН ПО ДАННЫМ ЛИТЕРАТУРЫ	4
1.1.1. Алгоритмы <i>Shape from Shading</i>	4
1.1.2. Алгоритмы <i>Shape from Focus and Defocusing</i>	7
1.1.3. Алгоритмы <i>Shape from Stereo</i>	12
1.1.4. Алгоритмы <i>Shape from Motion</i>	13
1.1.5. Экзотические методы	14
1.2. СРАВНИТЕЛЬНЫЙ АНАЛИЗ МЕТОДОВ ВОССТАНОВЛЕНИЯ 3D СЦЕН	15
2. СИСТЕМА ТРЕХМЕРНОГО ЗРЕНИЯ	17
2.1. ОБЩАЯ СТРУКТУРА СИСТЕМЫ ВОССТАНОВЛЕНИЯ ТРЕХМЕРНЫХ СЦЕН	17
2.2. ОБНАРУЖЕНИЕ ХАРАКТЕРИСТИЧЕСКИХ ТОЧЕК	20
2.2.1. <i>Сопровождение точек</i>	20
2.2.2. <i>Обнаружение характеристических точек</i>	22
2.3. ФОРМИРОВАНИЕ ПЛОТНОЙ МАТРИЦЫ ИЗМЕРЕНИЙ	23
2.3. ВОССТАНОВЛЕНИЕ 3D СЦЕН ПО ПОСЛЕДОВАТЕЛЬНОСТИ ЦИФРОВЫХ ИЗОБРАЖЕНИЙ МЕТОДОМ ФАКТОРИЗАЦИИ МАТРИЦ	24
2.3.1. <i>Постановка задачи</i>	24
2.3.2. <i>Математический вывод приближений</i>	28
2.3.3. <i>Геометрический смысл приближений</i>	31
2.3.3.1. Приближение ортографической проекции	31
2.3.3.2. Приближение масштабируемой ортографической проекции	31
2.3.3.3. Приближение параперспективной проекции	32
2.3.4. <i>Перспективная проекция</i>	36
2.3.5. <i>Сравнительный анализ приближений</i>	38
2.3.6. <i>Метод факторизации матриц</i>	39
2.3.7. <i>Итерационный метод решения 3D-задачи</i>	42
2.4. НАТЯГИВАНИЕ ПОВЕРХНОСТИ НА ВОССТАНОВЛЕННЫЕ ТОЧКИ И ПОСТРОЕНИЕ ТРЕХМЕРНОЙ МОДЕЛИ VRML	44
3. РЕЗУЛЬТАТЫ	47
3.1. ПОСТРОЕНИЕ МОДЕЛЕЙ ДЛЯ ГЕНЕРАЦИИ СИНТЕТИЧЕСКИХ ДАННЫХ	47
3.1.1. <i>Модель «Куб»</i>	47
3.1.2. <i>Модель «Спутник над плоскостью с прямоугольным выступом»</i>	49
3.2. СТРУКТУРА ПРОГРАММЫ ВОССТАНОВЛЕНИЯ ТРЕХМЕРНЫХ СЦЕН	50
3.3. СОВМЕЩЕНИЕ РЕЗУЛЬТАТОВ ВОССТАНОВЛЕНИЯ С МОДЕЛЬЮ	52
3.4. ОЦЕНКА ТОЧНОСТИ ВОССТАНОВЛЕНИЯ	54
3.4.1. <i>Модель «Куб»</i>	55
3.4.2. <i>Модель «Спутник над плоскостью с прямоугольным выступом»</i>	58
ЗАКЛЮЧЕНИЕ	59
ЛИТЕРАТУРА	60

1. Введение

Повышение быстродействия, резкий рост вычислительных ресурсов (оперативной памяти и дискового пространства), и доступность вычислительных средств, в настоящее время открывают новые возможности в области цифровой обработки изображений и их практической применимости в сферах робототехники и искусственного интеллекта. Если ранее усилия сосредоточивались на быстрых, однопроходных алгоритмах с минимальными затратами вычислительных ресурсов, которые, естественно, не могли обеспечить глубокой интеллектуальной переработки поступающей видеoinформации, то в настоящее время методы, основанные на преобразовании Фурье, которые ранее мало использовались в обработке изображений из-за больших вычислительных затрат и расхода ресурсов, сейчас стали стандартной частью систем MultiMedia. Считается, что производительность микропроцессоров возрастает вдвое каждые два года. Рост оперативной памяти и систем постоянного хранения информации идет, по-видимому, еще быстрее. В этих условиях возникает возможность разработки алгоритмов и аппаратно-алгоритмических комплексов, обеспечивающих детальную переработку видео данных от исходных изображений, до символического описания сцены, как перечня распознанных объектов, их положения, перемещений, поведения и изменения.

Задача восстановления и распознавания трехмерных сцен в настоящее время интенсивно разрабатывается большим числом исследователей и организаций (Massachusetts Technological Institute -Artificial Intellect laboratory and Media Lab¹, Carnegie-Melon University - Robotics institute and Computer Science Department², Central Florida University, Wisconsin-Madison University³, University of California Santa Barbara - Image Processing and Vision Research Lab, Rheinische Friedrich-Wilhelms-Universitat Bonn - Institut fur Informatik III, Brown University - Division of Engineering Laboratory for Engineering Man/Machine Systems, NASA⁴). Область применения этих алгоритмов также чрезвычайно широка. Это задачи навигации роботов и управления автомобилем [1], [2], [3], [4], предотвращение столкновений [5], лабораторные и промышленные системы измерения [6],[7]. Широкое применение алгоритмы восстановления трехмерных сцен в последнее время находят также в аэрокосмической отрасли. Здесь следует упомянуть совместный Советско-Американский проект РАМОС, конца 1980-х - начала 1990-х годов, стерео камеру, разработанную для Российского Марсианского зонда 1996 г. в Германии [8], спутники серии EOS-AM/PM Terra, Aqua (сканеры MISR, ASTER) [9],[10].

Разработано огромное количество методов и алгоритмов, однако все они имеют ограниченные области применения и "работают" не для всех сцен. В целом алгоритмы восстановления трехмерных сцен называют "Shape from X" (восстановление формы из X), где X-может принимать разные значения, основными подклассами этой большой группы алгоритмов являются:

- **"Shape from shading"** - восстановление формы (глубины) сцены по одиночному изображению на основе анализа изменения яркости. Как правило, используется ламбертовская модель рассеяния света объектом;
- **"Shape from Focusing and Defocusing"** - восстановление формы (глубины) сцены по набору изображений, снятых неподвижной камерой при различной степени расфокусировки (фокусировка на фрагменты сцены, расположенные на различном расстоянии от камеры);
- **"Shape from Stereo"** - восстановление формы (глубины) сцены из стерео пары изображений;
- **"Shape from Motion"** - восстановление формы сцены из последовательности изображений, снятых с разных позиций и в разные моменты времени (часто положения камеры тоже считаются неизвестными и восстанавливаются в ходе решения задачи);
- **"Shape from Zoom"** - восстановление формы сцены на основе последовательности изображений, снятых при фиксированном расположении камеры с различной степенью оптического увеличения. Более детальный обзор этих методов будет дан в следующем разделе.

¹ <http://www.ai.mit.edu/>, <http://www.media.mit.edu/>

² <http://www.ri.cmu.edu/>, <http://www-cgi.cs.cmu.edu/>

³ <http://www.wisc.edu/>

⁴ <http://www.lems.brown.edu/>

1.1. Обзор методов восстановления трёхмерных сцен по данным литературы

1.1.1. Алгоритмы Shape from Shading

Алгоритмы этой группы были разработаны одними из первых в начале 1970-х [11] для восстановления формы трехмерной сцены на основе видеоизображения. Однако они не утратили актуальности и поныне, разработка новых подходов к решению задачи продолжается до настоящего времени. Входными данными для этих алгоритмов является единственное серое полутоновое изображение сцены. Они основаны на том обстоятельстве, что для ламбертовых поверхностей яркость в каждой точке не зависит от положения наблюдателя и пропорциональна косинусу угла между нормалью к поверхности и направлением на источник освещения. Математически задача описывается следующим образом. Выберем систему координат (x,y,z) так, чтобы плоскость xu была параллельна плоскости изображения, а ось z была направлена к наблюдателю. Тогда поверхность можно описать функцией возвышений $Z(x,y)$. Вектор нормали к поверхности $Z(x,y)$ тогда записывается в виде:

$$\begin{aligned} \vec{n} = (n_x \quad n_y \quad n_z) &= \left(\frac{p}{\sqrt{1+p^2+q^2}} \quad \frac{q}{\sqrt{1+p^2+q^2}} \quad \frac{1}{\sqrt{1+p^2+q^2}} \right), \\ p(x,y) &= \frac{\partial Z(x,y)}{\partial x}, \\ q(x,y) &= \frac{\partial Z(x,y)}{\partial y} \end{aligned} \quad (1.1)$$

Здесь p и q - компоненты вектора градиента поверхности в направлении x и y соответственно. Пусть сцена освещается плоскопараллельным пучком в направлении

$$\begin{aligned} \vec{s} = (s_x \quad s_y \quad s_z) &= \left(\frac{p_s}{\sqrt{1+p_s^2+q_s^2}} \quad \frac{q_s}{\sqrt{1+p_s^2+q_s^2}} \quad \frac{1}{\sqrt{1+p_s^2+q_s^2}} \right), \\ p_s(x,y) &= \frac{\cos \tau \sin \Theta}{\cos \Theta}, \\ q_s(x,y) &= \frac{\sin \tau \sin \Theta}{\cos \Theta} \end{aligned} \quad (1.2)$$

где Θ - угол между направлением на источник излучения и осью z (зенитный угол, slant of the illuminant), τ - угол между проекцией направления на источник излучения от объекта на плоскость xu и осью x (азимутальный угол, tilt of the illuminant). Тогда для изображения $B(x,y)$ в предположении ламбертового рассеяния можно записать:

$$B(x,y) = A\rho_L(\vec{n},\vec{s}) = A\rho_L R(p,q) = A\rho_L \frac{\cos \Theta + p \sin \Theta + q \sin \tau \sin \Theta}{\sqrt{1+p^2+q^2}}, \quad (1.3)$$

где A - освещенность, создаваемая источником излучения в плоскости xu , ρ_L - альbedo поверхности для ламбертовского рассеяния. Уравнение (1.3) представляет собой нелинейное дифференциальное уравнение относительно $Z(x,y)$ в частных производных с переменными коэффициентами (из-за $B(x,y)$). В случае неламбертовского рассеяния, правая часть (1.3) принимает более сложный вид. С учетом спекулярного рассеяния, наиболее общее выражение выглядит так [12]:

$$B(x, y) = A\rho_L(\vec{n}, \vec{s}) + A\rho_{SL}e^{-\frac{\beta^2}{2\sigma^2}} + A\rho_{SS}\delta(\Theta_i - \Theta_r)\delta(\tau_i - \tau_r) \quad (1.4)$$

Здесь $\beta = \arccos([\mathbf{n}, \mathbf{s}])$ - угол между нормалью к поверхности и биссектрисой угла между направлениями на источник освещения \mathbf{s} и точку изображения \mathbf{b} , σ - ширина диффузной части specularного рассеяния (specular lobe), Θ_i, τ_i и Θ_r, τ_r - зенитный и азимутальные углы источника излучения и направления зрения (точки изображения) соответственно. Последнее слагаемое в (1.4) отвечает за зеркальную компоненту specularного рассеяния (specular spike), которая отлична от нуля только в очень узком диапазоне углов в направлении зеркального отражения на малой плоской площадке, касательной к поверхности Z\$ в точке (x,y). Одним из простейших методов является линейный метод Пентланда [13]. Уравнение (1.3) линеаризуется и, с учётом (1.1), принимает вид:

$$\tilde{B}(x, y) = \frac{B(x, y)}{A\rho_L} - \cos \Theta = \frac{\partial Z}{\partial x} \cos \tau \sin \Theta + \frac{\partial Z}{\partial y} \sin \tau \sin \Theta \quad (1.5)$$

От обеих сторон берется преобразование Фурье. Тогда Фурье-образ поверхности Z может быть выражен в явном виде:

$$F_z(\varepsilon, \eta) = \frac{F_{\tilde{B}}(\varepsilon, \eta)}{2i\pi(\varepsilon \cos \tau \sin \Theta + \eta \sin \tau \sin \Theta)} \quad (1.6)$$

Путем взятия обратного преобразования Фурье, получается форма поверхности Z(x,y). Детальный обзор современных алгоритмов по этой тематике дан в [14]. В [15] дан сравнительный анализ погрешностей этих алгоритмов. Существуют алгоритмы, пригодные для поверхностей с законом рассеяния, отличным от ламбертовского [16],[17]. Основными недостатками алгоритмов "Shape from Shading" являются требования постоянства альbedo по всей сцене и априорного знания закона рассеяния (то есть двунаправленной функции рассеяния BDRF⁵), а также, потеря информации об абсолютных размерах и расстояниях. Однако, в ряде случаев, эти алгоритмы успешно использовались для решения практических задач, например для расчета формы Лунной поверхности на основе снимков. Достоинством таких алгоритмов является возможность получения результата на основании единственного изображения. Знание точных условий освещения, что характерно для изображений планет и астероидов, существенно облегчает задачу, и повышает точность результатов. Для безатмосферных небесных тел, в свою очередь, постоянство альbedo в пространстве, является типичным. Если условия освещения не известны, они могут быть восстановлены из самого изображения с помощью группы алгоритмов IDE (Illumination Direction Estimation), однако, в этом случае, приходится делать определенные допущения для статистических свойств поверхности, например, о равной вероятности уклонов поверхности для всех направлений, или локальной сферичности поверхности. В случае, когда допущения не соответствуют действительности, оценка направления освещения дает неверный результат, и приводит к неправильному восстановлению формы поверхности с помощью алгоритмов Shape from Shading. Иллюстрирующий пример приведен на рис. 1.1.

⁵ BiDirectional Reflectance Function

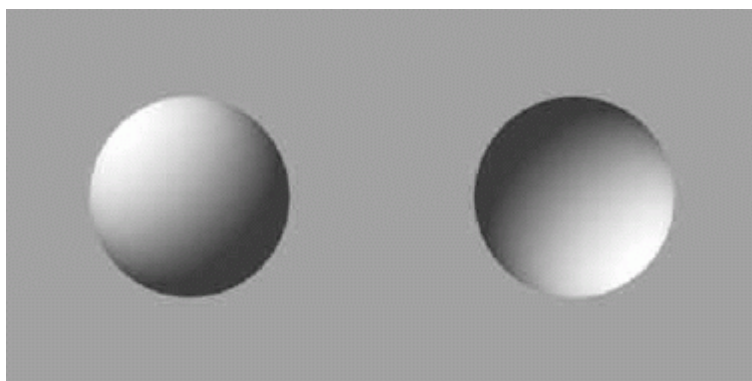


Рис. 1.1. Влияние направления освещения на восстановление формы объекта

Кажется, что слева изображена сферическая выпуклость на плоскости, а справа - вдавленность. На самом деле, это один и тот же объект, изображенный в условиях различной освещенности. Левая полусфера освещена с направления левого верхнего угла изображения, а правая получена поворотом изображения на 180 градусов, что соответствует освещению с правого нижнего угла изображения. Восприятие формы в качестве выпуклости, или вдавленности, основывается на давно известном для дешифровщиков изображений факте, что человек при просмотре изображений психологически считает, что освещение сцены направлено со стороны левого верхнего угла изображения. На этом примере легко понять, что, при работе алгоритмов восстановления формы по одному изображению неверное задание направления освещения, может привести к результату с точностью до наоборот.

Существуют методы, позволяющие в некоторых случаях применять алгоритмы типа "Shape from Shading" для сцен с неодинаковым альбедо, описанные, например [18]. В этой работе сначала оценивается величина альбедо с помощью локальных методов для каждой точки изображения, затем изображение сегментируется по величине альбедо, и значения пикселей исходного изображения делятся на средние значения альбедо по соответствующим сегментам, после чего используется обычный алгоритм восстановления формы для постоянного альбедо, равного единице. Используемый метод расчета альбедо [20] не работает для некоторых типов поверхности, в частности плоской. Очевидно, что для объекта с переменным альбедо общего вида не возможно восстановление формы по одному изображению. Так, например, изображение объекта, приведенное на рис. 1.1 (полусфера), и изображение фотографии этого объекта (плоскость), идентичны.

Заметим в заключении, что при анализе сложных сцен, алгоритмы этой группы тоже играют важную роль. Так, например, пусть сцена содержит изображение городского пейзажа, включая здания растительность и т.п. И пусть сцена включает объект X, сложной формы из неокрашенного бетона, не содержащий ярко выраженных ребер и углов, имеющий плавные, округлые поверхности; в простейшем случае, шар. Понятно, что восстановить всю сцену с помощью алгоритмов "Shape from Shading" не удастся. Однако восстановить форму объекта X с помощью триангуляции на основе стерео изображений или последовательности изображений тоже не удастся, поскольку на объекте X отсутствуют ярко выраженные характерные точки и текстуры. В этом случае единственный выход заключается в том, чтобы выделить такой объект из изображения сцены и применить для него один из алгоритмов типа "Shape from Shading", возможно при дополнительном краевом условии, что трехмерные координаты некоторых точек границы такого объекта, уже получены другими методами.

1.1.2. Алгоритмы Shape from Focus and Defocusing

Алгоритмы восстановления формы сцены по фокусировке и дефокусировке, основываются на конечности глубины резкости оптических систем. Действительно, пусть расстояние до точки объекта равно z , а фокусное расстояние объектива - f , тогда изображение этой точки будет резким только в случае, когда выполняется соотношение тонкой линзы:

$$\frac{1}{f} = \frac{1}{z} + \frac{1}{L} \quad (1.7)$$

где L -- расстояние от линзы до плоскости изображения. Так как для неплоского объекта расстояния от объектива до его различных точек разные, то оптическая система может быть наведена на резкость только для части точек объекта.

Методы восстановления по фокусировке основаны на управлении фокусировкой камеры таким образом, чтобы найти, при каких параметрах фокусировки, достигается резкое изображение различных точек объекта. Достижимое этими методами пространственное разрешение сцены, вдоль оси камеры (оси z), может быть оценено, исходя из выражения для глубины резкости [20]:

$$\delta z = \frac{z^2}{f_r} \Delta, \quad \Rightarrow \quad \delta \left(\frac{1}{z} \right) = \frac{\Delta}{f_r} \quad (1.8)$$

Здесь r - радиус входного зрачка объектива, Δ - диаметр кружка в плоскости изображения, которым представляется точка объекта, в цифровых системах изображение можно считать резким, если Δ меньше размера пикселя фоточувствительной матрицы. Заметим, что точность определения величины $1/z$, определяется только свойствами оптической системы, и не зависит от z .

В [21] предлагается алгоритм для нахождения глубины сцены по фокусировке, отличающийся от ранее известных, существенно большей точностью. В качестве меры сфокусированности в точке изображения, выбиралась величина M_2 [22],[23]:

$$M_2 = \iint |\nabla F(x, y)|^2 dx dy, \quad (1.9)$$

с оператором градиента Собела [24], который вычисляется следующим образом:

$$G(x, y) = |\nabla F(x, y)| = \sqrt{X^2 + Y^2} \quad (1.10)$$

где

$$\begin{aligned} X &= (A_2 + 2A_3 + A_4) - (A_0 + 2A_7 + A_6), \\ Y &= (A_0 + 2A_1 + A_2) - (A_6 + 2A_5 + A_4), \end{aligned} \quad (1.11)$$

X, Y берутся в обозначениях элементов окна для оператора Собела:

$$\begin{pmatrix} A_0 & A_1 & A_2 \\ A_7 & F(i, j) & A_3 \\ A_6 & A_5 & A_4 \end{pmatrix} \quad (1.12)$$

Практически выражение (1.9) вычислялось, как среднее по окну 40×40 пикселей от величин (1.10), посчитанных для каждой точки окна (i, j) . Оператор Собела (1.10) вычисляется по окну 3×3 , как показано в (1.12).

Поскольку вычисление глубины сцены из фокусировки требует плотного взаимодействия алгоритма с мотором камеры, управляющим фокусировкой, то, в силу линейности зависимости между позицией мотора и смещением объектива, а также в виду требования предварительной калибровки камеры, результаты приводятся относительно отсчетов позиции мотора (цифровое управление). На стадии калибровки снималось тестовое изображение (см. рис 1.2):

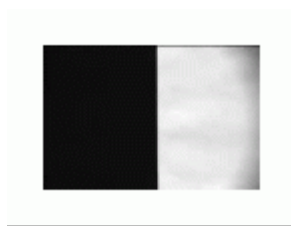


Рис. 1.2. Тестовое изображение для Shape from Defocus

На рис.1.3 приведено значение меры M_2 в зависимости от отсчетов мотора. Для нахождения точки максимума использовался метод поиска Фибоначчи [24], совместно с аппроксимацией зависимости гауссовой функцией вблизи максимума. Последнее является особенностью метода [21], которая позволяет существенно повысить точность восстановления за счет исключения обнаружения локальных максимумов.

На рис. 1.4, 1.5 приводятся кривые M_2 вблизи максимума, с наложенными на них аппроксимациями, полученными методом поиска Фибоначчи, и подгонкой к гауссовой функции.

