

## ПОСТРОЕНИЕ ОПТИЧЕСКОЙ ОБРАТНОЙ СВЯЗИ ДЛЯ УПРАВЛЕНИЯ РУКОЙ РОБОТА С ИСПОЛЬЗОВАНИЕМ ВИДЕОКАМЕРЫ

**Введение.** При создании человекоподобных роботов встает задача управления рукой робота. Отдельно взятую руку робота можно рассматривать как манипулятор с семью степенями свободы от плеча до ладони руки [1]. Степени свободы, связанные с пальцами упрощенно не рассматриваются в задаче инверсной кинематики, поскольку они влияют только на форму схвата, а не на общую ориентацию руки. В свою очередь форма схвата должна определяться в зависимости от вида предмета, который предстоит взять. Так для антропоморфной руки выделяют до 30 различных конфигураций пальцев и положения ладони относительно предплечья. Каждый из них предназначен либо для захвата определенных предметов, либо для выполнения «сервисных» операций. Для работы руки робота требуется определить местоположение объектов окружающей среды.

### Постановка задач исследования.

Задачу управления наводимой по видеокамере рукой можно разделить на несколько этапов.

1. Определение геометрических параметров руки, решение задачи инверсной кинематики.
2. Распознавание объекта по его изображению
3. Определение координат объекта в глобальной системе координат для наведения руки для схвата объекта.
4. Разработка системы управления рукой робота в динамике с учетом конкретных двигателей и механических передач.

Задача управления рукой робота в динамике в данной статье не рассматривается.

### 1. Определение геометрических параметров руки, решение задачи инверсной кинематики.

Существует несколько альтернативных способов описания кинематики манипуляторов [2,3]. Наиболее распространенным является т.н. описание Денавита Хартенберга, представляющее манипулятор как последовательное соединение звеньев, отличающихся друг от друга только главной осью вращения, смещением, углом закручивания звена и длиной звена [2,3].

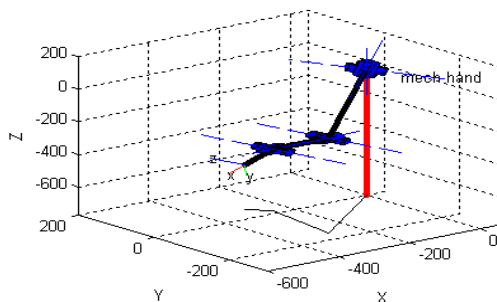
Таким образом можно записать параметры виртуальной руки в виде табл. 1

Таблица 1. Параметры Денавита Хартенберга для руки робота

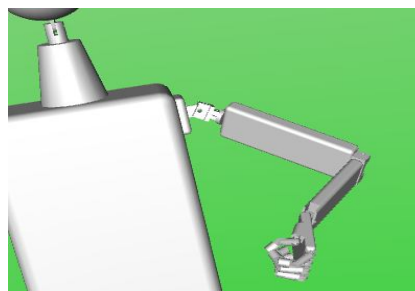
Звено	скручивание (Twist), Рад	смещение (offset), мм	Угол поворота, рад	Длина звена, мм
1	$\pi/2$	0	0	0
2	$-\pi/2$	0	0	0
3	$\pi/2$	0	0	337.4
4	$-\pi/2$	0	0	0
5	0	0	0	273.9
6	$\pi/2$	0	0	0
7	0	123	0	0

Как можно заметить, некоторые звенья имеют нулевую длину, это вызвано тем, что в представлении Денавита Хартенберга каждое звено должно иметь только одну степень свободы. Для сохранения соответствия с реальной системой, имеющей три степени свободы в одной точке для плечевого соединения, требуется введение таких псевдозвеньев, которые имеют нулевое смещение и длину, однако это не сказывается на расчетах кинематики системы [3,4].

На рис.1 представлена кинематическая схема руки робота и его модель в формате VRML для 3D визуализации в среде MATLAB.