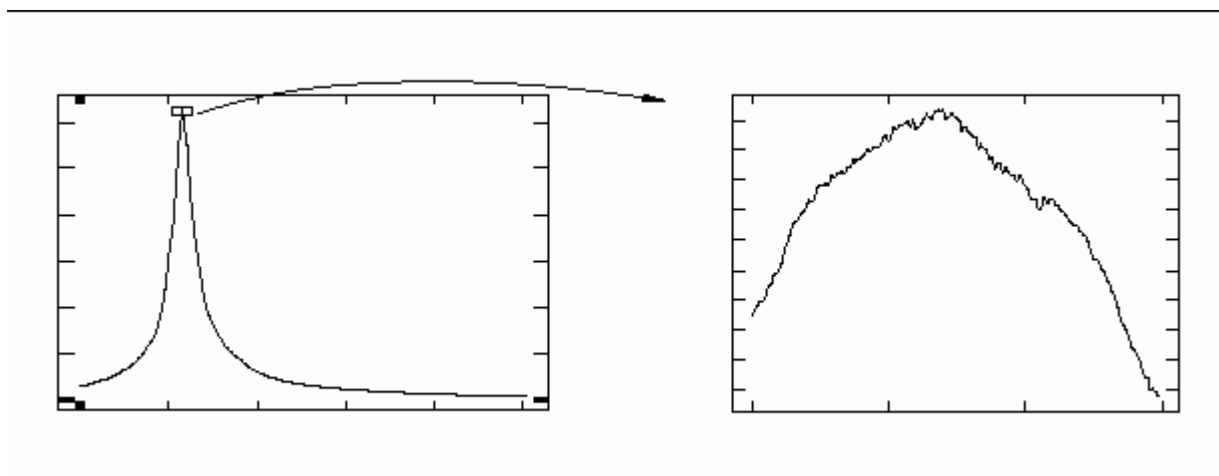


Рис. 1.3. Мера Фокусировки M_2

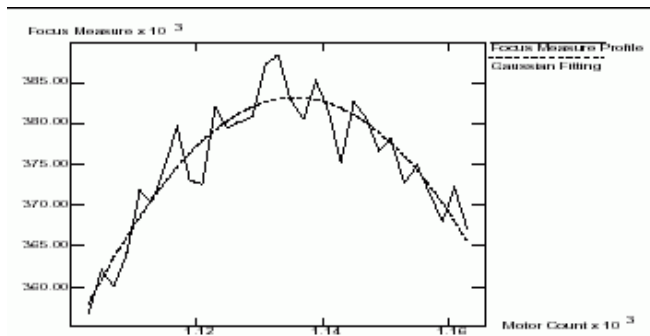


Рис. 1.3. Аппроксимация данных гауссовой кривой

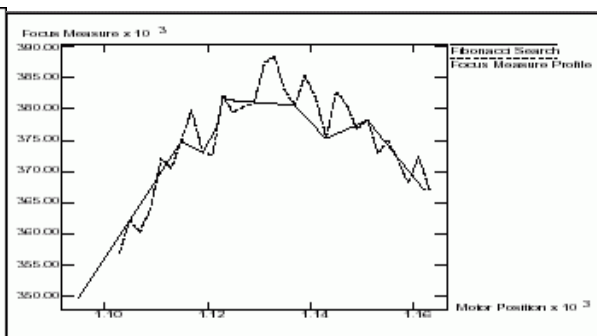


Рис. 1.4. Поиск Фибоначчи

В [21] докладывается о достигнутой относительной точности измерения расстояния 0.098 процентов, при расстоянии до объекта 120 см, и прецизионной камере с отношением сигнал/шум 400/1, фокусным расстоянием от 10мм до 130мм, с 11100 шагами мотора, и апертурой от F1.7 до почти закрытой с 2700 шагами

мотора. В статье не рассматривается восстановление формы сложного объекта этим методом, однако представляется, что для объекта, существенно отличающегося от плоскости, перпендикулярной оптической оси, потребуется фактически снять изображения при всех возможных значениях отсчетов мотора, и совместно обрабатывать все эти изображения, что потребует значительных вычислительных затрат и большого времени, ограничиваемого ещё и инерционностью мотора. В этой связи представляют интерес методы нахождения формы объекта из дефокусировки, поскольку при работе с ними требуется наличие только двух изображений, снятых при различных положениях мотора, что обещает более привлекательные характеристики по быстродействию и объему перерабатываемых данных, несмотря на большую сложность алгоритмов. Эффект размытия изображения за счет дефокусировки моделируется в [21] гауссовой функцией. Обозначая идеальное изображение через $I_0(x,y)$, функцию размытия $g_\sigma(x,y)$ для расфокусированного изображения можно записать так:

$$I(x,y) = \iint I_0(\xi,\zeta) g_\sigma(x-\xi, y-\zeta) d\xi d\zeta, \quad (1.13)$$

$$g_\sigma(x,y) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{x^2+y^2}{2\sigma^2}}.$$

Здесь σ определяет размер пятна дефокусировки в плоскости изображения, и зависит от глубины сцены и параметров камеры. Так как методы восстановления формы по дефокусировке получают глубину из двух изображений, а не осуществляют поиск, требуется модель, описывающая связь между размером пятна дефокусировки и глубиной сцены. Принимая во внимание формулу, полученную Пентландом [25] в приближении геометрической оптики, в [21] предлагается ее обобщение:

$$\sigma^2 = \left(k_1(m_z, m_f, m_a) + \frac{k_2(m_z, m_f, m_a)}{D + k_3(m_z, m_f, m_a)} \right)^2 + k_4^2(m_z, m_f, m_a), \quad (1.14)$$

где D -- расстояние от точки сцены до объектива, коэффициенты $k_1 - k_4$ зависят от отсчетов моторов для zoom-а, фокусировки и апертуры. В статье приводится калибровочная процедура, позволяющая построить эти зависимости, и показывается высокая точность аппроксимации, даваемая моделью (6).

Сначала предположим, что объект представляет собой плоскость, перпендикулярную оптической оси, то есть глубина картины есть константа. Тогда, переходя к Фурье-образам изображения $I \rightarrow F$, и функции размытия $g \rightarrow G$, из (1.13) можно получить следующее:

$$\ln \frac{F_1(u,v)}{F_2(u,v)} = \ln \frac{F_0(u,v)G_{\sigma_1}(u,v)}{F_0(u,v)G_{\sigma_2}(u,v)} = \ln \frac{G_{\sigma_1}(u,v)}{G_{\sigma_2}(u,v)} = -\frac{1}{2}(u^2+v^2)(\sigma_1^2 - \sigma_2^2) - \frac{1}{2}(u^2+v^2)\Delta_\sigma \quad (1.15)$$

Таким образом, исходя из двух изображений, снятых при различных позициях моторов, вычисляется их преобразование Фурье, и ищется методом наименьших квадратов наклон кривой в зависимости от частот (u^2+v^2) . Пользуясь моделью (6) можно вычислить глубину сцены. Если объект имеет более сложную форму, то глубину сцены надо вычислять для каждого пикселя изображения, что приводит к необходимости введения пространственного окна для вычисления спектров. В [21] предлагается использовать окно гауссовой формы, которое минимизирует произведение пространственной и частотной неопределенности. В результате задача сводится к расчету разности логарифмов преобразований Габора от двух исходных изображений, а задача аппроксимации прямой становится трехмерной. В статье предлагается итерационный алгоритм локальных оценок Δ_σ , названный авторами оценкой максимального правдоподобия (Maximal Resemblance Estimation).

Обозначая номер итерации индексом k , и подразумевая, что $\Delta_{(k)} = \sigma_1^2 - \sigma_1^2$ на итерации k , алгоритм может быть сформулирован следующим образом:

1. $I_1^{(0)} = I_1, I_2^{(0)} = I_2, \Delta_{(0)} = 0, k=0;$
2. Вычислить преобразование Фурье $I_1^{(k)} = F[I_1^{(k)}W], I_2^{(k)} = F[I_2^{(k)}W]$, где гауссова функция W описывает окно преобразования Габора.

3. Выполнить МНК-подгонку прямой $\frac{I_1(u, v)}{I_2(u, v)} = -\frac{1}{2}(u^2 + v^2)\Delta_{(k)}$
4.
$$\Delta = \sum_{i=1}^k \Delta_{(i)}$$
5. if $\Delta > 0$, then

$$I_1^{(k+1)} = I_1;$$

$$I_2^{(k+1)} = I_2 \otimes G_{\sigma=\sqrt{\Delta}};$$
 else

$$I_1^{(k+1)} = I_1 \otimes G_{\sigma=\sqrt{-\Delta}};$$

$$I_2^{(k+1)} = I_2;$$
 endif

Замечание: свертки могут быть выполнены очень локально из-за умножения на функцию окна на шаге 2.
6. if выполнено условие сходимости

$$\text{exit};$$
 else

$$k=k+1;$$

$$\text{goto step 2}$$
 endif

Все операции в приведенном алгоритме выполняются над пикселями только локально, и не требуют получения новых изображений, поэтому алгоритм хорошо распараллеливается.

Существенное отличие от ранее предложенных методов восстановления трехмерной формы по дефокусировке в подходе [21] заключается в том, что отсутствует выбор фиксированной пространственной частоты, на которой осуществляется сравнение амплитуд спектра, что снижает ошибки, обусловленные малой долей выбранной частоты в спектре изображений. Исследованы погрешности и точности, однако, как отмечают авторы, сходимость метода математически не доказана. Последнее типично для задач восстановления трехмерных сцен. Для характеристик аппаратуры, приведенных в разделе, посвященном нахождению формы из фокусировки, приводится достигнутая относительная точность восстановления, равная 0.5 процентов глубины, объект располагался на расстоянии 100 дюймов от камеры. Касаясь алгоритмов восстановления формы по фокусировке и дефокусировке, важно упомянуть фактор, осложняющий применение этих алгоритмов. Это изменение масштаба изображения при смещении плоскости изображения, суть явления становится понятна из рис. 1.6. Этот фактор особенно существенен для алгоритмов восстановления по дефокусировке, так как он затрудняет сопоставление образов одних и тех же точек объекта на изображениях различной степени дефокусировки, и более того, существенно зависит от расстояния до точки объекта.

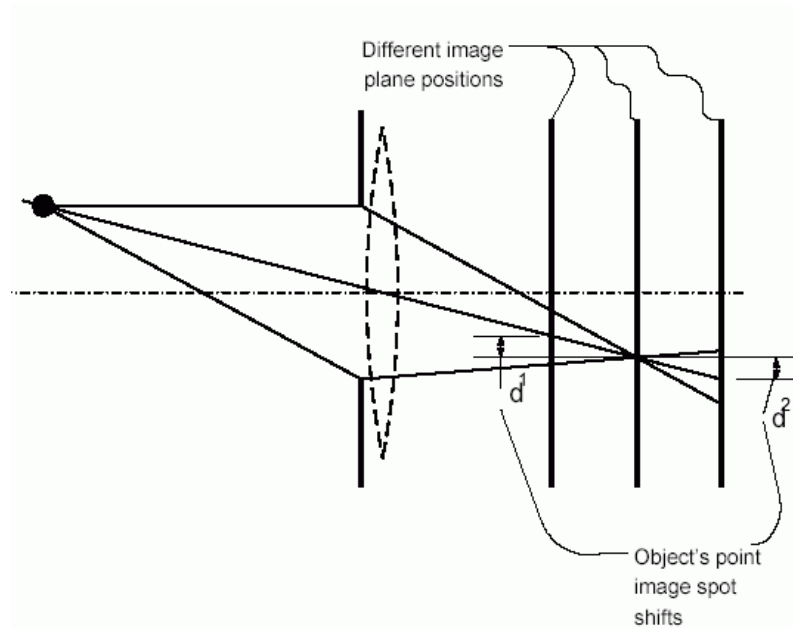


Рис. 1.6. Смещение изображения точки в зависимости от степени дефокусировки

Для устранения этого мешающего фактора в [26] предложена несложная модификация оптической системы (объектива), заключающаяся во введении дополнительной диафрагмы (См. также [20]). Интересно также отметить созданную на принципах восстановления по дефокусировке, установку для прецизионного восстановления формы объектов в лабораторных условиях [6],[7]. В контролируемых условиях эксперимента авторы изящно обошли ряд трудностей. Для устранения проблем, связанных с недостатком текстуры, снимаемый объект освещается структурированным освещением (шахматная клетка), причем масштаб этой сетки согласован с размером пикселя фоточувствительных матриц так, что вычисление сверток с гауссовыми функциями удастся заменить оператором Лапласа по окну размера 5x5. Съемка осуществляется одновременно двумя камерами через один объектив с использованием светоделительной пластины, таким образом отпадает необходимость в постоянном управлении мотором фокусировки, система один раз перед экспериментом может быть отъюстирована, и работает в реальном масштабе времени. Подобная схема, в принципе, может быть применена и в робототехнике. Ее достоинством является работа на минимальных расстояниях, что полезно для проблемы предотвращения столкновений, и в силу одновременного получения и синхронизации обоих снимков, возможность правильного определения формы как неподвижных, так и движущихся объектов. Освещение структурированным светом также может быть применено в некоторых условиях, так как это зачастую предпочтительнее использования лазера, СВЧ, или звуковой локации. Проблемой здесь является отсутствие в настоящее время интереса промышленности к массовому выпуску подобных сложных узкоспециализированных оптических систем.

1.1.3. Алгоритмы Shape from Stereo

По типу обрабатываемых особенностей, алгоритмы восстановления формы сцены на основании стереоизображений делятся на три класса: **алгоритмы для геометрических особенностей**, основанные на поиске характерных точек (feature-based approaches), **алгоритмы для областей** (area-based approaches) [27], а также **смешанные алгоритмы** (miscellaneous approaches) [28], [29], [30].

В методах, основанных на поиске характерных точек, ищутся точки со значительными перепадами яркости, или какого-либо признака изображения (например, текстуры, или цветового тона) на обоих кадрах. Часто ищутся не точки, а линии, в том числе, они могут получаться путем сегментации обоих изображений, и выбора границ сегментов в качестве характерных линий. Между найденными на двух изображениях точками устанавливается взаимно однозначное соответствие путем вычисления корреляции фрагментов изображений в окрестности этих точек или, в случае сегментации, путем сравнения интегральных характеристик сегментов. Для точек, в которых взаимно однозначное соответствие установлено, вычисляется расстояние до них методом триангуляции. Расстояние до остальных точек объекта получают посредством интерполяции. Математически, такие алгоритмы принадлежат к широкому классу алгоритмов "Shape from Motion".

Алгоритмы для областей базируются на поиске расстояния до всех точек изображения.

Существует несколько смешанных методов, таких как стохастический алгоритм Бернарда [28]. Этот метод плохо работает при недостатке пространственной структуры. Его суть заключается в вычислении карты смещений (disparity) для каждой точки изображения путем минимизации функционала, соответствующего критериям близких интенсивностей и гладкости:

$$E = \sum_{j=1}^m \sum_{i=1}^n (\|I_L(i, j) - I_R(i + D(i, j), j)\| + \lambda \|\nabla D(i, j)\|), \quad (1.16)$$

где $I_L(i, j)$, $I_R(i, j)$ - значения пикселей левого и правого изображений, $D(i, j)$ - карта смещений, λ - константа, оператор ∇ вычисляет сумму абсолютных разностей между смещением $D(i, j)$ и его 8 ближайшими соседями. Задача решается путем численного моделирования Монте-Карло.

Большинство алгоритмов группы "Shape from Stereo" находят только грубые детали формы объекта, особенно при недостатке пространственной структуры. В то же время, по сравнению с "Shape from Shading", пространственная локализация крупных деталей, особенно их ребер, и контрастных границ, происходит существенно точнее, а расстояния могут быть восстановлены в абсолютных величинах. Поэтому представляет интерес подход, изложенный в [31], который позволяет объединить эти два метода. Предложенный метод основан на модели человеческого зрения [32], которое выполняет ту же задачу. Представляя поверхность $Z(x, y)$ в виде Фурье-образа $F_Z(u, v)$, математически этот метод может быть описан так:

$$F_{Z_c}(u, v) = F_{Z_{st}}(u, v)H^{-1}(\omega) + F_{Z_{ss}}(u, v)(1 - H^{-1}(\omega)), \quad (1.17)$$

$$H(\omega) = \frac{\alpha^2 + \omega^2}{2\alpha\alpha_0 + (1 - \alpha_0)(\alpha^2 + \omega^2)}, \quad (1.18)$$

$$w = \sqrt{u^2 + v^2},$$

где $F_{Z_c}(u, v)$, $F_{Z_{st}}(u, v)$, $F_{Z_{ss}}(u, v)$ -- Фурье-образы поверхностей, полученных из стерео, полутонов, и в результате их объединения. $H(\omega)$ -- высокочастотный фильтр из модели, предложенной в [32] Для зрительной системы человека $\alpha = 0.01$, $\alpha_0 = 0.2$.

1.1.4. Алгоритмы Shape from Motion

В настоящее время, из всех методов реконструкции трехмерных сцен, алгоритмы восстановления сцены по изображениям, полученным с разных положений камеры, и в различные моменты времени, исследуются наиболее интенсивно [33], [34], [35]. При этом часто предполагается, что фокусное расстояние объектива неизвестно, или меняется от кадра к кадру. Кроме того, не редко, положения камеры на момент съемки каждого кадра также полагается неизвестным, и находится в ходе решения задачи реконструкции трехмерной сцены. Как правило, такие методы основываются на поиске характерных точек (feature points) на изображениях, в виде уголков или линий, а также на поиске соответствий между обнаруженными точками на последовательности кадров⁶, нахождении их пространственных позиций по принципу триангуляции, и построении аппроксимирующих поверхностей. Подробное изложение наиболее перспективных методов этой группы, основанных на факторизации матриц, будет приведено ниже в основной части настоящей работы. Отметим здесь лишь, что указанные методы не работают непосредственно с изображениями, а требуют на вход координаты характеристических точек изображений в пикселях, и наличие у каждой характеристической точки маркера (номера), причем, на всей последовательности изображений, одной и той же точке реальной сцены должен соответствовать одинаковый маркер.

Для нахождения характеристических точек существуют различные алгоритмы [1], [36], [37], [38]. Задача нахождения взаимно однозначного соответствия между характеристическими точками на различных изображениях обрабатываемой последовательности, обычно решается методами траекторного анализа и калмановской фильтрации [34].

Наиболее перспективными из этой группы являются методы, основанные на факторизации матриц [39], [40], [41], [42]. В [39] наиболее подробно освещаются основы метода и рассматриваются его детали. В [40] наиболее компактно и подробно описаны различные приближения. В [41] рассматривается применение метода в случае наличия движущихся объектов на сцене, и предлагается подход к определению числа таких объектов, и их разделению. В [42] предлагается модификация метода, не требующая предварительного накопления всех данных, но позволяющая уточнять модель по мере поступления новых данных, и отличающаяся лучшим быстродействием, и меньшими затратами вычислительных ресурсов. Однако, по-видимому, этот метод имеет ограниченное время непрерывной работы из-за переполнения плавающей арифметики, и накопления погрешностей, так как на каждом шаге получения новых данных, они добавляются к ковариационной матрице измерений, а никаких мер по предотвращению подобной ситуации не предлагается. В статье [39] упоминается подход к применению метода в тех случаях, когда не все характеристические точки видны на всех кадрах.

⁶ Correspondence problem

1.1.5. Экзотические методы

Среди методов восстановления трехмерных сцен по их оптическим цифровым изображениям, встречаются довольно необычные. Так, в [43] предлагается метод, основанный на учете влияния рассеяния и поглощения в атмосфере. Суть метода заключается в нахождении на изображении участков с предположительно похожими отражательными свойствами, и сопоставлении яркостей таких участков. Учитываются эффекты освещения рассеянным излучением от неба, яркость которого измеряется наведением камеры на участок неба, удаленный от направления на Солнце, или выделения на изображении участков, относящихся к поверхности, или небу. В случае тумана, за яркость "неба" принимается яркость сильно удаленных объектов, то есть невидимых сквозь туман. Индикатриса рассеяния считается симметричной относительно направления падения освещения. В статье приводятся некоторые результаты натурных экспериментов, точность измерения расстояний для приведенных результатов лежит в пределах от 2.7 до 15.9 процента, что для подобного метода представляется удивительно хорошим. Как справедливо отмечают авторы, этот метод имеет весьма ограниченную область применения, однако может служить дополнительным источником информации для мобильных роботизированных систем.

Другой интересный метод описан в статье [44]. Рассматривается одиночное цветное изображение. Выполняется сегментация не по цвето-яркостным и текстурным характеристикам, а по объектам, отличающимся трехмерной формой. Сначала выполняется обычная сегментация изображения по цветовому признаку, которая, в рамках поставленной задачи, является избыточной для раскрашенных объектов. Затем, для каждого такого сегмента строится набор гипотез, состоящих в его цвете, освещении (цветном или белом), форме (плоская или не плоская, что оценивается методами Shape from Shading, в частности методом [13]), пластик или металл (спекулярное отражение). Строится мера похожести формы границы между двумя сегментами. Так, если два граничащих сегмента представляют собой, различным образом окрашенные части одной поверхности, то на границе сегментов происходит скачок яркости (цвета), однако вдоль границы яркость точек изображения меняется согласованно. Из указанных гипотез о свойствах первичных сегментов и весов связей между ними, строится взвешенный мультиграф гипотез, описывающий сцену. Каждый сегмент представлен набором листьев графа, соответствующих различным гипотезам об этом сегменте. Приводятся таблицы совместимости гипотез для соседних листьев. Ребра мультиграфа соединяют листья, относящиеся только к разным соседствующим сегментам (таким образом, граф не имеет петель и не является псевдографом, но является мультиграфом, задача не заключается в выделении из него простого графа), веса ребер определяются на основе совместимости гипотез для соседних сегментов (листьев). Потом решается задача выбора подграфа с максимальным весом, и его редуцирования. Ценность предложенного подхода состоит в том, что это одна из не многих попыток интеграции разнородной информации, полезной для восстановления сцены, производящая не просто склейку фрагментов сцены, восстановленных различными методами, но и анализ возможности соседства таких фрагментов, их стыковку между собой. В том числе, предложенный подход может оказаться очень полезным в случае, когда какие-либо фрагменты сцены восстанавливаются ненадежно, и по-разному различными методами, - тогда анализ окружения и совместимости, дает подход к автоматическому выбору результата, который наилучшим образом вписывается в контекст - наиболее достоверной гипотезы. Последнее обстоятельство существенно, так как в реальных сценах даже глаз и мозг человека не всегда правильно интерпретируют наблюдения, несмотря на их сложность, и массу используемых дополнительных сведений и знаний. Примером тому могут служить приведенная выше иллюстрация к алгоритмам Shape from Shading, и успешное существование искусства иллюзионистов.

1.2. Сравнительный анализ методов восстановления 3D сцен

Из анализа литературы по теме реконструкции трехмерных сцен и анализу изображений, ясно, что для сцен общего вида, задача не может быть решена в рамках какого-то одного подхода (алгоритма). В настоящее время разработано огромное количество различных подходов, которые имеют как сильные, так и слабые стороны, и ограниченные условия применения. Таким образом, возникает необходимость разработки интегрированной системы, сочетающей в себе достоинства различных подходов для построения моделей сложных трехмерных сцен, путем интегрирования информации, извлекаемой различными методами, и выбора методов, наиболее подходящих для различных участков сцены. В качестве основы такой системы, представляется целесообразным выбрать методы класса "Shape from Motion", основанные на факторизации матриц. Оправданием такого выбора являются следующие достоинства этой группы методов:

1. относительно невысокие вычислительная сложность и объем перерабатываемой информации при обработке данных, извлеченных из значительного количества кадров;
2. потенциально высокая точность, обусловленная большими базами наблюдений, обеспечиваемыми движением системы зрения в пространстве объектов;
3. согласованная обработка информации, получаемой из большой последовательности кадров;
4. возможность построения системы, рекурсивно уточняющей, и достраивающей модель сцены на основании данных, получаемых при наблюдении с других позиций системы зрения в пространстве реальной сцены;
5. одновременное восстановление информации о движении самой системы зрения, что, в сочетании с данными инерциальных и спутниковых систем (гироскопы, акселерометры, компасы, GPS), дает обратную связь, что позволяет контролировать разумность получаемых результатов, и открывает возможности автокалибровки;
6. возможности автокалибровки собственно оптической части системы (фокусировка, zoom, aspect ratio);
7. возможности обработки и представления модели сцены с движущимися объектами, и их движений.

По существу, методы, основанные на факторизации, позволяют восстановить, и отслеживать каркас трехмерной сцены и траекторию движения системы технического зрения в пространстве сцены. Остальные методы восстановления формы могут быть интегрированы с такой моделью двояким образом. С одной стороны, методы, основанные на факторизации матриц, восстанавливают трехмерные координаты только некоторых точек сцены, поэтому возникает задача интерполяции поверхностей сцены между восстановленными точками модели.

Для поверхностей, имеющих текстуру, зависимость текстур от масштаба, расстояния до объекта, степени регулярности текстур, движения объектов сцены, и имеющегося в распоряжении времени, могут использоваться методы "Shape from Stereo", "Shape from defocusing", или "Shape from Focusing", а также их комбинации. Для поверхностей, не имеющих текстуры, применимы методы типа "Shape from Shading". Здесь важно отметить два обстоятельства. Во-первых, применение этих методов к уточнению каркаса, полученного факторизацией, заменяет задачу интерполяции на задачу восстановления истинной формы сцены. Во-вторых, путем интерполяции из модели, полученной факторизацией, может быть выбрано начальное приближение для указанных уточняющих методов, наложены граничные условия, и привлечены дополнительные данные для повышения устойчивости таких методов, например средняя нормаль к участку поверхности, примерный диапазон, в котором варьируется расстояние от камеры до участка поверхности, выбор для запуска таких методов точки зрения, обеспечивающей лучшие условия для применения выбранных методов. Другим важным способом интеграции прочих методов с каркасной моделью, является их использование до выполнения очередного шага уточнения, и достраивания каркаса с целью предварительной оценки расстояний (для упрощения задачи траекторного анализа и устранения неоднозначностей при сопоставлении характеристических точек), выявления движущихся объектов (в том числе движущихся только в некоторые интервалы времени), обнаружения внезапных изменений сцены, и предотвращения столкновений мобильной системы с объектами сцены. Последнее для мобильных систем представляется особенно важным, причем выбор

метода предотвращения столкновений существенно зависит от характера как сцены, так и самой мобильной системы. В первую очередь, это, разумеется, планирование движения, основываясь на каркасной модели.

Для медленно движущихся систем наиболее перспективными представляются методы, основанные на фокусировке и дефокусировке, поскольку они работают вплоть до расстояний прямого соприкосновения. Для более быстро движущихся систем предсказывать столкновение надо заранее, и здесь могут найти применение системы на основе стерео зрения, учитывая также, что их работа облегчается для больших расстояний за счет снижения диапазона поиска корреляции (disparity).

В некоторых случаях требуется комбинация методов. Так для самолета на аэродроме применимы методы по дефокусировке, а в воздухе - стерео, и, в ряде случаев, например при плохой видимости или высоких скоростях, - даже методы, производящие восстановление по рассеянию в атмосфере.

В заключении отметим, что полезную информацию для задачи восстановления трехмерных сцен, дают также алгоритмы распознавания образов и классификации. Наиболее очевидными их применениями является: выбор зон интереса, где требуется более точное восстановление, и уточнение расстояний и масштаба сцены в случаях, когда удастся найти объект с заранее известными размерами, например серийный автомобиль, или типовую бетонную панель, описание которых известно, и может быть помещено в базу знаний экспертной системы.

2. Система трехмерного зрения

2.1. *Общая структура системы восстановления трехмерных сцен.*

Учитывая сказанное выше, система должна строиться таким образом, чтобы в нее было возможно интегрировать различные алгоритмы восстановления трехмерных сцен. В качестве основы системы была выбрана группа алгоритмов, основанных, на факторизации матриц [39], [40], [46]. Основанием для такого выбора послужили следующие причины:

1. Возможность восстановления не только формы сцены, но и движения и ориентации камер, и их фокусных расстояний.
2. Возможность работы с нестационарными сценами [40].
3. Высокая точность, обусловленная потенциально большими базами съемки и самосогласованностью решения, основанного на сильных требованиях симметрии (ранг матрицы).
4. Возможность выбора различных приближений, применимых в диапазоне расстояний до объекта от менее и порядка базы [46] (максимального движения камеры) до бесконечности [39].
5. Невысокая вычислительная сложность, обусловленная тем, что в решении используются не изображения, а только координаты характеристических точек.

В результате применения подходов [39], [40], [46] получаются трехмерные координаты некоторых точек объекта с высокой точностью, по которым может быть построена поверхность в виде сетки или меша [47-49], координаты и ориентация камер, и при наличии близких объектов фокусные расстояния камер. Структура алгоритмов [39], [40], [46] открывает возможности интеграции с другими подходами и сопутствующими данными, которые могут быть доступны, двумя способами.

1. Для близких объектов первоначальная оценка расстояний может быть выполнена методами дефокусировки или стерео [21], [51], или использованы априорные данные, что дает информацию для выбора приближения и ускоряет сходимость [46]. Может быть выполнено грубое разделение сцены на области различной удаленности, и для них использованы наилучшие приближения. Использование стерео или дефокусировки с совмещенной камерой [6], позволяет произвести выделение быстро движущихся объектов и облегчить их обработку трэкером характеристических точек [45].
2. Построенный меш является хорошим начальным приближением с точно определенными реперными точкам (узлами меша), которое может уточняться различными методами до достижения плотной карты глубин сцены. Метод уточнения выбирается на основе анализа текстур изображения и полученного расстояния от камер до уточняемой области объекта, оцениваемой как расстояние от камеры до поверхности, задаваемой мешем. Для бестекстурных областей работают методы ShapeFromShading, для неоднородных областей – методы [50], причем наличие хорошего начального приближения существенно должно повысить скорость сходимости, для близких к камере областей объекта с мелкой текстурой наиболее перспективным представляется подход [21].

Общая структура системы изображена на рис. 2.1.1, реализованные в настоящий момент блоки отмечены жирной рамкой.

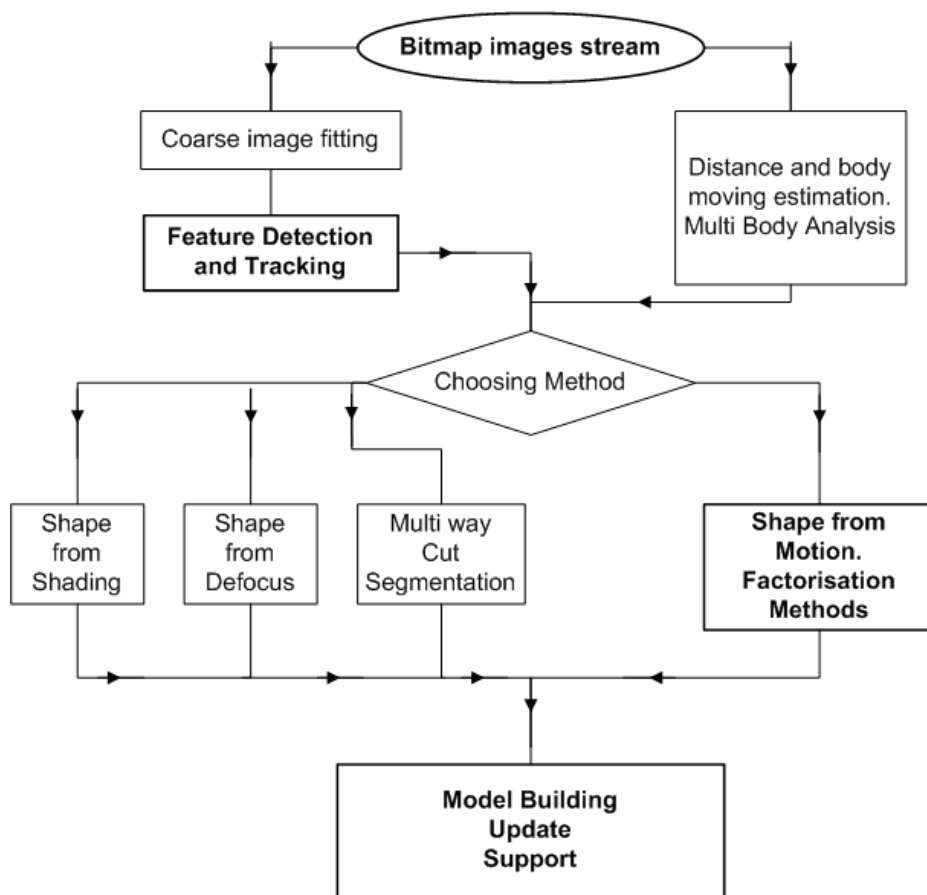


Рис. 2.1.1. Общий вид системы трехмерного зрения.

Особенностью методов факторизации [39], [40], [46] является требование наличия всех сопровождаемых характеристических точек на всех кадрах последовательности, что не всегда возможно, как по причине их потери трэкером, так и из-за заслонения их объектом, например при круговом объезде объекта. Это вызывает необходимость введения в систему блока выбора перекрывающихся поднаборов характеристических точек и кадров, образующих плотные матрицы и блока сшивки фрагментов модели (меша).

Возможна ситуация когда движение камеры носит случайный характер, например съемка с рук или разновременные аэрокосмические изображения. В этом случае требуется предварительное грубое совмещение изображений для обеспечения нормальной работы трэкера (Coarse Image Fitting). Работы в этом направлении еще не завершены.

Основное внимание в этой работе уделено блоку Shape from Motion и методу факторизации матриц. Детализированное представление этой части системы приведено на рис. 2.1.2. Здесь также реализованные блоки выделены жирным контуром. Следует отметить следующие особенности. Блок анализа на множественность объектов в этой схеме подразумевает разделение на независимо движущиеся объекты, в то время как одноименный блок на рис. 2.1.1. отвечает за разделение сцены на объекты по принципу их удаленности или разделенности. Выбор метода предполагается осуществлять на основе дополнительных данных о расстоянии до объекта. Обсуждение этого вопроса приведено в разделе результатов. Предполагается использовать дополнительную информацию о движении камер или объекта для устранения неопределенности в z компоненте в линейных алгоритмах. Работа над этим блоком еще не завершена.

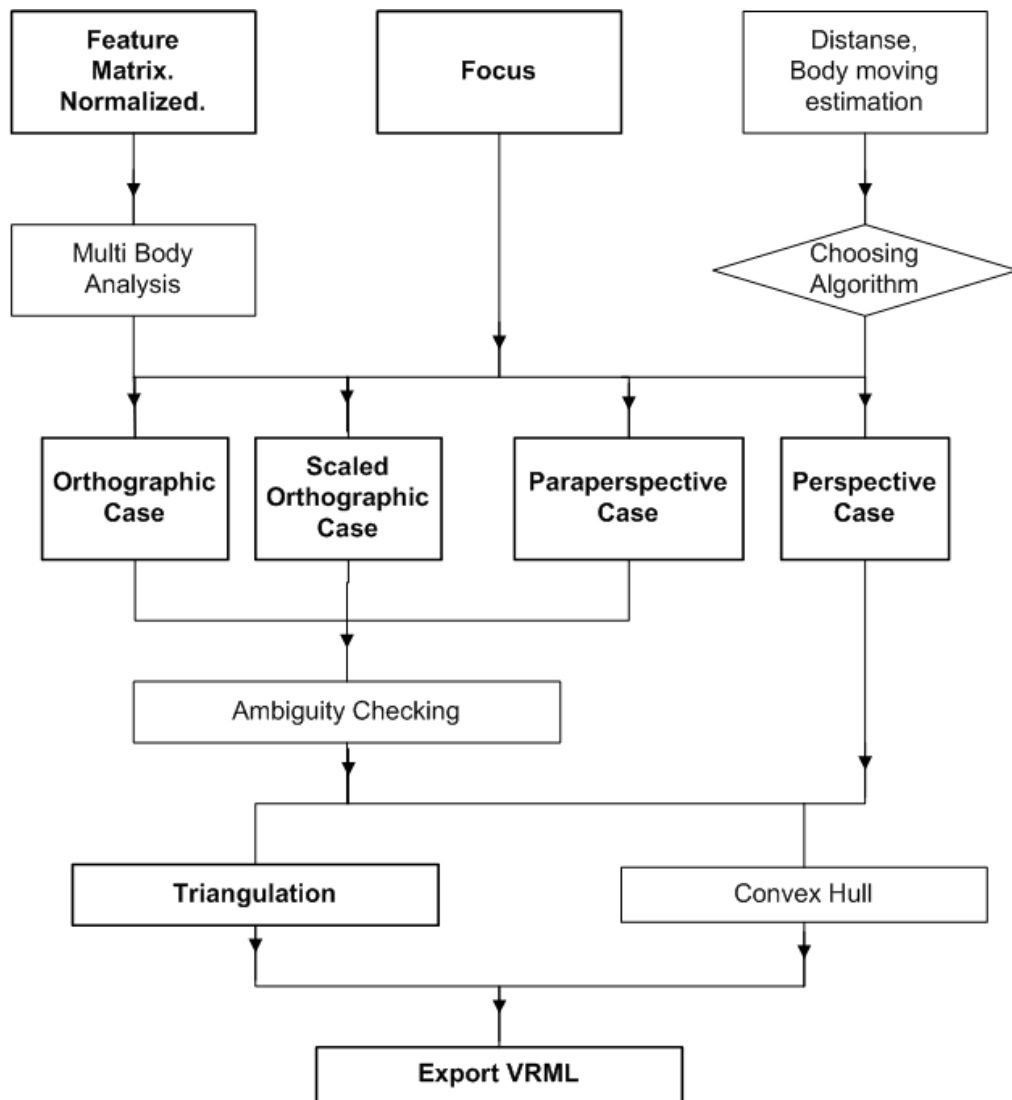


Рис. 2.1.2. Детализированное представление блока Shape from Motion. Factorization Methods.

Система реализована на языке C++ в независимом от операционной системы виде.

2.2. Обнаружение характеристических точек

В качестве детектора и трекера характеристических точек был выбран алгоритм [45], который использует при сопровождении отметок по последовательности кадров информацию корреляционного типа и оптимальный критерий обнаружения отметок с точки зрения надежности сопровождения. Алгоритм [45] был обобщен на случай многоканальных, в частности цветных изображений.

Постановка задачи обнаружения характеристических точек и прослеживания их по последовательности цифровых изображений подразделяется на два пункта:

1. Критерий выбора характеристических точек.
2. Алгоритм прослеживания выбранных точек по последовательности кадров.

Основой концепции решения этой задачи в [45] является то, что две данных подзадачи должны решаться взаимосвязано. При этом основной принцип выбора характеристических точек звучит следующим образом: характеристическими являются те точки, которые могут быть хорошо прослежены.

2.2.1. Сопровождение точек.

По мере движения камеры интенсивность точек изображения меняется сложным образом. Тем не менее изображения, полученные через малые промежутки времени, взаимосвязаны, т.к. относятся к одной и той же сцене, снятой с мало отличающихся точек. Обозначая интенсивность в точке как I , формально можно записать:

$$I(x,y,t+\tau) = I(x - \xi, y - \eta, t); \quad (2.2.1)$$

Другими словами, более поздний снимок, снятый в момент времени $t+\tau$ может быть получен из текущего снимка, снятого в момент времени t , сдвигом всех его точек на некоторую величину. Величина этого сдвига $\mathbf{d} = (\xi, \eta)$ называется смещением точки $\mathbf{x} = (x, y)$ между моментами времени t и $t+\tau$, и фактически, является функцией x, y, t и τ . Таким образом, задача сводится к поиску \mathbf{d} .

Проблемой нахождения смещения \mathbf{d} является тот факт, что невозможно проследить отдельный пиксель, если его интенсивность практически не отличается от интенсивностей его соседей. Кроме того, интенсивность пикселя может меняться из-за шума и смешиваться с ближайшими пикселями. Таким образом, зачастую невозможно определить сдвиг пикселя, основываясь лишь на локальной информации. Решением этой проблемы является прослеживание не отдельных пикселей, а окон пикселей. Тогда происходит прослеживание окон пикселей со сходной текстурой. Сложностью этого подхода является то, что точки выбранного окна пикселей могут двигаться с разной скоростью от изображения к изображению. Такие ситуации требуют описания движения с помощью сложных трансформаций, таких как аффинные преобразования и здесь рассматриваться не будут.

Введем обозначения: $J(\mathbf{x}) = I(x, y, t+\tau)$, $I(\mathbf{x}-\mathbf{d}) = I(x-\xi, y-\eta, t)$, где переменная может быть опущена для простоты. Тогда

$$J(\mathbf{x}) = I(\mathbf{x}-\mathbf{d}) + n(\mathbf{x}),$$

Где $n(\mathbf{x})$ – шум. Тогда вектор смещения \mathbf{d} выбирается таким, чтобы минимизировать добавочный шум, определяемый следующим двойным интегралом по окну W

$$\varepsilon = \int_W [I(\bar{\mathbf{x}} - \bar{\mathbf{d}}) - J(\bar{\mathbf{x}})]^2 \omega d\bar{\mathbf{x}} \quad (2.2.2)$$

Здесь ω - весовая функция. В простейшем случае она может быть положена равной 1. Она также может быть функцией наподобие Гауссова распределения, чтобы выделить центральную часть окна. Или же она может зависеть от распределения интенсивности на изображении.

Т.к. \mathbf{d} полагается малым, можно разложить функцию интенсивности в ряд Тейлора по \mathbf{d} до линейного члена.

$$I(\mathbf{x}-\mathbf{d}) = I(\mathbf{x}) - (\mathbf{g}, \mathbf{d}), \text{ где } \bar{\mathbf{g}} = \left[\frac{\partial I}{\partial x} \quad \frac{\partial I}{\partial y} \right] - \text{градиент яркости.}$$

Тогда выражение (2.2.2) перепишется в виде:

$$\varepsilon = \int_W [I(\bar{x}) - (\bar{\mathbf{g}}, \bar{\mathbf{d}}) - J(\bar{x})]^2 \omega d\bar{x} = \int_W [\bar{h} - (\bar{\mathbf{g}}, \bar{\mathbf{d}})]^2 \omega d\bar{x} \quad (2.2.3)$$

где $\mathbf{h} = I(\mathbf{x}) - J(\mathbf{x})$.

Это выражение является квадратичным относительно \mathbf{d} . Тогда минимизация производится следующим образом. Дифференцируем по \mathbf{d} правую часть выражения (2.2.3) для ε и приравняем результат к нулю:

$$\int_W [\bar{h} - (\bar{\mathbf{g}}, \bar{\mathbf{d}})] \bar{\mathbf{g}} \omega d\bar{x} = 0$$

Т.к. $(\mathbf{g}, \mathbf{d})\mathbf{g} = (\mathbf{g}, \mathbf{g}^T)\mathbf{d}$ и \mathbf{d} полагается постоянным на W имеем:

$$\left(\int_W (\bar{\mathbf{g}}, \bar{\mathbf{g}}^T) \omega d\bar{x} \right) \bar{\mathbf{d}} = \int_W \bar{h} \bar{\mathbf{g}} \omega d\bar{x}$$

Это система из двух уравнений с двумя неизвестными. Ее можно переписать следующим образом:

$$\mathbf{G}\mathbf{d} = \mathbf{e}, \quad (2.2.4)$$

Где матрица коэффициентов – симметричная матрица размера 2×2 :

$$G = \int_W (\bar{\mathbf{g}}, \bar{\mathbf{g}}^T) \omega d\bar{x},$$

а правая часть системы – двумерный вектор

$$\bar{\mathbf{e}} = \int_W (I - J) \bar{\mathbf{g}} \omega d\bar{x}$$

Уравнение (2.2.4) легко обобщается на случай многоканальных изображений. Тогда $\mathbf{I}(\mathbf{x})$ и $\mathbf{J}(\mathbf{x})$ являются вектор - функциями, и интеграл (1) распадается на сумму интегралов:

$$\varepsilon = \iint [J(\bar{x} + \frac{\bar{\mathbf{d}}}{2}) - I(\bar{x} - \frac{\bar{\mathbf{d}}}{2})]^2 d\bar{x} = \sum_{i=1}^N \iint [J_i(\bar{x} + \frac{\bar{\mathbf{d}}}{2}) - I_i(\bar{x} - \frac{\bar{\mathbf{d}}}{2})]^2 d\bar{x}$$

Таким образом, конечное уравнение имеет вид такой же, как и уравнение (2.2.4), но

$$G = \sum_{i=1}^N G_i \text{ и } \bar{\mathbf{e}} = \sum_{i=1}^N \bar{\mathbf{e}}_i$$

, где G_i – матрица G для i -го канала изображения, а \mathbf{e}_i – вектор \mathbf{e} i -го канала изображения.

2.2.2. Обнаружение характеристических точек

Критерий обнаружения характеристических точек в [45], как было упомянуто выше, базируется на методе сопровождения точек. Характеристическим окном пикселей считается то окно, которое может быть хорошо сопровождено. Математически это значит, что матрица коэффициентов системы G должна одновременно быть хорошо определена и превышать уровень шума изображения. Т.е. оба собственных значения матрицы должны быть велики и при этом быть одного порядка величины.

Два малых собственных значения обозначают, что интенсивность в окне меняется слабо. Большое и малое собственные значения получаются при ненаправленном распределении. Два больших собственных значения получаются на ребрах, пестрых текстурах и т.п., где точки могут быть надежно сопровождены.

На практике, если меньшее из собственных значений достаточно велико, чтобы соответствовать шумовому критерию, матрица G бывает также хорошо определена. Это происходит в результате того, что вариация интенсивности в окне ограничена максимально допустимым значением, т.е. большее собственное значение не может сильно отличаться от меньшего.

В результате получаем, что окно содержит характеристическую точку, если собственные значения λ_1 и λ_2 матрицы G удовлетворяют условию:

$$\min(\lambda_1, \lambda_2) > \lambda, \quad (2.2.5)$$

где λ - определенное пороговое значение.

Для определения λ можно воспользоваться следующим методом. Измерить собственные значения для области с постоянной яркостью, снятой с помощью камеры, которую планируется использовать в дальнейшем. Таким образом получим нижнюю границу для λ . Затем аналогичная операция продельвается для различных типов ребер и пестрых структур, что дает верхнюю границу λ . На практике эти границы оказываются сильно разделенными, что позволяет выбрать λ на середине полученного отрезка.

2.3. Формирование плотной матрицы измерений

При сопровождении точек по серии изображений, возможна как потеря, так добавление новых отметок. Однако, для алгоритма восстановления трехмерной сцены требуется наличие всех отслеживаемых точек на всех кадрах. Таким образом, появляется задача выделения наибольшей полностью заполненной матрицы, в которой строки – это кадры, а столбцы – отслеживаемые точки.

Одним из способов решения этой задачи, является нахождение клик определенным образом построенного графа Γ . Кликкой называется множество полностью зависимых вершин графа, которое теряет это свойство при добавлении любой другой вершины.

В условиях данной задачи нас интересует не только максимальная клика, но и все другие для дальнейшего использования их в задаче совмещения восстановленных точек одной и той же сцены, снятых с различных точек зрения, охватывающих все 360^0 обхода.