а)



в)

Рис. 1. а) Кинематическая конфигурация руки робота в) 3D Визуализация руки с пятью пальцами

Существует множество методов решения обратной задачи кинематики [3,4,5], отвечающей на вопрос, какими должны быть величины углов в каждой степени свободы, чтобы конечное звено находилось в заданной точке пространства с заданной ориентацией. Для манипуляторов, имеющих более 6 степеней свободы, задача имеет бесконечное множество решений. Однако наиболее простым и эффективным способом решения этой задачи можно назвать метод дифференциальной кинематики [5], основанный на расчете Якобиана, представляющий собой отношение приращений степеней свободы манипулятора, к приращению координат конечного звена в глобальной системе координат. Единственное условие надо задать начальное положение

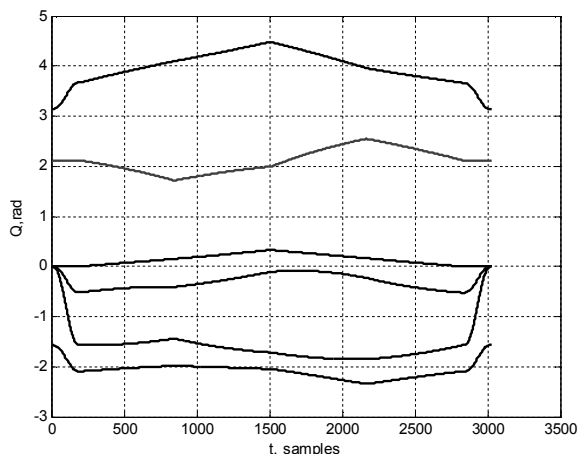


Рис. 2. График изменения углов во всех суставах

На рис. 3. представлена траектория конечного звена в глобальных координатах, как видно, рабочая точка

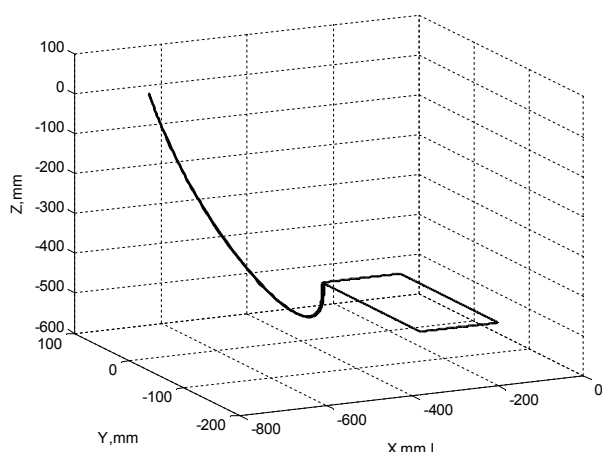


Рис. 3. Траектория конечного звена, движение вдоль прямоугольника с заданной ориентацией

сначала выходит из исходного положения в заданный угол прямоугольника по полиномиальной траектории [2,3], затем точка движется вдоль заданного прямоугольника с начальной ориентацией, как в первой точке прямоугольника. После того, как периметр фигуры был описан, точка возвращается в исходное положение по траектории, заданной полиномом.

Таким образом мы можем переместить схват руки робота в произвольно заданную точку с сохранением заданной ориентации.

## 2. Распознавание объекта по его изображению.

Для решения этой задачи существует огромное количество методов [8], такие как методы, основанные на сравнении контуров, или деформируемых моделей, или сравнении эталонов (template matching) [8]. Сегодня идет работа над тем, чтобы ослабить некоторые ограничения существующих алгоритмов распознавания и

добиться их устойчивой работы в случае изменения освещенности и других переменных параметров: деформации, изменения положения камеры при распознавании трехмерных объектов (проективные искажения). В данной работе предлагается использовать распознавание образов с помощью дерева решений на основе признаков Хаара [8,9], поскольку этот метод имеет исключительные достоинства с точки зрения устойчивости к помехам в изображении и изменчивости самих объектов [8,9].

Признаки Хаара представляют собой коэффициенты вейвлет преобразования изображения с базисом Хаара [8,9]. Рекуррентное преобразование Хаара имеет два ценных практических свойства:

- при хранении и передаче больших объемов информации степень ее сжатия и подробность представления можно варьировать в широких пределах, запуская (не передавая и не сохраняя) поправочные коэффициенты  $d$ , начиная с требуемого уровня подробности;

- передачу сигнала можно осуществлять последовательно, начиная с самых грубых уровней, несущих в предельно компактной форме основной объем информации, с дальнейшим последовательным уточнением, вплоть до полного восстановления. На нужном уровне подробности передачи данного фрагмента (кадра) можно прекратить и перейти к следующему фрагменту [8,9]. Для выделения области интереса используется следующий алгоритм, работающий на основе преобразования Хаара. Способ выделения объектов