Граф $\Gamma = \{X, E\}$ строится следующим образом. Множеством его вершин $X = \{F \cup P\}$ являются все кадры F и все отслеживаемые точки P . Множество ребер E задается двумя условиями. Подмножества F и P являются полностью зависимыми множествами. Ребро между вершинами $f \in F$ и $p \in P$ есть только в том случае, если отметка p присутствует в кадре f . (См. рис. 2.3.1)

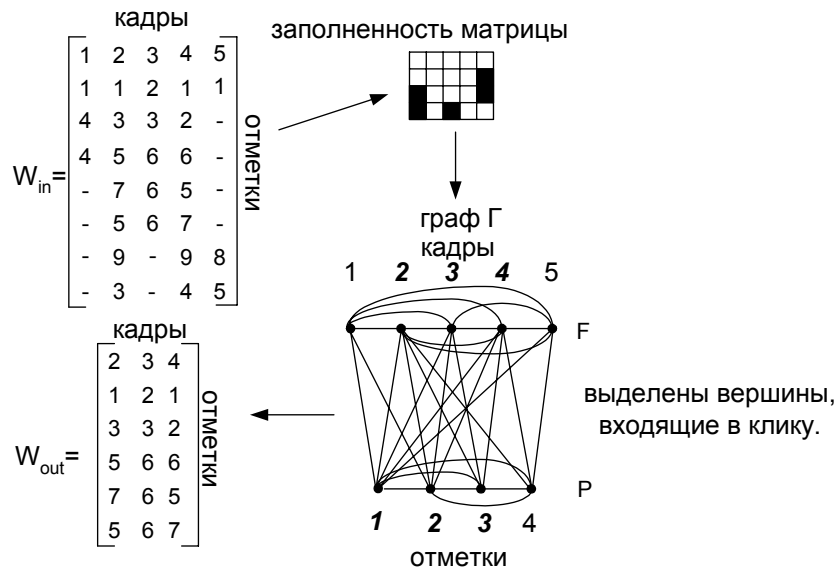


Рис. 2.2.1. Смещение изображения точки в зависимости от степени дефокусировки

Поиск клик в графе, вообще говоря, является задачей не полиномиальной сложности [52, стр 864]. Однако существует ряд приближенных алгоритмов решения этой задачи. Приближенными здесь мы называем алгоритмы, которые ищут за полиномиальное время приближенное к оптимальному, но не оптимальное решение. В данной работе для нахождения клики, используется алгоритм [53].

2.3. Восстановление 3D сцен по последовательности цифровых изображений методом факторизации матриц

2.3.1. Постановка задачи

Рассмотрим систему координат $(\mathbf{i}_f, \mathbf{j}_f, \mathbf{k}_f)$, связанную с камерой, такую, что орт \mathbf{k}_f направлен вдоль оптической оси в направлении наблюдаемой сцены (см. рис.2.3.1).

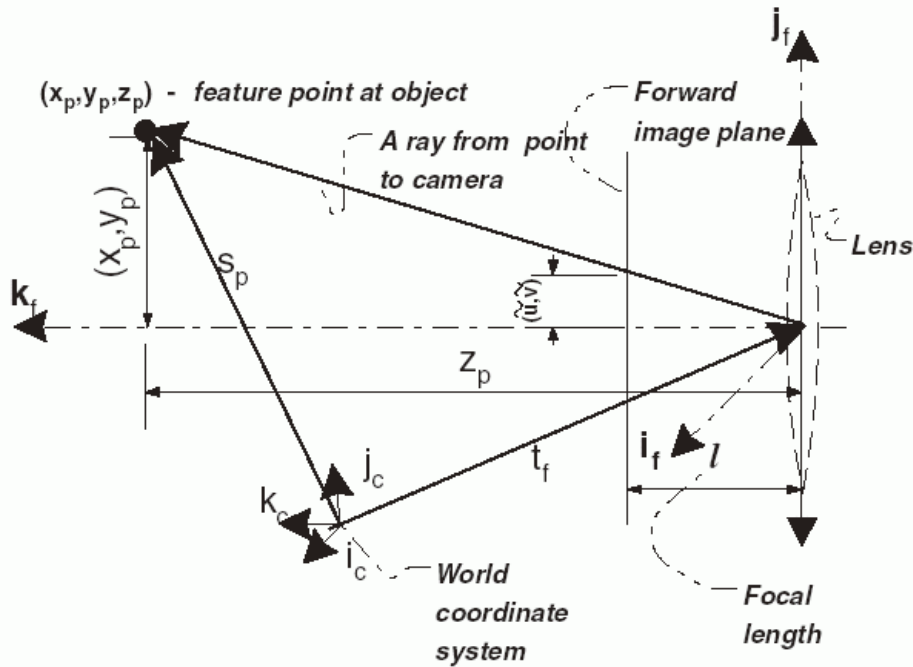


Рис. 2.3.1. Постановка задачи для одной камеры.

Векторы $(\mathbf{i}_f, \mathbf{j}_f, \mathbf{k}_f)$ образуют правую ортонормированную тройку. Отвлекаясь от эффектов, связанных с ограниченной глубиной резкости, характерной для реальных оптических систем, будем считать, что изображения всех точек сцены в плоскости изображения находятся в фокусе, что, в частности, реализуется для камеры Обскура.

Введем понятие передней плоскости изображения (ППИ), которая расположена в плоскости $z=l$, где l - фокусное расстояние объектива. В дальнейшем будем считать, что изображение формируется на ППИ. Это не влияет на описание задачи в рамках геометрической оптики, однако, позволяет избавиться от несущественных знаков и усложнения, связанного с лишним преобразованием системы координат. Орты $\mathbf{i}_f, \mathbf{j}_f$ направлены вдоль строки и столбца пикселей изображения, формируемого в ППИ камеры. Пусть на объекте находится точка \mathbf{p} с координатами (x, y, z) . Координаты ее изображения на ППИ обозначим (\tilde{u}, \tilde{v}) , тогда из подобия треугольников, выделенных на рисунке, можно записать:

$$\begin{cases} \frac{x}{z} = \frac{\tilde{u}}{l} \\ \frac{y}{z} = \frac{\tilde{v}}{l} \end{cases} \quad (2.3.1)$$

Уравнения (2.3.1) устанавливают связь между измеряемыми на опыте значениями \tilde{u}, \tilde{v} , координатами точки объекта в трехмерном пространстве, и фокусным расстоянием камеры.

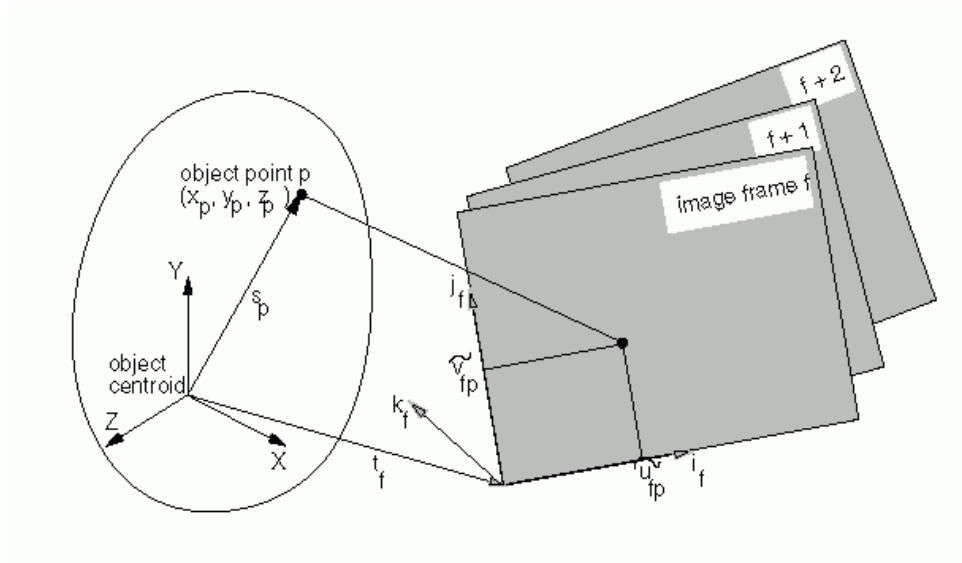


Рис. 2.3.2. Постановка задачи для F камер.

Пусть имеется P точек на объекте, и F камер, или кадров, снятых при различных положениях камеры (см. рис. 2.3.2). Принадлежность величины к точке объекта p (point) будем обозначать индексом p . Индексом f (frame) будем обозначать величины, относящиеся к определенному кадру (камере и ее положению). Тогда уравнения (2.3.1) принимают вид:

$$\begin{cases} x_{fp} = \frac{\tilde{u}_{fp}}{l_f} \\ z_{fp} = \frac{l_f}{z_{fp}} \\ y_{fp} = \frac{\tilde{v}_{fp}}{l_f} \\ z_{fp} = \frac{l_f}{z_{fp}} \end{cases} \quad (2.3.2)$$

Переходя к произвольной системе координат, и записывая (2.3.2) в векторной форме, получим:

$$\begin{cases} \tilde{u}_{fp} = l_f \frac{\vec{i}_f(\vec{s}_p - \vec{t}_f)}{z_{fp}} \\ \tilde{v}_{fp} = l_f \frac{\vec{j}_f(\vec{s}_p - \vec{t}_f)}{z_{fp}}, & \text{где } p=1..P, \\ z_{fp} = \bar{k}_f(\vec{s}_p - \vec{t}_f) & f=1..F. \end{cases} \quad (2.3.3)$$

Уравнения (2.3.3) являются основой для решения задачи восстановления трехмерной формы объекта. Рассмотрим их более подробно. При съемке объекта камерой, измеримыми являются величины $\tilde{u}_{fp}, \tilde{v}_{fp}$ - всего $2FP$ величин. Неизвестными являются $\vec{s}_p, \vec{t}_f, l_f$ - т.е. $3P+3F+F=3P+4F$ величин, и F троек векторов $(\vec{i}_f, \vec{j}_f, \vec{k}_f)$, на которые наложены ограничения ортонормированности:

$$\begin{cases} (\vec{i}_f, \vec{i}_f) = 1 \\ (\vec{j}_f, \vec{j}_f) = 1 \\ (\vec{i}_f, \vec{j}_f) = 0 \\ \bar{k}_f = [\vec{i}_f, \vec{j}_f] \end{cases} \quad (2.3.4)$$

Итого получаем $(6-3)F=3F$ неизвестных величин. Таким образом, $2FP$ уравнений определяют $3P+7F$ неизвестных. Система уравнений (2.3.3) может быть разрешена относительно этих неизвестных в смысле метода наименьших квадратов, если выполняется условие $2FP > 3P+7F$, которое при достаточном количестве точек объекта и снятых кадров всегда справедливо. Минимальными значениями F, P являются следующие пары: $(F=5, P=5), (F=12, P=4), (F=3, P=7)$.

Заметим, что количество неизвестных может быть уменьшено, если заданы дополнительные условия, например постоянство фокусного расстояния (отсутствие zoom-a), уменьшает число неизвестных на $F-1$. Из уравнений (2.3.1) видно, что размеры трехмерной сцены могут быть восстановлены только с точностью до масштабного коэффициента, поэтому для восстановления в абсолютных единицах, требуется знание какого-нибудь размера сцены, например, расстояния между двумя точками на объекте в трёхмерном пространстве, или же расстояния между двумя положениями камеры. Например, если заданы две камеры с одинаковыми фокусными расстояниями l , одинаково сориентированные, так что $(\mathbf{i}_f, \mathbf{j}_f, \mathbf{k}_f)$ совпадают и известны их положения, в частности $\mathbf{t}_1\mathbf{k}_f = \mathbf{t}_2\mathbf{j}_f = 0, \mathbf{t}_1\mathbf{i}_f = -\mathbf{t}_2\mathbf{i}_f = \frac{1}{2}d$, то при $P = 1$ из уравнений (2.3.3) легко получить решение задачи восстановления трехмерных координат сцены на стереопаре изображений с известной базой d .

$$\begin{cases} \frac{1}{z} = \frac{1}{(\bar{k}_f, \bar{s}_p)} = \frac{\tilde{u}_2 - \tilde{u}_1}{l_f d}, \\ \frac{x}{z} = \frac{(\bar{l}_f, \bar{s}_p)}{(\bar{k}_f, \bar{s}_p)} = \frac{\tilde{u}_2 + \tilde{u}_1}{2l_f}. \end{cases} \quad (2.3.5)$$

Форма записи решения (2.3.5) выбрана с учетом того обстоятельства, что по мере увеличения расстояния от камеры до объекта, точность определения z падает, в то же время точность определения направления на объект не зависит от расстояния. Деления на нуль в формулах (2.3.5) при малом z не возникает, так как минимальное расстояние до наблюдаемого объекта всегда ограничено в силу конструкции камеры, а в случае стерео – дополнительным требованием видимости точки объекта одновременно двумя камерами..

В ходе работы с цифровыми камерами, размер изображения в фокальной плоскости (т.е. размер пикселя), как правило, в метрических единицах не известен. Поэтому целесообразно перейти к измеримым величинам, таким как координаты пикселя. Преобразуем уравнения (2.3.2) к виду:

$$\begin{cases} \tilde{u}_{fp} \frac{a}{\tilde{u}_{fp}^e N} = \frac{x_{fp}}{z_{fp}} \frac{l_f a}{\tilde{u}_{fp}^e N}, \\ \tilde{v}_{fp} \frac{a}{\tilde{v}_{fp}^e N} = \frac{y_{fp}}{z_{fp}} \frac{l_f a}{\tilde{v}_{fp}^e N}, \end{cases} \quad (2.3.6)$$

где a - положительный масштабный коэффициент, определяющий диапазон изменения величин; $\tilde{u}_{fp}^e, \tilde{v}_{fp}^e$ - размеры одного пикселя на фоточувствительной матрице в метрических единицах; $N = \max(N_x, N_y)$, где N_x, N_y - количество пикселей на изображении (разрешение по x, y соответственно); обычно, $N_x > N_y$. Перейдём к новым переменным:

$$\left\{ \begin{array}{l} u_{fp} = \frac{a}{N} \frac{\tilde{u}_{fp}}{\tilde{u}_{fp}^e} = a \frac{n_x}{N}, \\ v_{fp} = \frac{a}{N} \frac{\tilde{v}_{fp}}{\tilde{v}_{fp}^e} = a \frac{n_y}{N}, \\ g_f = a \frac{l_f}{\tilde{u}_{fp}^e N} = \frac{1}{\gamma}, \\ \alpha = \frac{\tilde{u}_{fp}^e}{\tilde{v}_{fp}^e}, \end{array} \right. \quad (2.3.7)$$

где α - отношение разрешений по x и y (aspect ratio); величины $\frac{n_x}{N}$, $\frac{n_y}{N}$ - это координаты пикселя, (вычисляемые как расстояние от центра кадра), делённые на ширину изображения, - всегда в диапазоне $\left[-\frac{a}{2}, \frac{a}{2}\right]$; $\gamma = tg \frac{\alpha_{\max}}{2}$, где α_{\max} - максимальный угол зрения камеры. В новых обозначениях уравнения (2.3.6) можно переписать в виде:

$$\left\{ \begin{array}{l} u_{fp} = g_f \frac{x_{fp}}{z_{fp}}, \\ v_{fp} = \alpha g_f \frac{y_{fp}}{z_{fp}}. \end{array} \right. \quad (2.3.8)$$

Далее будем полагать $\alpha=1$. Значение нормировочного коэффициента a может быть существенным только для устойчивости и точности численных методов.

Перейдём к мировой системе координат (МСК). В ней координаты точки p задаются вектором \mathbf{s}_p , позиции камеры f - вектором \mathbf{t}_f , а её ориентация, - координатными ортами $(\mathbf{i}_f, \mathbf{j}_f, \mathbf{k}_f)$, причём $\mathbf{k}_f = [\mathbf{i}_f, \mathbf{j}_f]$. В итоге уравнения (2.3.8) приобретают вид:

$$\left\{ \begin{array}{l} u_{fp} = g_f \frac{\vec{i}_f(\vec{s}_p - \vec{t}_f)}{z_{fp}}, \\ v_{fp} = g_f \frac{\vec{j}_f(\vec{s}_p - \vec{t}_f)}{z_{fp}}, \\ z_{fp} = \vec{k}_f(\vec{s}_p - \vec{t}_f). \end{array} \right. \quad (2.3.9)$$

2.3.2. Математический вывод приближений

Выберем начало МСК в центре масс (ЦМ) точек объекта:

$$\sum_{p=1}^P \vec{s}_p = 0 \quad (2.3.10)$$

Тогда z-координата центра масс на данном изображении в МСК:

$$\frac{1}{P} \sum_{p=1}^P z_{fp} = \frac{\bar{k}_f}{P} \sum_{p=1}^P \vec{s}_p - \bar{k}_f \vec{t}_f = -\bar{k}_f \vec{t}_f = z_f \quad (2.3.11)$$

И z-координата точки p выражается следующим образом

$$z_{fp} = z_f + \bar{k}_f \vec{s}_p = z_f \left(1 + \frac{\bar{k}_f \vec{s}_p}{z_f} \right). \quad (2.3.12)$$

Выражение (2.3.9) перепишем в следующем виде, используя (2.3.12):

$$\begin{cases} u_{fp} = g_f \frac{\vec{i}_f (\vec{s}_p - \vec{t}_f)}{z_f \left(1 + \frac{\bar{k}_f \vec{s}_p}{z_f} \right)}, \\ v_{fp} = g_f \frac{\vec{j}_f (\vec{s}_p - \vec{t}_f)}{z_f \left(1 + \frac{\bar{k}_f \vec{s}_p}{z_f} \right)}, \end{cases} \quad (2.3.13)$$

и разложим в ряд Тейлора с точностью до членов второго порядка малости по $\frac{\bar{k}_f \vec{s}_p}{z_f}$, переобозначив сумму на

σ :

$$\begin{cases} u_{fp} = g_f \frac{(\vec{i}_f, (\vec{s}_p - \vec{t}_f))}{z_f} \left(1 - \frac{\bar{k}_f \vec{s}_p}{z_f} + \left(\frac{\bar{k}_f \vec{s}_p}{z_f} \right)^2 + \dots \right) = g_f \frac{(\vec{i}_f, (\vec{s}_p - \vec{t}_f))}{z_f} \sigma, \\ v_{fp} = g_f \frac{(\vec{j}_f, (\vec{s}_p - \vec{t}_f))}{z_f} \left(1 - \frac{\bar{k}_f \vec{s}_p}{z_f} + \left(\frac{\bar{k}_f \vec{s}_p}{z_f} \right)^2 + \dots \right) = g_f \frac{(\vec{j}_f, (\vec{s}_p - \vec{t}_f))}{z_f} \sigma. \end{cases} \quad (2.3.14)$$

Теперь, собирая члены при \vec{s}_f и \vec{t}_f :

$$\begin{cases} u_{fp} = g_f \left((\vec{i}_f, \vec{s}_p) \frac{\sigma}{z_f} - (\vec{i}_f, \vec{t}_f) \frac{\sigma}{z_f} \right), \\ v_{fp} = g_f \left((\vec{j}_f, \vec{s}_p) \frac{\sigma}{z_f} - (\vec{j}_f, \vec{t}_f) \frac{\sigma}{z_f} \right). \end{cases} \quad (2.3.15)$$

Пусть $|s_p| \ll z_f$ и $|t_f| \approx z_f$, тогда $z_f = z$ и $g_f = g$ не зависят от номера кадра f , и, вводя обозначения $s_p' = s_p \frac{g}{z}$ и

$t_p' = t_p \frac{g}{z}$, можно записать:

$$\begin{cases} u_{fp} = \vec{i}_f (s_p' - t_f'), \\ v_{fp} = \vec{j}_f (s_p' - t_f'). \end{cases} \quad (2.3.16)$$

Заметим, что нормирующий множитель $\frac{g}{z}$ не существен, поскольку, как отмечалось в разделе 2.3.1, восстановление происходит с точностью до масштабного коэффициента. Выражение (2.3.16) соответствует приближению ортографической проекции (ОП)⁷ [41].

Ограничиваясь только предположением $|s_p| \ll z_f$, и заменяя в (2.3.9) z_{fp} на z_f , что эквивалентно отбрасыванию в (2.3.14) всех слагаемых в сумме, кроме первого, приходим к приближению масштабируемой ортографической проекции (МОП)⁸ [39]:

$$\begin{cases} u_{fp} = \frac{\vec{i}_f (\vec{s}_p - \vec{t}_f)}{z_f'}, \\ v_{fp} = \frac{\vec{j}_f (\vec{s}_p - \vec{t}_f)}{z_f'}, \\ z_f' = \frac{z_f}{g_f} = -\frac{(\vec{k}_f, \vec{t}_f)}{g_f}. \end{cases} \quad (2.3.17)$$

Оставляя в разложении (2.3.15) только члены, линейные по s_p :

$$\begin{cases} u_{fp} = g_f \left((\vec{i}_f, \vec{s}_p) \frac{1}{z_f} - (\vec{i}_f, \vec{t}_f) \left(\frac{1}{z_f} - \frac{\vec{k}_f \vec{s}_p}{z_f^2} \right) \right), \\ v_{fp} = g_f \left((\vec{j}_f, \vec{s}_p) \frac{1}{z_f} - (\vec{j}_f, \vec{t}_f) \left(\frac{1}{z_f} - \frac{\vec{k}_f \vec{s}_p}{z_f^2} \right) \right). \end{cases} \quad (2.3.18)$$

получим приближение параперспективной проекции (ППП)⁹ [39]:

$$\begin{cases} u_{fp} = \frac{1}{z_f'} \left[\left(\vec{i}_f + \frac{(\vec{i}_f, \vec{t}_f)}{g z_f'} \vec{k}_f \right) \vec{s}_p - (\vec{i}_f, \vec{t}_f) \right], \\ v_{fp} = \frac{1}{z_f'} \left[\left(\vec{j}_f + \frac{(\vec{j}_f, \vec{t}_f)}{g z_f'} \vec{k}_f \right) \vec{s}_p - (\vec{j}_f, \vec{t}_f) \right], \\ z_f' = \frac{z_f}{g_f} = -\frac{(\vec{k}_f, \vec{t}_f)}{g_f}. \end{cases} \quad 2.3.19$$

⁷ Orthographic Projection

⁸ Scaled orthographic projection

⁹ Paraperspective Projection

Введём новые обозначения:

Таблица 2.3.1.