первоначально предложен П. Виолой [9]. Сначала классификатор (*cascade of boosted classifiers working with haar-like features*) обучается на нескольких сотнях видов образца конкретного объекта, которые являются положительными примерами. Все изображения должны быть приведены к стандартному виду (по параметрам освещенности, положению объекта и размеру). Также классификатору должны быть предъявлены отрицательные образцы: произвольные изображения того же размера, но не содержащие лиц. После предварительного обучения классификатора он может применяться к области интереса (того же самого размера, который был использован при обучении) во входном образе. Классификатор принимает решение "1" если область, с некоторой вероятностью, является требуемым объектом (лицом), и "0" - в противном случае. Для поиска объекта на фотографии классификатор применяется к фрагменту изображения, перемещаемому по фотографии. Принцип работы классификатора позволяет его применение к объектам различного размера без их предварительного масштабирования. Для этого процедура сканирования должна быть проведена несколько раз с разными весовыми коэффициентами. Классификатор состоит из различных более простых классификаторов (этапов), которые последовательно применяются к области интереса, пока на некотором этапе кандидат не отвергается или пока все этапы не закончены успешно. Классификаторы являются деревом решений не менее, чем с 2 уровнями. Итак путем использования изложенного выше алгоритма можно не только детектировать объект на изображении, но и определить его относительный размер, что дает примерную информацию о расстоянии до объекта. Метод так же работает и при наличии множества объектов на одном изображении. Распознавание объектов реализовано в библиотеке с открытыми ресурсами OpenCV [10].

#### 4. Определение координат объекта в глобальной системе координат для наведения руки для схвата объекта.

Для наведения руки на объект требуется определить его положение в глобальной системе координат. Для этого напрямую предназначены методы эпиполярной геометрии [8,10,11], позволяющие по нескольким изображениям одной и той же сцены получить перспективную проекцию и из нее вычислить координаты точек в пространстве. Однако на практике эти методы достаточно трудоемки и в реальном масштабе времени не позволяют получить приемлемой точности. Поэтому в случае сложного взаимодействия с окружающей средой методы эпиполярной геометрии, дающие сведения о расстоянии от предметов до камеры, должны все равно комбинироваться с методами распознавания образов. Процесс определения координат объектов начинается с калибровки камеры [11]. В результате калибровки определяются внутренние и внешние параметры камеры, позволяющие рассчитать положение объекта в глобальной системе координат.

Для восстановления трехмерной модели объекта по изображению с нескольких камер чаще всего используют модель камеры-обскуры (pinhole camera). При моделировании камеры-обскуры светонепроницаемый ящик заменяют системой «точка наблюдения – плоскость изображения».

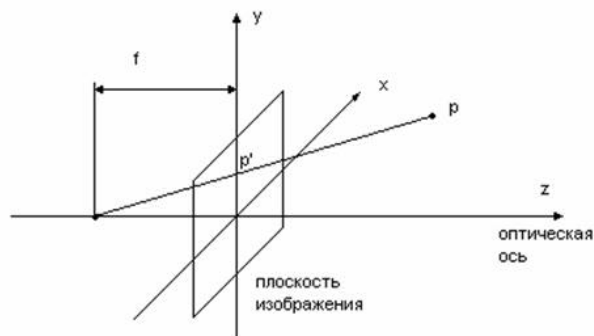


Рис. 4. Система «точка наблюдения – плоскость изображения»

Между координатами точки  $P(X, Y, Z)$  в пространстве сцены и точкой  $(x, y)$  в плоскости изображения камеры-обскуры существует следующее соотношение:

$$\begin{aligned} x &= f \cdot X / Z, \\ y &= f \cdot Y / Z, \end{aligned}$$

где  $f$  – фокусное расстояние камеры.

При использовании реального оборудования возникает следующая проблема: линзы цифровых камер дают радиальные искажения, которые могут быть смоделированы следующими уравнениями:

$$\begin{aligned} x &= x_d(1 + k_1 \cdot r_d^2) \\ y &= y_d(1 + k_1 \cdot r_d^2) \end{aligned}$$

где  $(x_d, y_d)$  – координаты искаженного изображения,  $r_d = \sqrt{x_d^2 + y_d^2}$ ,  $k_1$  – коэффициент искажения линзы. Пусть  $(i, j)$  – дискретные координаты буфера кадра с началом в верхнем левом углу,  $(O_x, O_y)$  – фокусная точка линзы (место пересечения оптической осью плоскости изображения) в координатах буфера кадра и  $(x, y)$  – координаты изображения.

Отношение координат плоскости изображения и координат буфера кадра:

$$\begin{aligned} x &= (i - O_x) \cdot S_x; \\ y &= (j - O_y) \cdot S_y, \end{aligned}$$

где  $S_x$ ,  $S_y$  – горизонтальное и вертикальное расстояние двух соседних пикселей в буфере кадра в единицах глобальной системы координат. Оцифрованное изображение хранится в буфере кадра, который можно рассматривать как матрицу пикселей размером  $W \times H$ . В зависимости от используемой математической модели камеры параметры, которые необходимо определить, различны. В нашем случае при использовании модели камеры-обскуры все параметры, которые необходимо определить, можно разделить на две категории: внутренние и внешние параметры.

*Внутренние параметры:* внутренняя геометрия и оптические характеристики линз, параметры устройства отображения.

В используемой нами модели камеры существуют следующие внутренние параметры, которые необходимо определить:

$f$  – фокусное расстояние камеры,  $S_x$ ,  $S_y$  – горизонтальное и вертикальное расстояние двух соседних пикселей в буфере кадра,  $k_1$  – коэффициент искажения линзы,  $O_x$ ,  $O_y$  – фокусная точка линзы.

Как правило, в фотометрических системах фокусное расстояние камеры  $f$  (обычно это фокусное расстояние линзы объектива) известно из спецификации камеры, а за величины  $S_x$ ,  $S_y$  принимается расстояние между соседними элементами ПЗС матрицы. Такие системы используются для определения очень точных пространственных координат объектов реального мира (например задачи пространственного ориентирования роботов, картографирования местности), но в контексте нашей системы этого не требуется, т.к. все данные, полученные с исследуемого объекта движения в конечном счете будут переноситься на виртуальную модель, которая может иметь какие угодно размеры. Основное требование – это сохранение пропорций расположения маркеров. Следовательно, все расчеты мы можем производить в собственных системных единицах. Кроме того, система рассчитывается на использование различного типа камер, для которых не всегда можно найти техническую спецификацию (например домашние web-камеры). Таким образом, расстояние двух соседних пикселей в буфере кадра  $S_x$ ,  $S_y$  мы можем определить любым в нашей системе. Для простоты дальнейших вычислений примем его за единицу:  $S_x = 1$ ,  $S_y = 1$ . Вследствие этого допущения нам теперь необходимо определить фокусное расстояние, но для того чтобы это корректно сделать, прежде необходимо скорректировать искажение линзы объектива камеры, которое описывается тремя параметрами: фокусной точкой линзы ( $O_x$ ,  $O_y$ ) и коэффициентом  $k_1$ . Следует отметить, что калибровка искажений линз камер является сложной математической задачей. Существует много методов ее решения (обратная фотограмметрическая засечка, решение замкнутых форм, полная нелинейная оптимизация). В реальных системах отклонение фокусной точки линзы от геометрического центра изображения составляет  $\pm 0,2\%$ , что, допустим, при разрешении  $640 \times 480$  составляет  $0,96$ – $1,28$  пикселя. Данной погрешностью можно пренебречь и принять за фокусную точку линзы геометрический центр изображения.  $O_x = W/2$ ,  $O_y = H/2$ , где  $W$ ,  $H$  – размеры буфера кадра.

Ввиду сложности и долгого времени работы известных алгоритмов коэффициент  $k_1$  подбирается пользователем вручную. Для этого камерой снимается прямоугольная сетка, которая может быть расчерчена на бумаге, а коэффициент  $k_1$  вводится пользователем в диалоговом режиме. При этом камеру направляют таким образом, что бы она «смотрела» в центр сетки.

После того, как исправлено искажение линз, остается найти последний параметр камеры – ее фокусное расстояние. Фокусное расстояние можно найти из соотношения  $x = f * X / Z$ .

Таким образом, мы определили все *внутренние параметры камеры*, и она готова к следующему этапу.