ОП	МОП	ППП
$x_f = -(\vec{i}_f, \vec{t}_f')$	$x_f = -\frac{(\vec{i}_f, \vec{t}_f)}{z_f'}$	$x_f = -\frac{(\vec{i}_f, \vec{t}_f)}{z_f'}$
$y_f = -(\vec{j}_f, \vec{t}_f')$	$y_f = -\frac{(\vec{j}_f, \vec{t}_f)}{z_f'}$	$y_f = -\frac{(\vec{j}_f, \vec{t}_f)}{z_f'}$
$z_f' = \frac{-(\vec{k}_f, \vec{t}_f)}{g} = \frac{z}{g} \approx 1$	$z_f' = \frac{-(\vec{k}_f, \vec{t}_f)}{g_f}$	$z_f' = \frac{-(\vec{k}_f, \vec{t}_f)}{g_f}$
$\vec{m}_f = \frac{\vec{i}_f}{z_f'}$	$\vec{m}_f = \frac{\vec{i}_f}{z_f'}$	$\vec{m}_f = \frac{1}{z_f'} \left(\vec{i}_f - \frac{x_f}{g} \vec{k}_f \right)$
$\vec{n}_f = \frac{\vec{j}_f}{z_f'}$	$\vec{n}_f = \frac{\vec{j}_f}{z_f'}$	$\vec{n}_f = \frac{1}{z_f'} \left(\vec{j}_f - \frac{y_f}{g} \vec{k}_f \right)$

Используя Таблицу 2.3.1, все три приведенных приближения можно привести к единому математическому виду:

$$\begin{cases} u_{fp} = \vec{m}_f \vec{s}_p + x_f \\ v_{fp} = \vec{n}_f \vec{s}_p + y_f \end{cases} \quad (2.3.20)$$

2.3.3. Геометрический смысл приближений

Ранее мы строго математически ввели несколько приближений к перспективной проекции. Для того, чтобы осветить геометрический смысл уравнений, описывающих различные приближений, рассмотрим подробнее геометрическую модель упомянутых приближений.

2.3.3.1. Приближение ортографической проекции

В ортографии каждая точка объекта проецируется на плоскость изображения параллельно оптической оси камеры (см. рис. 2.3.3).

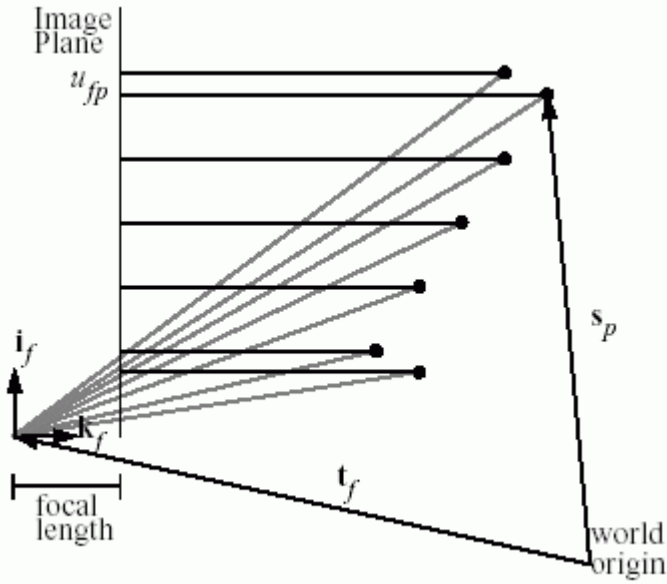


Рис. 2.3.3. Ортографическая проекция в двумерном пространстве

Из геометрии рис. 2.3.3 можно непосредственно получить выведенные нами ранее формулы 2.3.16:

$$\begin{cases} u_{fp} = \vec{i}_f (s_p - t_f), \\ v_{fp} = \vec{j}_f (s_p - t_f). \end{cases}$$

Таким образом, ОП - приближение описывается линейными уравнениями. С одной стороны, это делает его привлекательным, но, в то же время, существенно ограничивает его применимость т.к. с его помощью невозможно восстановить расстояние до объекта. Кроме того, возникает неоднозначность в определении координаты z точек объекта, связанная с "уплощением" объекта в плоскости изображения. Поэтому в рамках ОП - приближения принципиально нельзя восстановить, выпуклый объект, или вогнутый.

2.3.3.2. Приближение масштабируемой ортографической проекции

В приближении масштабируемой ортографической проекции¹⁰ (рис. 2.3.4) все точки объекта сначала проецируются параллельно оптической оси камеры на воображаемую плоскость, проходящую через ЦМ точек объекта (на рис. 2.3.4 ЦМ совмещён с началом МСК). Понятно, что и в этом случае имеет место "уплощение" объекта, а следовательно, и неоднозначность в восстановлении знака глубины сцены.

¹⁰ Иногда она упоминается в литературе как Weak Perspective проекция

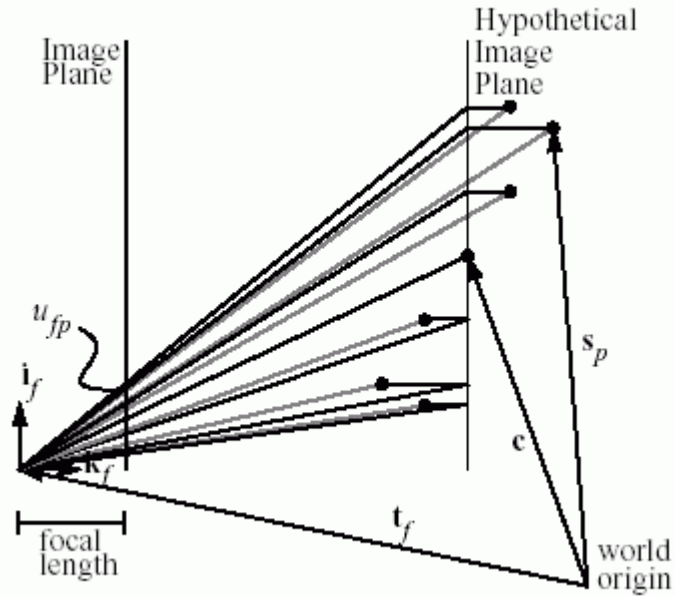


Рис. 2.3.4. Масштабируемая ортографическая проекция в двумерном пространстве

Из этой модели получаем систему (2.3.17):

$$\begin{cases} u_{fp} = \frac{\vec{i}_f(\vec{s}_p - \vec{t}_f)}{z_f'}, \\ v_{fp} = \frac{\vec{j}_f(\vec{s}_p - \vec{t}_f)}{z_f'}, \\ z_f' = \frac{z_f}{g_f} = -\frac{(\vec{k}_f, \vec{t}_f)}{g_f}. \end{cases}$$

Благодаря введению в систему величины z_f , метод восстановления трёхмерных координат точек объекта, основанный на МОП - приближении, работает объективно лучше, чем метод с применением ортографической проекции, поскольку появилась возможность восстановить не только форму объекта, но и расстояние до него. Это обусловлено тем фактом, что линейный масштаб изображения изменяется обратно

пропорционально расстоянию z : $u_{fp}, v_{fp} \propto \frac{1}{z_f}$

2.3.3.3. Приближение параперспективной проекции

Построение, соответствующее приближению параперспективной проекции, отличается от построения в МОП - приближении только направлением проецирования на воображаемую плоскость, в данном случае оно происходит параллельно направлению на ЦМ точек объекта (рис. 2.3.5). Ранее было показано, что ППП - приближение соответствует более аккуратному разложению $1/z_{fp}$ в ряд Тейлора, то есть этот вариант линейного приближения даст лучшую точность восстановления формы объекта, чем все рассмотренные ранее.

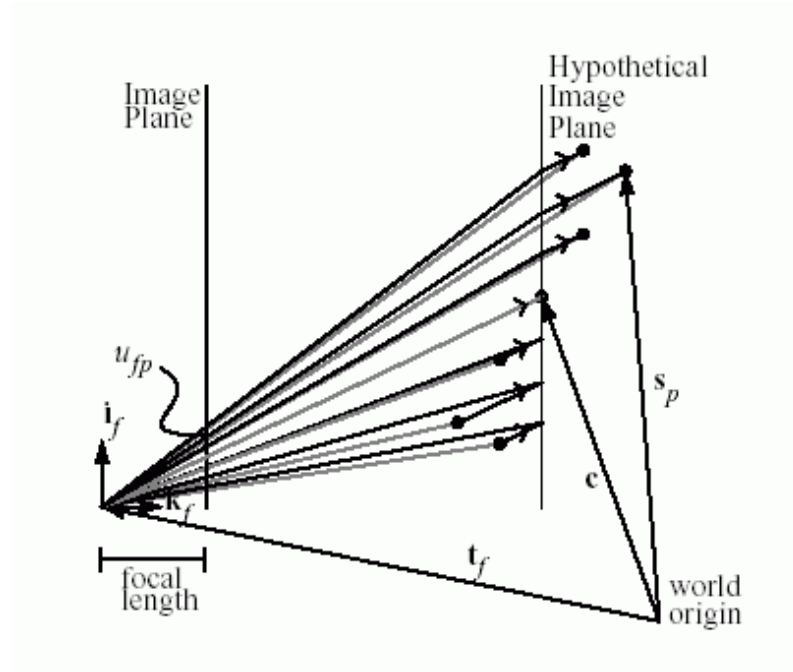


Рис. 2.3.5. Параллельная проекция в двумерном пространстве.

Из аналитической геометрии известно, что проекция точки, заданной вектором \vec{p} на плоскость с нормалью \vec{n} , в направлении \vec{r} , описывается следующим уравнением:

$$\vec{p}' = \vec{p} - \frac{(\vec{p}, \vec{n}) - d}{(\vec{r}, \vec{n})} \vec{r} \quad (2.3.21)$$

здесь d - расстояние до плоскости из начала СК. На изображении f , каждая точка объекта s_p проецируется в направлении вектора $(\vec{c} - \vec{t}_f)$ соединяющего ЦМ точек объекта с началом КСК, на плоскость, которую мы зададим нормалью \vec{k}_f , и расстояние из начала СК $d = (\vec{c}, \vec{k}_f)$, здесь \vec{c} - луч, соединяющий начало МСК с ЦМ точек объекта. Воспользовавшись уравнением (2.3.21), получаем:

$$\vec{s}_{fp}' = \vec{s}_p - \frac{((\vec{s}_p - \vec{t}_f), \vec{k}_f) - ((\vec{c} - \vec{t}_f), \vec{k}_f)}{((\vec{c} - \vec{t}_f), \vec{k}_f)} (\vec{c} - \vec{t}_f) \quad (2.3.22)$$

Подставляя полученные \vec{s}_{fp}' в уравнения для приближения масштабируемой проекции (2.3.17), получим:

$$\begin{cases} u_{fp} = \frac{1}{z_f'} \left[\left(\vec{i}_f - \frac{(\vec{i}_f, (\vec{c} - \vec{t}_f))}{gz_f'} \vec{k}_f \right) (\vec{s}_p - \vec{c}) - (\vec{i}_f, (\vec{c} - \vec{t}_f)) \right], \\ v_{fp} = \frac{1}{z_f'} \left[\left(\vec{j}_f - \frac{(\vec{j}_f, (\vec{c} - \vec{t}_f))}{gz_f'} \vec{k}_f \right) (\vec{s}_p - \vec{c}) - (\vec{j}_f, (\vec{c} - \vec{t}_f)) \right], \\ z_f' = \frac{z_f}{g_f} = - \frac{(\vec{k}_f, \vec{t}_f)}{g_f}. \end{cases} \quad (2.3.23)$$

Поместим начало МСК в ЦМ точек объекта:

$$c = \frac{1}{P} \sum_{p=1}^P \bar{s}_p. \quad (2.3.24)$$

С учетом (2.3.24), система (2.3.23) приобретает вид (2.3.19):

$$\left\{ \begin{array}{l} u_{fp} = \frac{1}{z_f'} \left[\left(\bar{i}_f + \frac{(\bar{i}_f, \bar{l}_f)}{g z_f'} \bar{k}_f \right) \bar{s}_p - (\bar{i}_f, \bar{l}_f) \right], \\ v_{fp} = \frac{1}{z_f'} \left[\left(\bar{j}_f + \frac{(\bar{j}_f, \bar{l}_f)}{g z_f'} \bar{k}_f \right) \bar{s}_p - (\bar{j}_f, \bar{l}_f) \right], \\ z_f' = \frac{z_f}{g_f} = - \frac{(\bar{k}_f, \bar{l}_f)}{g_f}. \end{array} \right.$$

Для случая параперспективного приближения, ограничения на орты КСК, выглядят следующим образом:

$$\left\{ \begin{array}{l} \bar{i}_f = z_f' \bar{m}_f + \frac{x_f}{g_f} \bar{k}_f, \\ \bar{j}_f = z_f' \bar{n}_f + \frac{y_f}{g_f} \bar{k}_f, \\ |\bar{i}_f| = 1, \\ |\bar{j}_f| = 1, \\ \bar{k}_f = [\bar{i}_f, \bar{j}_f]. \end{array} \right. \quad (2.3.25)$$

Переписывая (2.3.25) в матричной форме, получаем:

$$G_f \mathbf{k}_f = H_f \quad (2.3.26)$$

где

$$G_f = \begin{pmatrix} [\bar{m}_f, \bar{n}_f] \\ \bar{m}_f \\ \bar{n}_f \end{pmatrix} \quad H_f = \begin{pmatrix} 1 \\ -\frac{x_f}{g_f} \\ -\frac{y_f}{g_f} \end{pmatrix}, \quad (2.3.27)$$

$$\begin{cases} \tilde{m}_f = \sqrt{1 + \left(\frac{x_f}{g_f}\right)^2} \frac{\bar{m}_f}{|\bar{m}_f|}, \\ \tilde{n}_f = \sqrt{1 + \left(\frac{y_f}{g_f}\right)^2} \frac{\bar{n}_f}{|\bar{n}_f|}. \end{cases} \quad (2.3.28)$$

В случае нецентрального наведения камеры на объект, ППП - метод восстановления даст наилучшие результаты, поскольку в нём учтена существенная непараллельность лучей, соединяющих выбранные точки сцены и точку пересечения фокальной плоскости с оптической осью камеры.

2.3.4 Перспективная проекция

В модели перспективной проекции (ПП)¹¹ лучи идут именно так, как это происходит в реальности, то есть по прямой (см. рис. 2.3.6).

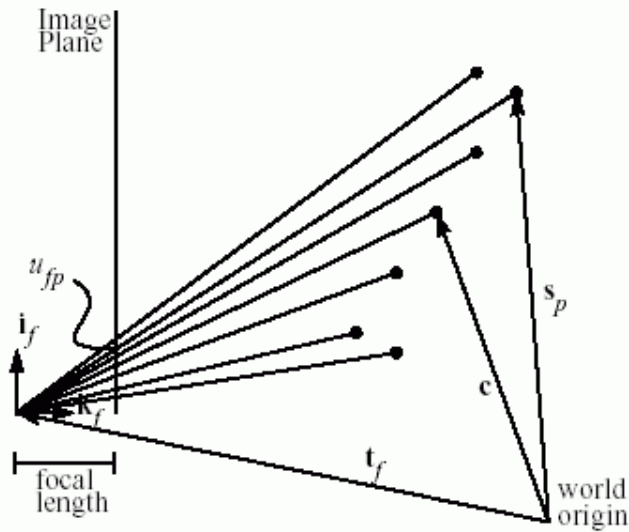


Рис. 2.3.6. Перспективная проекция в двумерном пространстве

Уравнения, описывающие перспективную проекцию, существенно нелинейные по переменным движения и формы. В принятой в настоящем разделе системе обозначений, они имеют вид:

$$\begin{cases} u_{fp} = g_f \frac{\vec{i}_f(\vec{s}_p - \vec{t}_f)}{\vec{k}_f(\vec{s}_p - \vec{t}_f)}, \\ v_{fp} = g_f \frac{\vec{j}_f(\vec{s}_p - \vec{t}_f)}{\vec{k}_f(\vec{s}_p - \vec{t}_f)}, \end{cases} \Rightarrow \begin{cases} u_{fp} = g_f \frac{(\vec{i}_f, \vec{s}_p) + x_f}{(\vec{k}_f, \vec{s}_p) + z_f}, \\ v_{fp} = g_f \frac{(\vec{j}_f, \vec{s}_p) + y_f}{(\vec{k}_f, \vec{s}_p) + z_f}, \end{cases} \quad (2.3.29)$$

где

$$\begin{cases} x_f = -\vec{i}_f \vec{t}_f \\ y_f = -\vec{j}_f \vec{t}_f \\ z_f = -\vec{k}_f \vec{t}_f \end{cases} \quad (2.3.30)$$

В настоящий момент существует несколько методов решения точной задачи. Все они сводятся к переопределённой нелинейной задаче наименьших квадратов в переменных движения и формы. В качестве иллюстрации, рассмотрим метод решения точной задачи, предложенный в[39].

Предлагается минимизировать ошибку:

$$\varepsilon = \sum_{f=1}^F \sum_{p=1}^P \left\{ \left(u_{fp} - g_f \frac{\vec{i}_f \vec{s}_p + x_f}{\vec{k}_f \vec{s}_p + z_f} \right)^2 + \left(v_{fp} - g_f \frac{\vec{j}_f \vec{s}_p + y_f}{\vec{k}_f \vec{s}_p + z_f} \right)^2 \right\} \quad (2.3.31)$$

¹¹ Perspective Projection

Из ортов КСК формируется матрица вращения, как функция трёх углов α_f , β_f , γ_f :

$$R = \begin{pmatrix} \vec{i}_f & \vec{j}_f & \vec{k}_f \end{pmatrix} = \begin{pmatrix} \cos \alpha_f \cos \beta_f & (\cos \alpha_f \sin \beta_f \sin \gamma_f - \sin \alpha_f \cos \gamma_f) & (\cos \alpha_f \sin \beta_f \cos \gamma_f - \sin \alpha_f \sin \gamma_f) \\ \sin \alpha_f \cos \beta_f & (\sin \alpha_f \sin \beta_f \sin \gamma_f - \cos \alpha_f \cos \gamma_f) & (\sin \alpha_f \sin \beta_f \cos \gamma_f - \cos \alpha_f \sin \gamma_f) \\ \sin \beta_f & \cos \beta_f \sin \gamma_f & (\cos \beta_f \cos \gamma_f) \end{pmatrix} \quad (2.3.32)$$

Таким образом, мы имеем 6 параметров, отвечающих за движение камеры: x_f , y_f , z_f , α_f , β_f , γ_f , и 3 параметра, характеризующие форму сцены: $s_p = [s_{p1}, s_{p2}, s_{p3}]$. Уравнения (2.3.29) определяют переопределённую систему 2FP уравнений для 6F+3P неизвестных, и мы можем минимизировать ошибку (2.3.31), учитывая (2.3.32). Задаваясь начальными значениями переменных, в методе предлагается итерационно уточнять их. Однако, для того, чтобы пользоваться техникой решения нелинейных задач методом наименьших квадратов, необходимо на каждом шагу итерации создавать и обращать матрицу размером (6F+3P)×(6F+3P). При достаточно больших F, P, задача минимизации будет просто неразрешима в силу слишком большого числа переменных. Поэтому предлагается использовать специфическую структуру уравнений, решая их отдельно для переменных движения и формы. Сначала мы фиксируем переменные формы, и находим решение уравнений, минимизирующее ошибку (т.е. предлагается решать F систем уравнений с 6 неизвестными). При этом, минимизация производится независимо по каждой переменной движения. Затем фиксируем переменные движения, и минимизируем ошибку по переменным формы (т.е., решаем P систем с тремя неизвестными). Итерации необходимо производить до тех пор, пока ε не достигнет своего минимального значения. Таким образом, сложность задачи была существенно понижена, и она стала разрешимой.

2.3.5. Сравнительный анализ приближений

Понятно, что все рассмотренные методы находят своё применение, и отказаться от какого-то из них, в пользу другого, невозможно. Зачастую необходимо параллельно использовать все методы. Поэтому нужно разрабатывать, и применять гибридный алгоритм, который, в зависимости от внешних параметров, использовал бы то или иное приближение.

В случае, когда расстояние до объекта велико по сравнению с его размером и перемещениями камеры, возможные различия в изображениях, обусловлены только вращениями объекта. При этом, отсутствуют изменения видимого размера объекта и перспективные искажения, поэтому принципиально невозможно восстановление расстояния до объекта. Это приводит к плохой обусловленности уравнений для приближений МОП и ППП, поэтому необходимо применять ортографию. Отметим здесь очевидный вырожденный случай: если объект не вращается, то есть, повернут к наблюдателю всегда одной и той же стороной, тогда, необходимая для восстановления формы, информация отсутствует, и применение ОП ничего не даст. Все изображения сцены будут одинаковыми, и, следовательно, уравнения не будут зависеть от индекса f . Примером, хорошо иллюстрирующим данную ситуацию, может служить невозможность восстановления рассмотренными методами, формы поверхности Луны по наблюдениям с Земли. Сопоставимость перемещения камеры с расстоянием до объекта в направлении на него, приводит к измеримым изменениям размера изображения объекта, что и даёт дополнительную возможность восстановления расстояния до сцены в рамках МОП- и ППП-методов. Пока этот эффект не велик, достаточно использовать МОП приближение, избегая лишних вычислительных сложностей, учитывая, что точность восстановления расстояния принципиально невелика. По мере приближения к объекту, погрешности МОП метода становятся заметными, и следует переходить к ППП приближению. Существенным недостатком рассмотренных выше линейных методов (ПОП, МОП, ППП), является принципиальная невозможность восстановить, выпуклый объект, или нет¹².

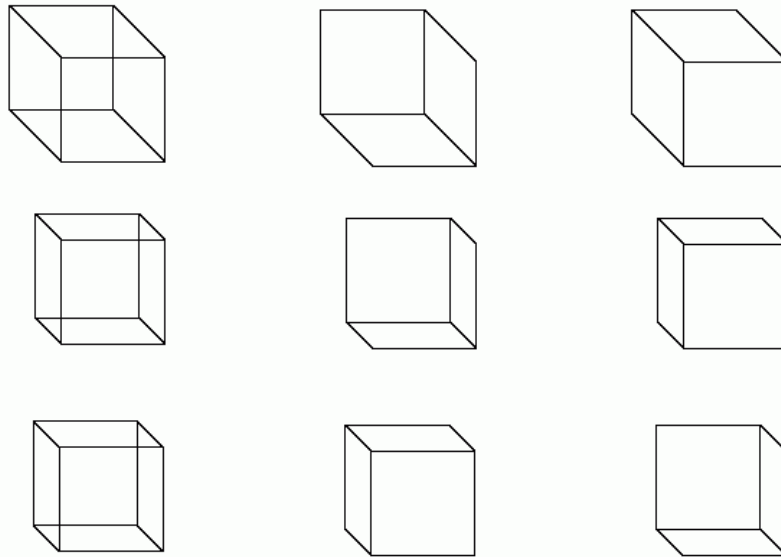


Рис. 2.3.7. Неоднозначность интерпретации формы трехмерного объекта.

С этой задачей справляется только перспективный метод. Причина проблемы заложена в физике рассматриваемых приближений: вместо объёмного представления объекта, мы работаем с плоской моделью.

При дальнейшем приближении к сцене, становятся заметными перспективные искажения. Это приводит к росту погрешностей при восстановлении формы объекта в виде его искривления. Однако, наличие таких искажений даёт дополнительную информацию, позволяющую, например, различать выпуклые и вогнутые объекты (становится возможным устранение неоднозначности восстановления формы по z), и восстановить фокусное расстояние камер. Здесь необходимо использовать модель перспективной проекции.

¹² В литературе часто упоминается как проблема removal ambiguity.

2.3.6. Метод факторизации матриц

Одним из наиболее эффективных методов восстановления 3D формы и движения объекта является метод, основанный на факторизации матриц. Поясним суть метода для линейных приближений. Выше было показано, что уравнения, описывающие линейные приближения, имеют вид (2.3.20) :

$$\begin{cases} u_{fp} = \bar{m}_f \bar{s}_p + x_f \\ v_{fp} = \bar{n}_f \bar{s}_p + y_f \end{cases}$$

Сформируем из координат $\{(u_{fp}, v_{fp}) : f=1..F, p=1..P\}$ матрицу измерений W размера $2F \times P$:

$$W = \begin{pmatrix} u_{11} & u_{12} & \cdots & u_{1P} \\ v_{11} & v_{12} & \cdots & v_{1P} \\ \vdots & \vdots & \cdots & \vdots \\ u_{F1} & u_{F2} & \cdots & u_{FP} \\ v_{F1} & v_{F2} & \cdots & v_{FP} \end{pmatrix} \quad (2.3.33)$$

Каждая строка матрицы W содержит координаты точек (u_{fp}, v_{fp}) , относящиеся к определённому кадру последовательности, а каждый столбец, - те же величины для конкретной точки, присутствующей на всех изображениях. Тогда выражение (2.3.33) можно переписать в матричной форме:

$$W = MS + T \underbrace{[1 \dots 1]}_P \quad (2.3.34)$$

где M - матрица движения (ориентации камер) размера $2F \times 3$, S - матрица формы (точки на объекте), размера $3 \times P$, T - вектор трансляций камер (положение камер в пространстве), размера $2F \times 1$, формируемые следующим образом:

$$M = \begin{pmatrix} \bar{m}_1^T \\ \bar{n}_1^T \\ \vdots \\ \bar{m}_F^T \\ \bar{n}_F^T \end{pmatrix}, \quad S = (\bar{s}_1 \quad \dots \quad \bar{s}_P) \quad T = \begin{pmatrix} x_1 \\ y_1 \\ \vdots \\ x_F \\ y_F \end{pmatrix} \quad (2.3.35)$$

Поместим начало МСК в ЦМ точек объекта (2.3.10):

$$\frac{1}{P} \sum_{p=1}^P \bar{s}_p = 0$$

Заметим, что вектор T можно получить построением суммированием элементов матрицы W :

$$T = \frac{1}{P} \sum_{f=1}^F w_f, \quad (2.3.36)$$

тогда дальше будем работать с матрицей W^* :

$$W^* = W - T \underbrace{[1 \dots 1]}_P = MS \quad (2.3.37)$$

Поскольку матрица W^* представима в виде произведения матрицы M размера $2F \times 3$ и матрицы S размера $3 \times P$, её ранг не может быть больше 3. В настоящей работе выполнение факторизации ранга 3 матрицы W , производилось посредством сингулярного разложения матриц (SVD):

$$W^* = U \Sigma V^T = (U \sqrt{\Sigma})(\sqrt{\Sigma} V^T) = \bar{M} \bar{S}, \quad (2.3.38)$$

здесь U, V - ортогональные матрицы, а $\Sigma = \text{diag}(\sigma_1, \dots, \sigma_n)$, $n = \min(2F, P)$, $\sigma_1, \dots, \sigma_n$ - сингулярные числа матрицы W . Упорядочим все сингулярные значения по убыванию, и выберем из матрицы Σ верхнюю подматрицу размера 3×3 , значениями которой будут 3 максимальных сингулярных числа, соответствующих основным компонентам матрицы W . Тогда, выбирая из матриц U и V подматрицы из строк, соответствующих трем наибольшим сингулярным числам, получаем аппроксимацию к матрице W . Понятно, что такое разложение не единственно, поскольку между матрицами \bar{M} и \bar{S} всегда можно вставить произведение прямой и обратной матриц Q ранга 3, ничего при этом не изменив, тогда можно записать:

$$W^* = \bar{M} \bar{S} = \bar{M} (Q Q^{-1}) \bar{S} = (\bar{M} Q) (Q^{-1} \bar{S}) = MS, \quad (2.3.39)$$

Здесь M и S подразумевают истинные матрицы движения и формы.

Так, как $MM^T = M Q Q^T M^T$, то, в силу ортонормированности базисных векторов, задающих КСК:

$$\begin{cases} |\vec{i}_f| = 1, \\ |\vec{j}_f| = 1, \\ (\vec{i}_f, \vec{j}_f) = 0. \end{cases} \Rightarrow \begin{cases} |\vec{m}_f| = 1, \\ |\vec{n}_f| = 1, \\ (\vec{m}_f, \vec{n}_f) = 0. \end{cases} \quad (2.3.40)$$

получим следующие системы уравнений для приближений для нахождения матрицы Q :

$$\begin{array}{ll} \text{ОП:} & \text{МОП, ППП:} \\ \begin{cases} \vec{m}_f Q Q^T \vec{m}_f^T = 1, \\ \vec{n}_f Q Q^T \vec{n}_f^T = 1, \\ \vec{m}_f Q Q^T \vec{n}_f^T = 0. \end{cases} & \begin{cases} \vec{m}_f Q Q^T \vec{m}_f^T - \vec{n}_f Q Q^T \vec{n}_f^T = 0, \\ \vec{m}_f Q Q^T \vec{n}_f^T = 0, \\ \vec{m}_f Q Q^T \vec{m}_1^T = 1. \end{cases} \end{array} \quad (2.3.41)$$

Где значения \vec{m}_f и \vec{n}_f соответствуют таблице 2.3.2. (см. также табл. 2.3.1)

Таблица 2.3.2.

ОП	МОП	ППП
$\vec{m}_f = \vec{i}_f$	$\vec{m}_f = \frac{\vec{i}_f}{z_f}$	$\vec{m}_f = \frac{1}{z_f} \left(\vec{i}_f - \frac{x_f}{g} \vec{k}_f \right)$
$\vec{n}_f = \vec{j}_f$	$\vec{n}_f = \frac{\vec{j}_f}{z_f}$	$\vec{n}_f = \frac{1}{z_f} \left(\vec{j}_f - \frac{y_f}{g} \vec{k}_f \right)$

В данной работе системы (2.3.41) решались методом наименьших квадратов (МНК) с применением SVD.

Неоднозначность восстановления формы сцены связана с тем, что в задачу легко можно внести знаковую неоднозначность, заменив матрицу Q на следующую матрицу:

$$Q \begin{pmatrix} \pm 1 & 0 & 0 \\ 0 & \pm 1 & 0 \\ 0 & 0 & \pm 1 \end{pmatrix} \quad (2.3.42)$$

Зафиксировав ориентацию осей системы координат, например, выбрав направление осей таким же, как у системы, связанной с одной из камер, неоднозначность в первых двух знаках можно устранить. При этом более целесообразно выбирать камеру, ориентация которой ближе всего к нормальной к обнимаемой поверхности. Учитывая выравнивание по первой камере, получим искомые матрицы движения и формы:

$$\begin{cases} \tilde{M} = MR_0 \\ \tilde{S} = R_0^T S \\ R_0 = [\vec{i}_f, \vec{j}_f, \vec{k}_f], f = 1..F \end{cases} \quad (2.3.43)$$

Неоднозначность в третьем знаке связана с тем, что во всех рассмотренных нами линейных приближениях, глубиной объекта по сравнению с расстоянием до него пренебрегают (происходит так называемое "уплощение" сцены). Как следствие, это приводит к неоднозначности восстановления формы сцены S и движения камер M .

2.3.7. Итерационный метод решения 3D-задачи

Основными недостатками рассмотренных выше линейных приближений, являются неоднозначность в определении знака глубины сцены и невозможность вычисления фокусных расстояний камер. Наличие на изображениях перспективных искажений является дополнительным источником информации, позволяющим устранить перечисленные недостатки.

Поскольку метод точный, он описывается уравнениями (2.3.39), в которых мы полагаем g постоянным для всех кадров.

$$\begin{cases} u_{fp} = g \frac{(\vec{i}_f, \vec{s}_p) + x_f}{(\vec{k}_f \vec{s}_p) + z_f}, \\ v_{fp} = g \frac{(\vec{j}_f, \vec{s}_p) + y_f}{(\vec{k}_f \vec{s}_p) + z_f}, \end{cases} \quad (2.3.44)$$

Перепишем (2.3.44) в виде, используя уже встречавшуюся замену z_f на z_f' :

$$\begin{cases} (1 + \frac{1}{g} \frac{(\vec{k}_f \vec{s}_p)}{z_f'}) u_{fp} = \frac{1}{z_f'} ((\vec{i}_f, \vec{s}_p) + x_f), \\ (1 + \frac{1}{g} \frac{(\vec{k}_f \vec{s}_p)}{z_f'}) v_{fp} = \frac{1}{z_f'} ((\vec{j}_f, \vec{s}_p) + y_f), \end{cases} \quad (2.3.45)$$

Таким образом, правая часть полученных уравнений соответствует правой части уравнений для МОП (2.3.17), а левая представляет собой сумму левой части линейных приближений и добавки. В матричной форме получим теперь:

$$W = W_1 + \alpha W_2, \text{ где } \alpha = \frac{1}{g},$$

$$W_1 = \begin{pmatrix} u_{11} & u_{12} & \dots & u_{1P} \\ v_{11} & v_{12} & \dots & v_{1P} \\ \vdots & \vdots & \dots & \vdots \\ u_{F1} & u_{F2} & \dots & u_{FP} \\ v_{F1} & v_{F2} & \dots & v_{FP} \end{pmatrix}, \quad W_2 = \begin{pmatrix} \frac{\vec{k}_1 \vec{s}_1}{z_1'} u_{11} & \frac{\vec{k}_1 \vec{s}_2}{z_1'} u_{12} & \dots & \frac{\vec{k}_1 \vec{s}_P}{z_1'} u_{1P} \\ \frac{\vec{k}_1 \vec{s}_1}{z_1'} v_{11} & \frac{\vec{k}_1 \vec{s}_2}{z_1'} v_{12} & \dots & \frac{\vec{k}_1 \vec{s}_P}{z_1'} v_{1P} \\ \vdots & \vdots & \dots & \vdots \\ \frac{\vec{k}_F \vec{s}_1}{z_F'} u_{F1} & \frac{\vec{k}_F \vec{s}_2}{z_F'} u_{F2} & \dots & \frac{\vec{k}_F \vec{s}_P}{z_F'} u_{FP} \\ \frac{\vec{k}_F \vec{s}_1}{z_F'} v_{F1} & \frac{\vec{k}_F \vec{s}_2}{z_F'} v_{F2} & \dots & \frac{\vec{k}_F \vec{s}_P}{z_F'} v_{FP} \end{pmatrix}. \quad (2.3.46)$$

Т.е. матрица W представлена в виде двух слагаемых, первое из которых, W_1 , соответствует введённой ранее матрице измерений, а второе, W_2 , зависит от W_1 , s_p , k_f , и имеет смысл поправки на перспективные искажения:

Как было показано ранее, ранг правой части (2.3.45) всегда равен 3, поэтому предлагается по итерациям подбирать параметр α , используя известные с предыдущего шага значения s_p , k_f , таким образом, чтобы сохранить равенство рангов обеих частей (2.3.45). Итак, система уравнений (2.3.45) может быть решена итерациями, путём уточнения поправочной матрицы.

Предложенный итерационный алгоритм хорошо работает в случае центрального наведения на объект, вследствие того, что в качестве начального приближения выбрана МОП - проекция. Если есть существенная необходимость работать строго в ситуации нецентрального наведения, целесообразнее использовать в качестве

начального приближения парapersпективную проекцию. Сформулируем предлагаемый алгоритм решения нелинейной задачи восстановления.

Алгоритм.

1. Полагаем в (2.3.46) Номер итерации $q=0$, параметр $\alpha=0$, $W^{(0)}:=W_1$, $W_2^{(0)}:=0$.
2. Решаем систему уравнений, описывающую МОП-приближение: $W^{(q)} = M^{(q)}S^{(q)} + T^{(q)}$.
3. Полагаем теперь: $q:=q+1$.
4. Вычисляем поправочную матрицу: $W_2^{(q)}=W_2^{(q)}(M^{(q-1)}, S^{(q-1)}, T^{(q-1)})$.
5. Находим коэффициент $\alpha^{(q)}$, подбирая его таким, чтобы ранг матрицы $W^{(q)}$ оставался равным 3, то есть надо искать:

$$\min_{\alpha^{(q)}} \frac{\sigma_4}{\sigma_1} (W_1 + \alpha^{(q)}W_2^{(q)}) \quad (2.3.47)$$

где σ_1, σ_4 - это первое и четвёртое сингулярные значения матрицы $W^{(q)}$. Диапазон изменения параметра $\alpha^{(q)}$ определяется из условия: $1 + \alpha^{(q)} \frac{\bar{k}_f \bar{S}_f}{z_f'} > 0$.

6. Обновляем матрицу $W^{(q)}=W_1+\alpha^{(q)}W_2^{(q)}$.
7. Переходим к п.2, если не выполняется условие выхода: $|\alpha^{(q-1)} - \alpha^{(q)}| < \epsilon$, где значение ϵ выбирается достаточно малым.
8. Устраняем неоднозначность знака глубины сцены:

if $\alpha^{(q)} < 0$, then

$$\alpha^{(q)} = -\alpha^{(q)}$$

$$S^{(q)} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{pmatrix} S^{(q)}; \quad M^{(q)} = M^{(q)} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{pmatrix};$$

endif

9. Выравниваем МСК по СК, связанной с одной из камер.

2.4. Натягивание поверхности на восстановленные точки и построение трехмерной модели VRML

В результате работы описанных выше алгоритмов получаются координаты точек объекта, координаты положения камер и тройки векторов ориентаций камер. Одним из наиболее простых и наглядных способов просмотра результатов восстановления является импорт в VRML полученных данных с натягиванием поверхности и текстуры на точки объекта. Текстуры выбираются с кадра, который был получен с направления наиболее близкого к среднему значению нормали к восстановленному фрагменту сцены. Координаты и направления камер также записываются в модель VRML.

Задача натягивания поверхности на трехмерный объект может быть решена двумя способами:

- натягивание выпуклой оболочки на трехмерный объект
- двумерная триангуляция

Каждый из методов имеет свои достоинства и недостатки и могут использоваться совместно, дополняя друг друга.

Выпуклая оболочка может применяться, если требуется грубая оценка формы объекта для избежания столкновения, или если заведомо известно, что исследуемый объект имеет выпуклую форму (или если допустимо пренебречь невыпуклостью). Такой подход позволяет сократить количество необходимых характеристических точек, ограничиваясь лишь самыми «сильными», т.к. точки, попадающие внутрь выпуклой поверхности не используются. Для реализации натяжения выпуклой оболочки может использоваться алгоритм [47].

Триангуляция дает более точную и правильную картину формы объекта. Тем не менее адекватный вид поверхности можно получить только при достаточном количестве точек и при условии, что характеристические точки были найдены и на углах и ребрах объекта. Выбор двумерной триангуляции определяется тем, что этот подход позволяет избежать ряда ошибок, связанных с взаимным расположением точек в пространстве. К примеру представим объект наподобие перевернутой буквы «П» (см. рис 2.4.1), где высота «ножек» больше расстояния между ними. Очевидно, что при трехмерной триангуляции точки A и B будут соединены и впадина не будет обнаружена. Проекция же этих точек A', B', C' имеют адекватное расположение и наличие впадины будет зафиксировано. Разумеется, форма впадины будет не прямоугольная, т.к. четвертой точки не видно. В дальнейшем ее форма может быть уточнена в результате обьёмки объекта с других углов и наложением результатов.

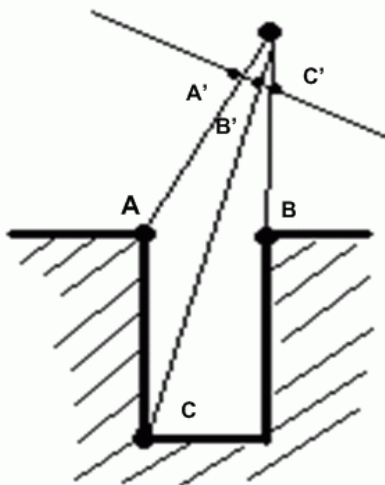


Рис. 2.4.1. Точки в пространстве и на плоскости.

Таким образом, в ситуации, когда нас интересует именно форма восстановленного объекта, целесообразно для натягивания поверхности использовать двумерную триангуляцию [48],[49].

Триангуляцией называется разбиение геометрической области, как правило массива точек и многоугольника, на простые геометрические фигуры, которые имеют общие ребра и не пересекаются. Таким

образом, в двумерном пространстве триангуляция состоит из треугольников, имеющих общие ребра и вершины. Оптимальной триангуляцией называется такая триангуляция, которая соответствует некоторым условиям, накладываемым на такие параметры как размер, форма и количество ячеек разбиения.

Постановка задачи триангуляции определяется тремя факторами:

- тип входных данных,
- тип триангуляции,
- критерии оптимизации.

Наиболее часто встречающиеся типы входных данных в двумерном пространстве – многоугольник, в том числе многоугольник с дырами, массив точек и простой планарный граф, который может использоваться как обобщенная форма вышеперечисленных форм данных.

Выбор типа триангуляции в первую очередь определяется следующей альтернативой. Триангуляция может быть построена только на входных точках или с использованием дополнительных, называемых точками Штейнера.

Критерии оптимизации, как правило, имеют своей целью повышение производительности и точности, что, как правило, приводит к противоположно направленным требованиям к сетке разбиения. В такой ситуации обычно выбирают два наиболее значимых критерия и работают с ними. Тем не менее, такой основополагающий метод, как триангуляция Делоне оптимизирует разбиение одновременно по нескольким критериям.

Задача, поставленная в данной работе в данной терминологии выглядит следующим образом: триангуляция без точек Штейнера массива входных данных с использованием минимизации наибольшего угла.

Рассмотрим принципы, на которых базируется триангуляция и, в частности, алгоритм триангуляции Делоне, используемый в настоящей работе.

Триангуляция без оптимизации базируется на утверждении, что каждый многоугольник с более чем тремя сторонами имеет диагональ. Диагональю здесь традиционно называется отрезок прямой, соединяющий две вершины, полностью лежащий внутри многоугольника и не имеющий с границей многоугольника других общих точек, кроме концов отрезка. Существование диагонали многоугольника определяет возможность разделить многоугольник на два с меньшим количеством вершин в каждом и далее рекурсивно строя диагонали для каждого получим разбиение на треугольники. Т.е. триангуляция всегда существует. Такой подход приводит к алгоритму со сложностью $O(n^2)$, но может быть приведен к сложности $O(n \log n)$, в частности для входных данных в виде массива точек.

Оптимизация. Т.к. триангуляция без точек Штейнера строит фиксированное количество треугольников, в основу критерия оптимизации ложится форма треугольника. Обычно критерий строится с использованием таких величин как углы, длина ребра, высоты и площадь треугольника.

Триангуляция Делоне одновременно оптимизирует несколько параметров, упомянутых выше: максимальный угол, радиус описанной и вписанной окружностей. триангуляция Делоне двумерного массива точек это по сути известная диаграмма Вороного. Диаграмма Вороного – это разделение плоскости на многоугольные ячейки по одной на каждую точку входных данных таким образом, что ячейка точки а представляет собой геометрическое место точек плоскости расстояние от которых до а меньше, чем расстояние до других точек входного массива. Если во входном массиве нет таких четырех точек, которые лежат на одной окружности, то из каждой вершины выходят по три ребра и триангуляция Делоне, которая по сути есть диаграмма Вороного для этих вершин, и будет искомым триангуляцией. (см. рис. 2.4.2). Если же существуют такие четыре точки, лежащие на одной окружности, то они становятся вершинами более крупной ячейки, которая затем может быть триангулирована для получения искомым триангуляции Делоне.

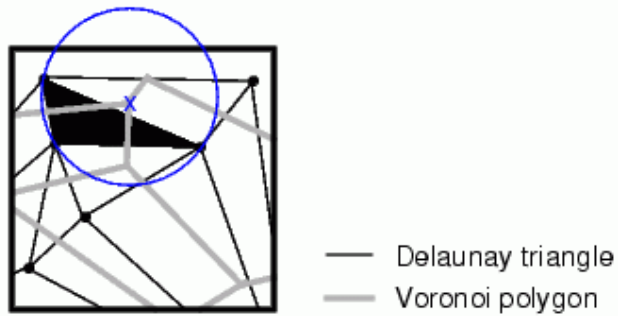


Рис. 2.4.2. Триангуляция Делоне и диаграмма Вороного.

Существует интересная взаимосвязь между триангуляцией Делоне и построением трехмерной выпуклой оболочки. Поместим каждую точку входного двумерного массива в трехмерное пространство, переводя точку (x,y) в точку (x,y,x^2+y^2) , таким образом эти точки окажутся на параболоиде. Выпуклая оболочка этих точек может быть разделена на две части – верхнюю и нижнюю. Ячейка относится к нижней части выпуклой оболочки, если она принадлежит плоскости, отделяющей массив точек от области $(0,0, -\infty)$. Можно показать, что триангуляция Делоне входного массива точек есть проекция нижней части выпуклой оболочки на плоскость xy , как изображено на рис. 2.4.3.

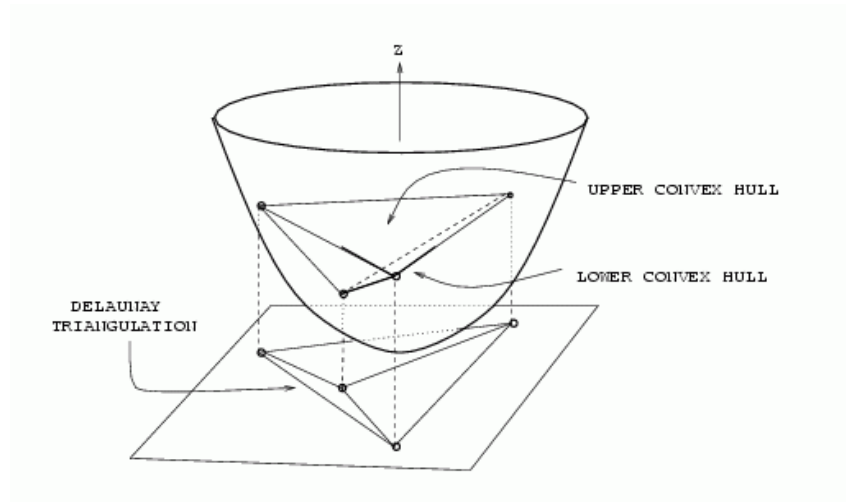


Рис. 2.4.3. Триангуляция Делоне и выпуклая оболочка.

Наконец, приведем четкий критерий: если a и b – входные точки, триангуляция Делоне содержит ребро $\{a,b\}$ тогда и только тогда, когда существует окружность, проходящая через a и b , которая не пересекает других точек входного массива и не содержит их внутри себя. Более того каждая описанная вокруг треугольника Делоне окружность не содержит внутри себя точек входного массива.