Необходимо отметить, что внутреннюю калибровку камер следует выполнить только один раз, т.к. внутренние параметры камеры не изменяются.

Определение ориентации и расположения камеры в пространстве по изображению, полученному с ее помощью, также называют задачей внешней калибровки камеры или определения ее *внешних параметров*, заданных в виде матрицы однородных преобразований вида  $M = [R_{3 \times 3}, P_{3 \times 1}; 0001]$ , где  $R$  – матрица поворота, а  $P$  – матрица переноса. В итоге матрица  $M$  позволяет перейти от координат камеры к координатам наблюдаемого объекта (точки).

Во многих случаях геометрия некоторых объектов сцены известна заранее. На основании этого рассматривается модель, представляющая собой набор черт и их взаимное расположение. Модель налагает ограничения и сужает область решений задачи. В качестве черт используются определенные точки (вершины), отрезки (границы) и углы объекта. В зависимости от используемых черт выделяются следующие постановки задачи внешней калибровки: PnP проблема (Perspective n Points problem), PnL проблема (Perspective n Lines problem) и PnA проблема (Perspective n Angles problem) [11]. При нахождении внешних параметров камеры задаются известным геометрическим положением точек, которые так же должны быть видны в кадре камеры, далее, используя оптимизационный алгоритм, находят все элементы матрицы  $M$ .

Таким образом, если задаться положением точки в виде  $P = [px, py, pz, 1]^T$  и ее проекцией на плоскость изображения в виде  $I = [u, v, 1]^T$ , то матрицу переноса можно выразить в виде  $\lambda \cdot I = M \cdot P$ , где  $\lambda$  – некоторая константа.

Матрицу  $M$  можно выразить в виде:

$$M = K \cdot Ip \cdot G = \begin{bmatrix} fx, 0, 0x \\ 0, fy, 0y \\ 0, 0, 1 \end{bmatrix} * \begin{bmatrix} 1, 0, 0, 0 \\ 0, 1, 0, 0 \\ 0, 0, 1, 0 \\ 0, 0, 0, 0 \end{bmatrix} * \begin{bmatrix} R_{3 \times 3}, t_{3 \times 1} \\ 000, 1 \end{bmatrix}, \quad (1)$$

где матрица  $R_{3 \times 3}$  может быть выражена через матрицу простых поворотов  $r = [r_x, r_y, r_z]^T$ ,  $t = [t_x, t_y, t_z]^T$  - вектор параллельного переноса.

Для наглядности предположим, что  $K$  является единичной матрицей. Пусть вектор  $p$  определяет положение некоторой точки  $P$  модели объекта в системе координат камеры. Тогда координаты проекции  $P$  на картинную плоскость вычисляются по формулам:

$$u = \frac{p \times r_x + t_x}{p \times r_z + t_z};$$

$$v = \frac{p \times r_y + t_y}{p \times r_z + t_z}. \quad (2)$$

Из формулы 2 видно, что по одному проективному изображению трехмерной точки невозможно восстановить координаты в трехмерном пространстве из за неопределенности  $t_z$ . Однако, из алгоритма распознавания объектов можно получить сведения о величине  $t_z$ , а матрица  $R$ ,  $t_x, t_y$  определяется при калибровке. Таким образом мы получаем полное описание матрицы связи координат изображения с координатами реального объекта. В этом случае на определении дальности будет сказываться погрешность определения размера объекта. Эта проблема является темой дальнейших исследований. Так же величину дальности объекта  $t_z$  от камеры можно вычислить по нескольким изображениям одной сцены, в частности по т.н. стереопаре – двум изображениям, полученным из двух разных точек при небольшом отклонении положений камеры. Стереопару можно получить либо с неподвижных двух камер, либо с одной подвижной камеры.

На рис. 5 представлен результат работы программы, созданной на базе библиотеки OpenCV [10]. Полученные в реальном времени кадры с двух видеокамер сравниваются друг с другом, производится автокалибровка и далее вычисляется карта глубин(слева), где более светлые участки обозначают, что точки находятся ближе к камерам. Более высокое качество карты глубин можно получить при тщательной настройке и предварительной калибровке камер. Большое влияние имеет условие, что кадры камеры должны максимально совпадать (на приведенной ниже иллюстрации это не так).