Существует много реализаций триангуляции Делоне. Сложность этих алгоритмов как правило имеет величину $O(n \log n)$. Тем не менее отдельные алгоритмы имеют и линейную сложность.

3. Результаты

3.1. Построение моделей для генерации синтетических данных

3.1.1. Модель «Куб»

В среде Matlab была разработана модель, представляющая собой три грани прямоугольного параллелепипеда (см. рис. 3.1.1), на каждой из которых характеристические точки образуют равномерную прямоугольную сетку. С помощью построенной модели, был организован автоматический генератор характеристических точек, который необходим в процессе реконструкции трёхмерных сцен, поскольку большинство используемых алгоритмов, и особенно их сходимость, как правило, зависят от числа характеристических точек. Помимо этого, появилась возможность сравнения полученных результатов восстановления с эталоном.

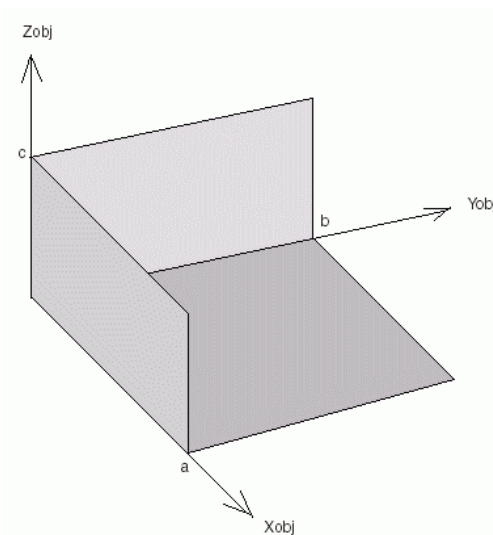


Рис. 3.1.1. Модель «Куб».

Рассмотрим детали построения данной модели. Поверхность трёхгранного угла была представлена в виде равномерной сетки прямоугольников, количество отрезков на каждом ребре угла равно N . В качестве характеристических точек брались координаты вершин прямоугольников, принадлежащих поверхности трёхгранного угла. Характеристические точки хранятся в матрице X , в процессе формирования которой было учтено, что на пересечении рёбер трёхгранного угла лежат одни и те же точки:

$$X = \begin{pmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_p \\ \dots & \dots & \dots \\ x_p & y_p & z_p \end{pmatrix} \quad (3.1)$$

где $P = 3*N*(N+1)+1$.

В модели введены 3 системы координат: объектная система координат (ОСК), с центром в вершине трёхгранного угла, МСК с началом в ЦМ точек объекта, и КСК, связанная с f -й камерой. Все камеры позиционированы таким образом, чтобы объект всегда полностью находился в поле зрения каждой из них. Для этого необходимо сориентировать камеры на центр объекта, и расположить их в соответствующем секторе пространства с заданным углом $\xi \in (0,90)$ (в градусах). При этом расположение сектора таково, что осью его симметрии является прямая, задаваемая уравнением $x = y = z$. Для получения ограничений, определяющих этот сектор, поместим наш объект в положительном секторе ОСК. В таком случае камеры должны задаваться в

отрицательном секторе ОСК, тогда все грани кубика будут находиться в поле зрения камер. В модели введены ограничения на внутренние параметры камеры (в данном случае, на фокус), а также, на минимальное и максимальное расстояния до объекта. Поэтому вводим r_{\min} и r_{\max} такие что:

$$\begin{cases} r_{\min} = \frac{\sqrt{a^2 + b^2 + c^2}}{g} \\ r_{\max} = k * r_{\min}, k \in [1,5] \end{cases} \quad (3.2)$$

Позиции камер $CamPos_f$, $f = 1..F$, где F – количество камер, в МСК задаются в полученном секторе, с помощью случайного распределения. Формируется матрица:

$$CamPos = \begin{pmatrix} t_{x1} & t_{x2} & \dots & t_{xF} \\ t_{y1} & t_{y2} & \dots & t_{yF} \\ t_{z1} & t_{z2} & \dots & t_{zF} \end{pmatrix} \quad (3.3)$$

Все камеры ориентированы в модели на ЦМ точек объекта:

$$CubeCenter = \frac{1}{P} \sum_{p=1}^P X_p \quad (3.4)$$

Фиксируем направление орта \mathbf{k}_{cam}^f . Остальные орты правой тройки, \mathbf{i}_{cam}^f , \mathbf{j}_{cam}^f , можно задавать произвольно. Например, так:

$$\begin{cases} \vec{i}_{cam} = \frac{\vec{j}_{obj} \times \vec{k}_{cam}}{|\vec{j}_{obj} \times \vec{k}_{cam}|}, \\ \vec{j}_{cam} = -\vec{i}_{cam} \times \vec{k}_{cam}, \\ \vec{k}_{cam} = \frac{CubeCenter - CamPos}{|CubeCenter - CamPos|}. \end{cases}, \quad \text{где} \quad \begin{cases} \vec{i}_{obj} = [1 \ 0 \ 0]^T, \\ \vec{j}_{obj} = [0 \ 1 \ 0]^T, \\ \vec{k}_{obj} = [0 \ 0 \ 1]^T. \end{cases} \quad (3.5)$$

Здесь подразумевается, что для любого вектора \mathbf{a} $|\bar{a}| = \sqrt{(\bar{a}^T \bar{a})}$. Если окажется, что $\mathbf{j}_{obj} \parallel \mathbf{k}_{cam}$, то есть $[\mathbf{j}_{obj}, \mathbf{k}_{cam}] = 0$, то орты $\mathbf{i}_{obj}^f, \mathbf{j}_{cam}^f$ считаем следующим образом:

$$\begin{cases} \vec{j}_{cam} = \frac{\vec{i}_{obj} \times \vec{k}_{cam}}{|\vec{i}_{obj} \times \vec{k}_{cam}|}, \\ \vec{i}_{cam} = \vec{j}_{cam} \times \vec{k}_{cam}, \end{cases} \quad (3.6)$$

В модели также предусмотрено, что камеры могут наводиться не только на ЦМ точек объекта, но и на любую другую точку, а также каждая камера в отдельности может наводиться на произвольную точку массива X . Помимо этого, в модели существует возможность зашумления входных данных, основанные на различных явлениях:

- внесение погрешности, связанной с конечным количеством пикселей в строке изображения
- внесение погрешности плавающей характеристической точки для имитации ошибок трекера
- зашумление и сферическое искажение, портящее непосредственно изображение камеры.

Таким образом модель определяется таким набором изменяемых параметров: Минимально допустимое расстояние до камеры, соотношение между минимальным и максимальным расстоянием, углы съемки, количество камер, разрешение в пикселах.

3.1.2. Модель «Спутник над плоскостью с прямоугольным выступом»

По аналогии с моделью «Куб» была разработана модель спутника, летящего по окружности радиуса $R+H$, где R – радиус Земли, H – высота орбиты над поверхностью, $H \sim 1000$ км, над участком плоскости, имеющим небольшую приподнятую часть малой, но разрешимой высоты, см рис ...

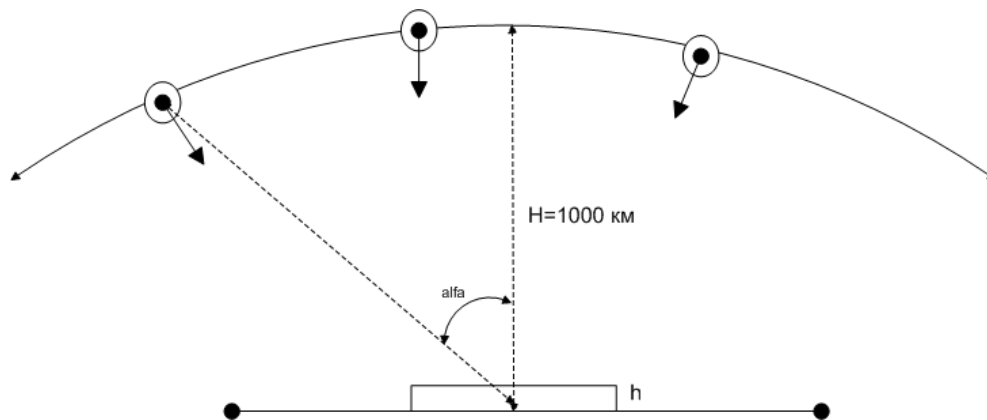


Рис. 3.1.2. Модель «Спутник над плоскостью с прямоугольным выступом».

Обснимаемая плоскость со ступенькой помещается в плоскость осей xu и ее центр совпадает с началом координат. Плоскость разбивается на N отрезков, что соответствует $(N+1)^2$ точек. Спутник движется по окружности, центр которой находится в точке $(0,0,-R)$. Окружность лежит в плоскости yz . Съемка производится, когда угол между нормалью к плоскости-объекту и отрезком, соединяющим центр плоскости-объекта и спутник, меньше или равен заданного угла $\alpha \in (0, 90)$ (в градусах). Производится n кадров через равные углы.

Параметрами, определяющими модель являются высота ступеньки, площадь, занимаемая ею на снимаемой поверхности, разрешение в метрах, определяющее видимый размер объекта, допустимые углы съемки и количество кадров.

Основной проблемой этой модели оказался тот факт, что спутник производит съемку вдоль плоской траектории и по одной из координат точки на изображении не двигаются. Если при этом камера всегда ориентирована на центр масс объекта, который в силу малой выпуклости объекта находится практически в плоскости объекта, то при заглублении, связанном с конечным количеством пикселей в строке изображения, появляется «мертвая» точка, которая, находясь по центру кадра, имеет координаты на изображении $(0,0)$ на всех кадрах серии и приводит к сбою в работе программы.

3.2. Структура программы восстановления трехмерных сцен

Программа восстановления трехмерных сцен была реализована на языке C++ в независимом от операционной системы виде. Основные блоки программы соответствуют схеме на рис. 2.1.2. В нее были включены восстановление трехмерной сцены методом факторизации матриц с помощью линейных алгоритмов, основанных на описанных в предыдущей главе приближениях, и точного итерационного алгоритма. В режиме поиска критерия перехода от одного приближения к другому все алгоритмы работали параллельно. Далее программа получает триангуляцию, используя модуль Triangle и строит VRML файл, используя восстановленные данные и, если это задано, натягивая текстуру с одного из исходных изображений, снятого с направления наиболее близкого к среднему значению нормали к восстановленному фрагменту сцены.

Входными параметрами программы являются имена файлов в формате *.mtr:

1. Файл с матрицей W , размера $[2F, P]$, нормализованной.
2. Файл с вектором g , размера $[F, 1]$, т.е. с фокусом. В простейшем случае – вектор из единиц. Подразумевает возможность произведения съемки с различными фокусами, однако пока итерационный алгоритм не был обобщен на многомерный случай, это может быть применимо лишь для линейных алгоритмов.
3. Файл с именами сохраняемых файлов, задаваемых в следующей последовательности:
 - I. Файлы ортографии:
 - i. форма
 - ii. положение камер
 - iii. ориентация камер
 - iv. файл VRML
 - II. Файлы масштабируемой ортографии:
 - i. форма
 - ii. положение камер
 - iii. ориентация камер
 - iv. файл VRML
 - III. Файлы параперспективы:
 - i. форма
 - ii. положение камер
 - iii. ориентация камер
 - iv. файл VRML
 - IV. Файлы перспективы:
 - i. форма
 - ii. положение камер
 - iii. ориентация камер
 - iv. файл VRML
 - V. Файл текстуры с путем относительно директории где лежат VRML файлы. За неимением - -1
 - VI. Номер кадра которому эта текстура соответствует. За неимением - 0.

Последние два пункта включены временно и не вписываются по смыслу в данный файл, в дальнейшем это планируется изменить. Отдельным входным параметром будет файл со списком имен имеющихся кадров в последовательности, соответствующей нумерации точек.

Программа сама должна выбрать нужный, основываясь на информации об углах ориентации камер, полученной при восстановлении

4. Необязательный параметр. В программе предполагается реализовать возможность восстановления направления выпуклости для линейных алгоритмов. Для этого предполагается использовать дополнительную информацию. К примеру, в достаточно распространенном случае съемке статической сцены, когда движется только камера, можно получить вектор направления относительного смещения по выбранной оси (вправо или влево поехала камера) в виде набора 1,-1 и 0, если камера по оси не двигалась. Этот параметр должен быть именем файла, содержащего дополнительную информацию подобного рода.

На выходе программа сохраняет файлы, указанные в п.3.

3.3. Совмещение результатов восстановления с моделью

Будем обозначать координаты точек модели в КСК вектором \mathbf{x}_p , а координаты восстановленных точек в восстановленной КСК, - вектором \mathbf{s}_p , и смещение камеры - вектором \mathbf{t} .

Переход от одних координат к другим, осуществляется следующим преобразованием:

$$\mathbf{x}_p' = \alpha R \mathbf{s}_p + \mathbf{t} \quad (3.7)$$

где α - масштабирующий коэффициент, R - матрица поворота. Для того, чтобы совместить результаты восстановления с моделью, необходимо минимизировать следующий функционал по параметрам α , R , \mathbf{t} :

$$\Phi = \sum_{p=1}^P (x_p - x_p')^2 = \sum_{p=1}^P (x_p - \alpha R \bar{s}_p - \bar{t})^2 \quad (3.8)$$

Условия минимума:

$$\frac{\partial \Phi}{\partial \alpha} = 0, \quad \frac{\partial \Phi}{\partial \bar{t}} = 0, \quad \frac{\partial \Phi}{\partial R} = 0, \quad (3.9)$$

в последнем равенстве подразумевается производная по параметрам матрицы R . Распишем подробнее (3.8):

$$\Phi = \sum_{p=1}^P (\bar{x}_p \bar{x}_p' + \alpha^2 \bar{x}_p^T R^T R \bar{s}_p + \bar{t}^T \bar{t} - 2\alpha \bar{x}_p^T R \bar{s}_p - 2\bar{x}_p^T \bar{t} + 2\alpha \bar{t}^T R \bar{s}_p) \quad (3.10)$$

Отбрасывая в (3.10) все постоянные члены, и учитывая, что R - матрица поворота, то есть $R^{-1} = R^T$, получаем:

$$\begin{aligned} \Phi &= Const + \sum_{p=1}^P (\alpha^2 \bar{s}_p^T \bar{s}_p + \bar{t}^T \bar{t} - 2\bar{x}_p \bar{t} - 2\alpha (\bar{x}_p^T - \bar{t}^T) R \bar{s}_p), \\ \frac{\partial \Phi}{\partial \alpha} &= \sum_{p=1}^P (2\alpha \bar{s}_p^T \bar{s}_p - 2\alpha (\bar{x}_p^T - \bar{t}^T) R \bar{s}_p) = 0, \\ \frac{\partial \Phi}{\partial \bar{t}} &= \sum_{p=1}^P (2\bar{t} - 2\bar{x}_p + 2\alpha R \bar{s}_p) \end{aligned} \quad (3.11)$$

Отсюда получаем явные выражения для α, \mathbf{t} :

$$\begin{aligned} \alpha &= \frac{\frac{1}{P} \sum_{p=1}^P \bar{x}_p^T R \bar{s}_p - \left(\frac{1}{P} \sum_{p=1}^P \bar{x}_p^T \right) R \left(\frac{1}{P} \sum_{p=1}^P \bar{s}_p \right)}{\frac{1}{P} \sum_{p=1}^P \bar{s}_p^T \bar{s}_p - \left(\frac{1}{P} \sum_{p=1}^P \bar{s}_p^T \right) \left(\frac{1}{P} \sum_{p=1}^P \bar{s}_p \right)}, \\ \bar{t} &= \frac{1}{P} \sum_{p=1}^P \bar{x}_p - \alpha R \left(\frac{1}{P} \sum_{p=1}^P \bar{s}_p \right) \end{aligned} \quad (3.12)$$

Поскольку мы ищем решение задачи в МСК с началом в ЦМ точек объекта:

$$\frac{1}{P} \sum_{p=1}^P \bar{s}_p = 0$$

то формулы (3.12) упрощаются.

$$\alpha = \frac{\frac{1}{P} \sum_{p=1}^P \bar{x}_p^T R \bar{s}_p}{\frac{1}{P} \sum_{p=1}^P \bar{s}_p^T \bar{s}_p}, \quad (3.13)$$

$$\bar{t} = \frac{1}{P} \sum_{p=1}^P \bar{x}_p$$

Выпишем слагаемые в Φ , зависящие от R :

$$\begin{aligned} \Phi &= -\sum_{p=1}^P (\bar{x}_p^T - \bar{t}^T) R \bar{s}_p = -\sum_{p=1}^P \bar{x}_p^T R \bar{s}_p - \bar{t}^T R \bar{s}_p = \\ &= -\sum_{p=1}^P \bar{x}_p^T R \bar{s}_p + \left(\frac{1}{P} \sum_{p=1}^P \bar{x}_p^T \right) R \left(\frac{1}{P} \sum_{p=1}^P \bar{s}_p \right) - \frac{\alpha}{P} \left(\frac{1}{P} \sum_{p=1}^P \bar{s}_p \right) \underbrace{R^T R}_{=I} \left(\frac{1}{P} \sum_{p=1}^P \bar{s}_p \right) \cong \\ &\cong -\frac{1}{P} \sum_{p=1}^P (\bar{x}_p^T - \frac{1}{P} \sum_{p=1}^P \bar{x}_p^T) R \bar{s}_p. \end{aligned} \quad (3.14)$$

Далее действуем следующим образом: из условия $\frac{\partial \Phi}{\partial R} = 0$, находим матрицу R :

$$\begin{aligned} R &= \begin{pmatrix} \bar{t}_f & \bar{j}_f & \bar{k}_f \end{pmatrix} = \\ &= \begin{pmatrix} \cos \alpha_f \cos \beta_f & (\cos \alpha_f \sin \beta_f \sin \gamma_f - \sin \alpha_f \cos \gamma_f) & (\cos \alpha_f \sin \beta_f \cos \gamma_f - \sin \alpha_f \sin \gamma_f) \\ \sin \alpha_f \cos \beta_f & (\sin \alpha_f \sin \beta_f \sin \gamma_f - \cos \alpha_f \cos \gamma_f) & (\sin \alpha_f \sin \beta_f \cos \gamma_f - \cos \alpha_f \sin \gamma_f) \\ \sin \beta_f & \cos \beta_f \sin \gamma_f & (\cos \beta_f \cos \gamma_f \end{pmatrix} \end{aligned} \quad (3.15)$$

затем, зная R , находим масштабный множитель α и вектор трансляций камер \mathbf{t} по формулам (3.13).

3.4. Оценка точности восстановления

Предложенный алгоритм тестировался на синтетических данных. В настоящей работе приводится сопоставление результатов восстановления в приближении масштабируемой ортографической проекции и предлагаемым методом. Результаты восстановления сопоставляются также с исходной моделью. Так как МОП не восстанавливает фокусные расстояния камер, то, при сравнении этого приближения с моделью, использовались истинные значения фокусных расстояний, полученные из модели.

Вычисление погрешностей производился следующим образом. Сначала производилось совмещение моделей, как описано выше и проверялось направление выпуклости. Далее расчет производился по формулам:

$$ErrR = \frac{1}{3F} \sqrt{\sum_{f=1}^{3F} (r_f - r_f^0)^2}; \quad R = [r_1^T \quad r_2^T \quad \dots \quad r_{3F}^T];$$

$$R^0 = [r_1^{0T} \quad r_2^{0T} \quad \dots \quad r_{3F}^{0T}].$$

$$ErrShape = \frac{1}{P * a} \sqrt{\sum_{p=1}^P (x_p - x_p^0)^2}; \quad X = [x_1^T \quad x_2^T \quad \dots \quad x_P^T];$$

$$X^0 = [x_1^{0T} \quad x_2^{0T} \quad \dots \quad x_P^{0T}].$$

Здесь обозначения с нулем обозначают эталонные данные, а без нуля – восстановленные. a – характерный размер объекта, в случае с кубом – 1. Более сложно вычисление погрешности координат камер, разделенное условно на погрешность по XY и Z обусловлено тем, что смещение камер при удалении от объекта в основном происходит вдоль некоторой оси, эта ось и полагается за Z. X и Y выбираются произвольно в плоскости, перпендикулярной к этой оси. Ось находим как средний вектор смещения восстановленных координат относительно истинных:

$$\bar{Z} = \frac{1}{F} \sum_{f=1}^F (t_f - t_f^0); \quad CamPos = [t_1^T \quad t_2^T \quad \dots \quad t_F^T];$$

$$CamPos^0 = [t_1^{0T} \quad t_2^{0T} \quad \dots \quad t_F^{0T}].$$

Теперь проекции смещений восстановленных камер относительно истинных на эту ось и дадут нам искомую погрешность, которую мы берем в отношении к минимальному расстоянию до камеры.

$$ErrCamPosZ = \frac{1}{F * r_{\min}} \sqrt{\sum_{f=1}^F \left(t_f * \frac{(t_f, \bar{Z})}{|(t_f, \bar{Z})|} \right)^2}$$

$$ErrCamPosXY = \frac{1}{F * r_{\min}} \sqrt{\sum_{f=1}^F (t_f - \bar{Z} - t_f^0)^2}$$

Относительная погрешность в восстановлении фокуса может быть вычислена следующим очевидным способом:

$$Errg = \frac{1}{F * g} \sqrt{\sum_{f=1}^F (g_f - g_f^0)^2}$$

3.4.1. Модель «Куб»

В результате операций, описанных выше были построены следующие графики зависимостей, см рис. ... Эти данные могут использоваться для построения границ областей применимости алгоритмов. Данные снимались с одним и тем же случайным распределением для камер. Рассматривалась задача с бесконечным (ограниченным лишь машинной точностью) количеством пикселей в строке и с загрубленным до resolution=1024 пикселей в строке. При этом при внесении загрубления очевидно появление дисперсии в данных, и графики перестают быть гладкими. Величина дисперсии в зоне низких погрешностей для данного алгоритма порядка $2 \cdot 10^{-3} \sim 1/(\text{resolution}/2)$, но при выходе в зону больших погрешностей становится неуправляемой. Особенно ярко это выражено для перспективы на больших расстояниях, где погрешность в восстановлении координат камер случайным образом может выдавать огромные скачки, чего не происходит при тех же данных, но бесконечном разрешении.

В целом перспектива при расстояниях более 10^2 размеров объекта становится неустойчивой и перестает восстанавливать направление выпуклости объекта. Тем не менее восстановление точек объекта и ориентаций камер происходит с той же точностью, что и у линейных методов, чего не скажешь о погрешности в восстановлении координат камер, которая резко возрастает как по Z, так и по XY, равно как и точность в восстановлении фокуса.

При построении критерия следует учитывать влияние сразу нескольких факторов: расстояний до объекта, соотношение между минимальным и максимальным расстояниями, угол сектора, в котором располагаются камеры, количество камер и вносимые шумы. Все эти параметры влияют на положение точки слияния (как в графиках формы и ориентаций) или пересечения (как в графиках положения камер) графиков погрешностей. Внесение огрубления и шумов будут размывать положение этой точки и потребуются вычисление дисперсии. Тем не менее, при построении графиков для моделей с различными распределениями камер и их усреднении было отмечено, что основным параметром является отношение расстояния до объекта L к характерному размеру объекта a, а другие коэффициенты не вносят каких-либо координатных изменений в форму графиков и в положение точки их пересечения/слияния. Следует лишь учитывать допустимые диапазоны этих величин, наиболее важной из которых является угол сектора, в котором находятся камеры. Допустимый диапазон этой величины [20,70] градусов.

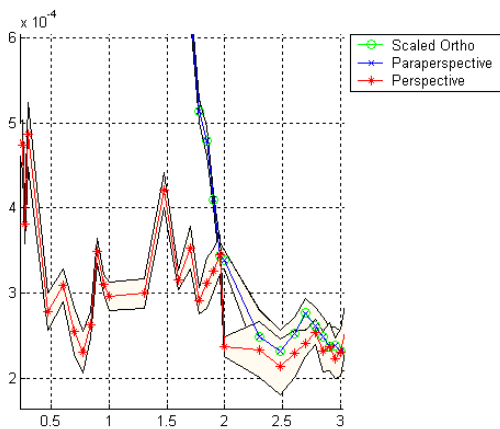


Рис. 3.4.1. Зависимость погрешности в восстановлении формы от $\lg(L/a)$

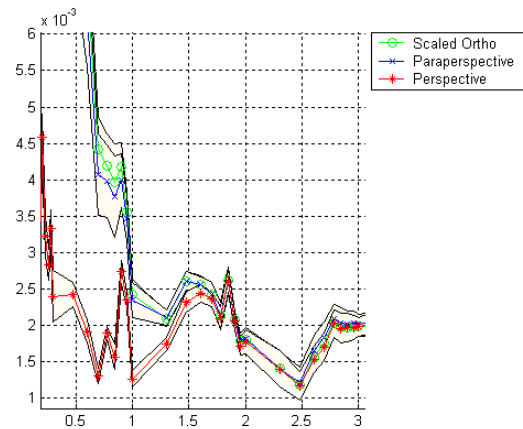


Рис. 3.4.2. Зависимость погрешности в восстановлении ориентаций камер от $\lg(L/a)$

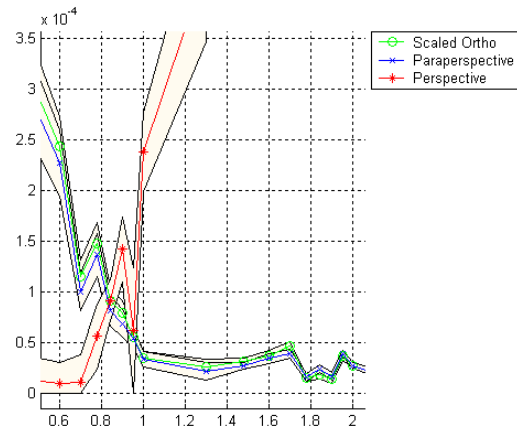


Рис. 3.4.2. Зависимость погрешности в восстановлении позиций камер по XY от $\lg(L/a)$

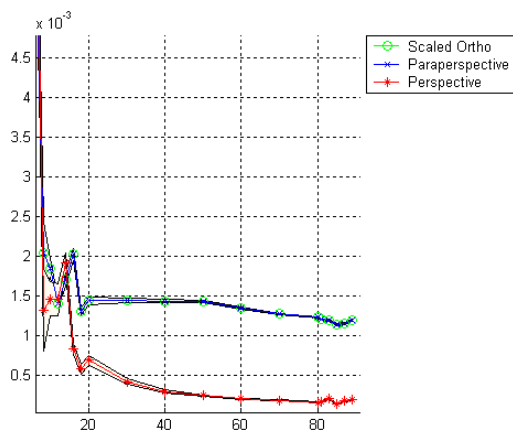


Рис. 3.4.3. Зависимость погрешности в восстановлении формы от угла сектора

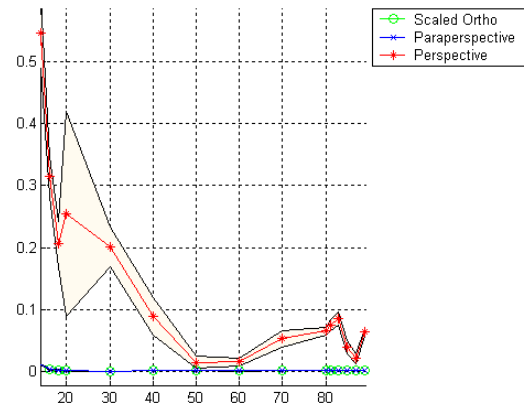


Рис. 3.4.4. Зависимость погрешности в восстановлении позиций камер по Z от угла сектора

Замечание. Зависимости погрешностей восстановления камер по XY и Z носят одинаковый характер, но по величине восстановление по Z менее точно на 2 порядка.

При построении графиков зависимости от одного параметра все остальные ставились в соответствие некоторым нейтральным «средним» значениям из диапазонов: $L/a = 20$; $ksi = 45$; $kmm = 1$; $nCam = 40$; $dpix = 0$.

Из приведенных графиков видно, что для различных восстанавливаемых величин получается различная величина порога, на котором происходит переход. Для более точного восстановления формы объекта и ориентаций камер перспективный метод следует использовать до $L/a = 10^2$, но для точного восстановления позиций камер следует переключаться на линейные методы уже при $L/a = 10$.

Совпадение графиков масштабируемой ортографии и параперспективы обусловлено тем, что камеры направлены на центр масс. Параперспектива дает выигрыш при произвольном направлении камеры, когда точность масштабируемой ортографии ухудшается..

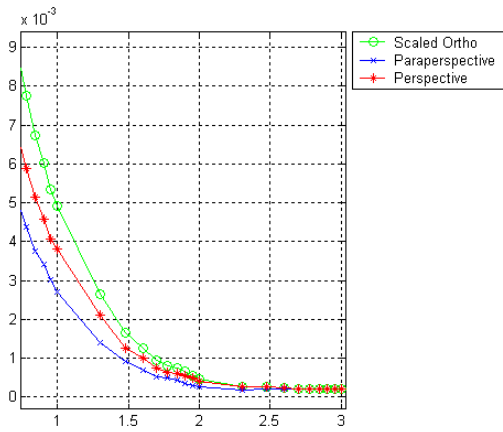


Рис. 3.4.5. Зависимость погрешности в восстановлении формы от $\lg(L/a)$

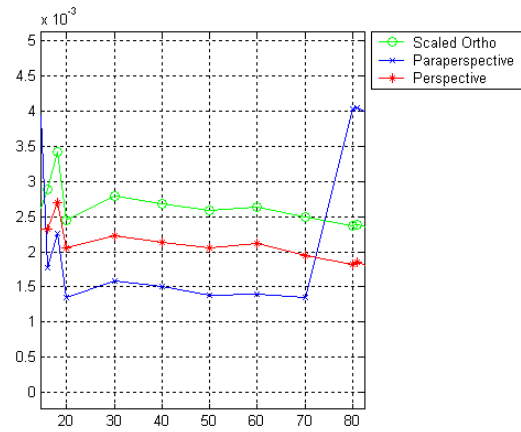


Рис. 3.4.6. Зависимость погрешности в восстановлении формы от угла сектора

3.4.2. Модель «Спутник над плоскостью с прямоугольным выступом»

По аналогии с тем, что было сказано выше для модели «Куб» для исследования погрешностей целесообразно исследовать гладкие зависимости, полученные при бесконечном разрешении, а затем учесть дисперсию, возникающую при внесении закругления.

При этом перспективный метод не использовался, т.к. расстояния заведомо больше тех, при которых перспективный метод устойчив. Задавались следующие параметры: угол съемки 45^0 , количество кадров 40, высота ступеньки на плоскости в два раза превышает разрешение изображения.

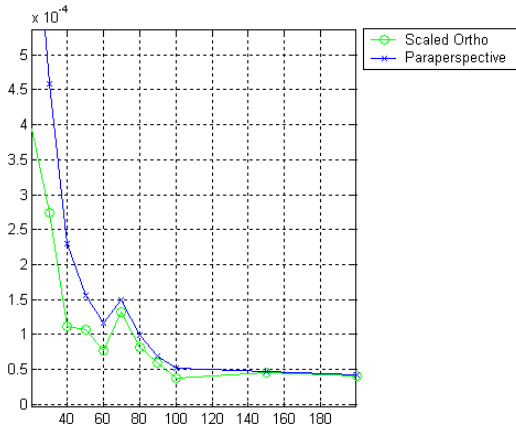


Рис. 3.4.7. Зависимость погрешности в восстановлении ориентаций камер от разрешения (в метрах). Размер объекта определяется как разрешение (в метрах) умноженное на количество пикселей в строке изображения.

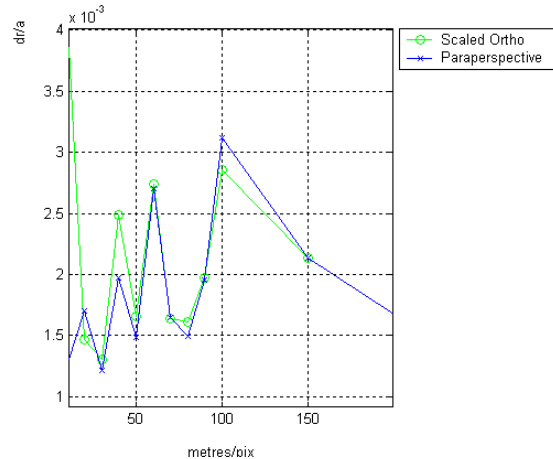


Рис. 3.4.6. Зависимость погрешности в восстановлении формы от разрешения (в метрах).

Линейные методы показали достаточную точность, но оказались чувствительны ко многим факторам, таким как точка, на которую ориентируется камера, угол к нормали, под которым располагается камера, по которой происходит выравнивание. Выравнивание по сильно отклоненной камере приводит к тому, что в результате совмещения эталона и результата они оказываются повернутыми друг относительно друга.

Таким образом представленные алгоритмы имеют перспективы применения при восстановлении рельефа по кадрам, снятым со спутника, но требуют особого внимания к параметрам и четкой проработки нюансов линейных алгоритмов.

Заключение

В настоящей работе дан обзор литературы по тематике восстановления трёхмерных сцен. Анализ литературных данных показал, что для эффективного восстановления трёхмерных сцен требуется интеграция различных методов. Сделан вывод о том, что в качестве основы такой интеграции, наиболее перспективными представляются методы, базирующиеся на факторизации матриц. Реализованы на языке Matlab 6.1 несколько известных из литературы алгоритмов, соответствующих линейным приближениям (ПОП, МОП и ППП).

Разработана базовая составляющая системы восстановления трехмерных сцен по последовательности цифровых изображений. Путем численного моделирования найден критерий адаптивного выбора модели (приближения) вычислений. Проведены оценки точности восстановления. Предложен подход к выделению плотной подматрицы измерений, основанный на поиске клики графа. Установлено, что наиболее слабым местом текущего состояния системы является требование малого изменения изображений между парами последовательных кадров, и планируемый блок грубого совмещения существенно необходим. Из-за отказа подсистемы восстановления фокусных расстояний в случае больших дальностей, для этого случая необходим альтернативный способ устранения неоднозначности формы [], основанный, например, на априорном грубом знании направления смещения камеры между некоторыми парами последовательных кадров.

Алгоритмы тестировались на синтетических данных. Для генерации последних, на языке Matlab была разработана физическая модель данных. И осуществлено исследование погрешностей восстановления в зависимости от ряда изменяемых параметров. Результаты восстановления визуализировались как в среде Matlab, так и посредством внешних программ, что обеспечивалось сохранением результатов в формате VRML.

Представляется целесообразным модифицировать алгоритмы с учётом движущихся внутри сцены объектов. Предполагается развитие алгоритмов в направлении рекурсивной обработки по мере поступления очередных кадров, совмещения результатов восстановления, полученных в результате съемки объекта с разных сторон (восстановление при обходе на 360°).

Также система требует дальнейшего исследования на предмет выявления более четкого критерия применимости описанных алгоритмов в целом и каждого из них в отдельности и дополнения альтернативными алгоритмами, упомянутыми во введении для расширения области применимости системы.

Литература

- [1] S. M. Smith and J. M. Brady. Susan: New approach to low-level image processing. *Journal of computer vision*, 23(1):45--78, May 1997.
- [2] S. M. Smith. and J. M. Brady. Real-time motion segmentation and shape tracking. In *IEEE Trans. On Pattern Analysis and Machine Intelligence*, volume~17, pages 814--820, 8 1995.
- [3] S. M. Smith and J. M. Brady. Susan - new approach to low-level image processing. Technical report TR95SMS1c, *Oxford Centre of Functional Magnetic Resonance Imaging of the Brian.Department of Clinical Neurology*, Oxford University, UK, 1995.
- [4] S. M. Smith. Integrated real-time motion segmentation and 3d interpretation. Technical Report TR96SMS1, *Oxford Centre of Functional Magnetic Resonance Imaging of the Brian.Department of Clinical Neurology*, Oxford University, UK, 1996. <http://www.fmrib.ox.ac.uk/~steve>.
- [5] A. Broggi M. Bertozzi and S. Castelluccio. A real-time oriented system for vehicle detection. *Journal of Systems Architecture*, 1996.
- [6] M. Watanabe S. K. Nayar and M. Noguchi. Real-time focus range sensor. Technical report, *Department of Computer Science, Columbia University*, New York, USA, June 1994. CUCS-028-94.
- [7] M. Watanabe S. K. Nayar and M. Noguchi. Real-time focus range sensor. *Proc. of Intl. Conf. on Computer Vision*, pages 995--1001, New York, USA, September 1995. ftp://ftp.cs.columbia.edu/pub/CAVE/papers/nayar/nayar\~nabe\~noguchi\~sensor_iccv\~95.ps.gz.
- [8] J. Krieger U. Schonfeldt and T. Behnke. Electrical subsystem of the high resolution stereo camera for mars express.
- [9] Nasa's earth observing system. the terra spacecraft. Technical report, *National Aeronautics and Space Administration. Goddard Space Flight Center*, 1999.
- [10] Nasa's earth observing system eos am-1. the first eos satellite. Technical report. , *National Aeronautics and Space Administration. Goddard Space Flight Center*, 1999.
- [11] B. K. P. Horn. Shape from Shading: A Method for Obtaining the Shape of a Smooth Opaque Object from One View. PhD thesis, *Massachusetts Institute of technology*, 1970.
- [12] K. Ikeuchi S. K. Nayar and T. Kanade. Surface reection: physical and geometrical perspectives. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 13, pages 611--634, 1991.
- [13] A. Pentland. Shape information from shading: a theory about human perception. In *In Proceedings of International Conference on Computer Vision*, pages 404--413, 1988.
- [14] Ruo Zhang, Ping-Sing Tsai, James Edwin Cryer and Mubarak Shah. Shape from shading: A survey. In *IEEE Transactions on PAMI*, 21(8):690--706, August 1999.
- [15] Ruo Zhang, Ping-Sing Tsai, James Edwin Cryer, and Mubarak Shah. Analysis of shape from shading techniques. In *IEEE CVPR - 94*, pages 377--384, Seattle, Washington, June 1994.
- [16] Ping-Sing Tsai and Mubarak Shah. Shape from shading using linear approximation. *Image and Vision Computing*, 12(8):487--498, October 1994.
- [17] Ping sing Tsai and Mubarak Shah. Shape from shading using linear approximation. In *IEEE IV CJ*, 1994.

- [18] Ping-Sing Tsai and Mubarak Shah. Shape from shading with variable albedo. *Optical Engineering*, pages 121--1220, April 1998.
- [19] C. H. Lee and A. Rosenfeld. Albedo estimation for scene segmentation. *Pattern Recognition Lett.* 1:155--160, 1983.
- [20] Д.В.Сивухин. Общий курс физики. Том 4. Оптика. - М.:Наука, ГРФМЛ, 1985.
- [21] Y. Xiong and S. Shafer. Depth from focusing and defocusing. <http://www.cs.cmu.edu/~yx/papers/IUW93.ps.Z>.
- [22] T. Choi M. Subbarao and A. Nikzad. Focusing techniques. Technical report, *Department of Electrical Engineering, State University of New York at Stony Brook*, September 1992.
- [23] Eric P. Krotkov. Focusing. *International Journal of Computer Vision*, (1):223--237, 1987.
- [24] Ф. П. Васильев. Численные методы решения экстремальных задач: Учебное пособие для вузов. - М.:Наука, ГРФМЛ, 1988.
- [25] A. Pentland. A new sense for depth of field. In *IEEE Transactions on PAMI*, volume 4, pages 523--531, 1987.
- [26] S. K. Nayar and M. Watanabe. Telecentric optics for constant-magnification imaging. Technical report, *Dept. of Computer Science, Columbia University, New York, USA*, September 1995. CUCS-026-95. ftp://ftp.cs.columbia.edu/pub/CAVE/papers/watanabe/nabe\~nayar\~tele_iuw\~95.ps.gz.
- [27] V. Roberto A. Fusiello and E. Trucco. Experiments with a new area-based stereo algorithm. In *International Conference on Image Analysis and Processing, Florence*, 1997.
- [28] S.T. Bernard. A stochastic approach to stereo vision. In *Proc. 5th National Conference AI.*, pages 676--680, Philadelphia, Pennsylvania, August 1986.
- [29] Andreas Koschan and Volker Rodehorst. Towards real-time stereo employing parallel algorithms for edge-based and dense stereo matching. In *Proc. of the IEEE Workshop on Computer Architectures for Machine Perception CAMP'95*, Como, Italy, September 1995.
- [30] Sebastien Roy and Ingemar J. Cox. A maximum-flow formulation of the n-camera stereo correspondence problem. In *IEEE. Proc. of Int. Conference on Computer Vision*.
- [31] J. E. Cryer Ping-Sing Tsai and Mubarak Shah. Integration of shape from shading and stereo. *Pattern Recognition.*, 28(7):1033--1043, 1995.
- [32] C.F. Hall and E.L. Hall. A nonlinear model for the spatial characteristics of the human visual system. *IEEE Trans. System Man Cybern.*, 6:161--170, 1977.
- [33] Takeo Kanade Mei Han. Scene reconstruction from multiple uncalibrated views. Technical report, *The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213*, January 2000. CMU-RI-TR-00-09.
- [34] Ali Azarbayejani Tony Jebara and Alex Pentland. 3d structure from 2d motion. Technical report, *MIT Media Laboratory*, May 1999. Perceptual Computing Tech. Rep.\#523.
- [35] G. S. Bestor. Recovering Feature and Observer Position by Projected Error Refinement. *PhD thesis, Computer Science Department, University of Wisconsin - Madison*, August 1998. Tech. Rep. 1381.
- [36] Chetverikov D. and Zsolt Szabo. A simple and efficient algorithm for detection of high curvature points in planar curves. In *Proc. 23rd Workshop of the Austrian Pattern Recognition Group*, pages 175--184, 1999.
- [37] Ramesh V., Bedekar A. S., Haralick R.M. and Zhang X. A bayesian corner Betector: Theory and performance evaluation. *Proceedings of 1994 ARPA Image Understanding Workshop*, pages 703--715, 1994.

- [38] Юрин Д. В. Непомнящий П. В., Хельвас А.В. Обнаружение угловых структур на контурных изображениях полученных сегментацией растра. Обработка информации и моделирование., 2002.
- [39] C. J. Poelman and T. Kanade. A paraperspective factorization method for shape and motion recovery. Technical report, *School of Computer Science, Carnegie Mellon University.*, Pittsburgh, PA 15213, December 1993. CMU-CS-93-219.
- [40] J. Costeria and T. Kanade. A multi-body factorization method for motion analysis. Technical report, *School of Computer Science, Carnegie Mellon University*, Pittsburgh, PA 15213, September 1994. CMU-CS-TR-94-220.
- [41] C. Tomasi and T. Kanade. Shape and motion from image streams: a factorization method (orthographic method). Technical report, *School of Computer Science, Carnegie Mellon University*, Pittsburgh, PA 15213, 1992.
- [42] T. Morita and T. Kanade. A sequantual factorization method for recovering shape and motion from image streams. *Proceedings of 1994 ARPA Image Understanding Workshop*, 2:1177--1188, November 1994.
- [43] Eric Krotkov Fabio Cozman. Depth from scattering. Technical report, *Robotics Institute, Carnegie Mellon University*, Pittsburgh,
http://www.ri.cmu.edu/pub/_files/pub2/cozman/_fabio/_1997/_1/cozman/_fabio/_1997/_1.pdf.
- [44] Bruce A. Maxwell. Segmentation and interpretation using multiple physical hypotheses of image formation. Technical report, *Carnegie Mellon University.*, July 1996.
- [45] Takeo Kanade Carlo Tomasi. Shape and motion from image streams: a factorization method - part 3. Detection and tracking of point features. Technical report, *School of Computer Science, Carnegie Mellon University*, Pittsburgh, PA 15213, April 1991. CMU-CS-91-132.
- [46] Н. В. Янова, Д.В. Юрин. Итеративный алгоритм восстановления трехмерных сцен, движения и фокусного расстояния камеры в перспективной проекции, основанный на факторизации матриц. – В сб. *Труды конференции. 12-я Международная конференция по Компьютерной графике и Машинному зрению ГрафиКон'2002* – стр. 123-129. Нижний Новгород, 2002.
- [47] Barber C.B., Dobkin D.P., Huhdanpaa H.T., "The Quickhull Algorithm for Convex Hulls," *ACM Transactions on Mathematical Software*, Vol. 22, No. 4, Dec. 1996, p. 469-483. [Available at <ftp://geom.umn.edu/pub/software/qhull-96.ps>].
- [48] Shewchuk J.R. Triangle: Engineering a 2d quality mesh generator and delaunay triangulator. *First ACM Workshop on Applied Computational Geometry*. 1996.
[Available at <http://www.cs.cmu.edu/~quake/triangle.html>].
- [49] M. Bern, D. Eppstein. Mesh Generation and Optimal Triangulation. *Dept. of Information and Computer Science, University of California Irvine*, California, USA.
- [50] Stan Birchfield and Carlo Tomasi. Multiway Cut for Stereo and Motion with Slanted Surfaces. *Proceedings of the Seventh IEEE International Conference on Computer Vision*, Kerkyra, Greece, September 1999.
- [51] Vladimir Kolmogorov and Ramin Zabih. Computing Visual Correspondence with Occlusions using Graph Cuts. In: *International Conference on Computer Vision*, July 2001. <http://www.cs.cornell.edu/rdz/Papers/KZ-ICCV01-tr.pdf>
- [52] Т. Кормен, Ч. Лейзерсон, Р. Ривест. Алгоритмы: построение и анализ. – М.: МЦНМО, 2001.
- [53] C. Bron and J. Kerbosch. Algorithm 457: Finding All Cliques of an Undirected Graph. *CACM*, 16(9):575-577, 1973.