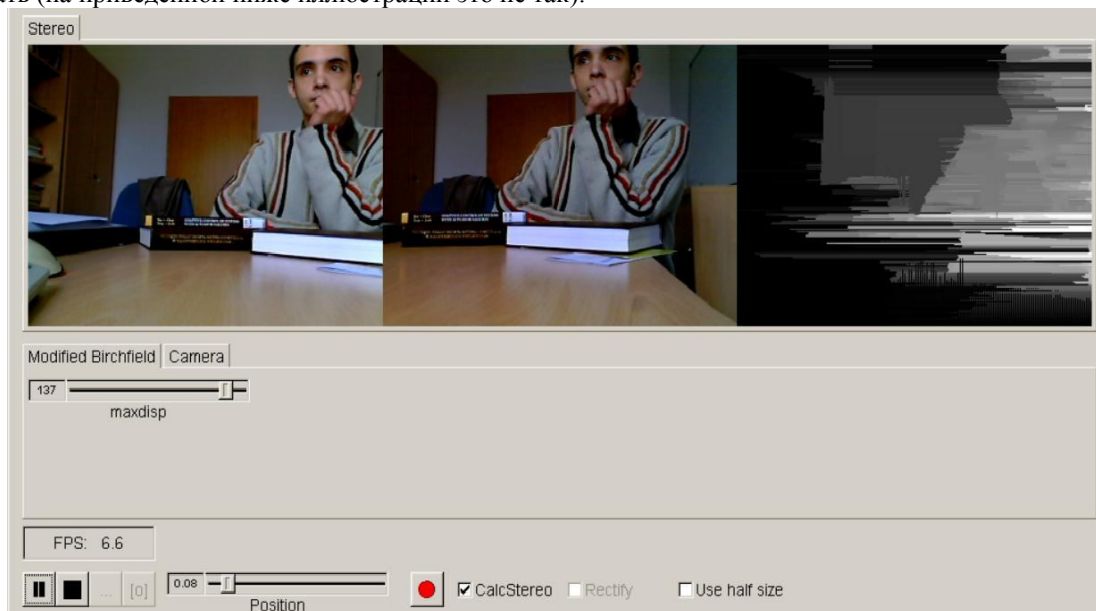


Рис. 5. Построение карты глубин с автокалибровкой камер со скоростью 6.6 кадров в секунду

Существует несколько способов определения и задания как матрицы внешних параметров камеры, так и формул связи координат объекта с координатами проекции [11], однако они отличаются только разными допущениями и наложенными ограничениями на основную формулу связи координат (1) объекта и его проекции.

Как правило, матрица перехода между базовой системой координат и системой координат камеры заранее известна и также выражается в виде матрицы однородных преобразований  ${}^B T_K$ . В свою очередь из расчета прямой кинематики руки робота легко получить матрицу преобразования базовой системы координат, в координаты конечного звена руки робота  ${}^B T_E$ . Матрица перехода от координат объекта к координатам камеры равна  $M$ , то есть  $M = {}^O T_K$ . Для наведения руки требуется найти матрицу перехода от конечного звена руки к объекту, то есть  ${}^E T_O$ . Таким образом, основываясь на свойстве матриц преобразования координат, что система1  $T_{система2} = (система2 T_{система1})^{-1}$ , можно записать:

$${}^B T_K = {}^B T_O \cdot {}^O T_K; \quad (3)$$

$${}^B T_O = {}^B T_K \cdot {}^K T_O = {}^B T_K \cdot ({}^O T_K)^{-1} = {}^B T_K \cdot M^{-1}; \quad (4)$$

$${}^E T_O = {}^E T_B \cdot {}^B T_O = ({}^B T_E)^{-1} \cdot {}^B T_K \cdot M^{-1}. \quad (5)$$

И так, мы получили конечное выражение матрицы преобразования координат  ${}^E T_O$  конечного звена манипулятора (руки робота) к координатам объекта (5). И, следовательно, динамическую ошибку наведения схвата на объект можно записать в виде  $E = {}^B T_O^* - {}^B T_E$ , где  ${}^B T_O^*$  - идеальная трансформация от глобальной системы координат к координатам объекта,  ${}^B T_E$  - текущая трансформация от глобальных координат к координате схвата руки робота.

**Выводы.** В статье указывается способ описания и решения задачи инверсной кинематики для руки робота с семью степенями свободы. Рекомендован метод распознавания объектов по изображению с видеокamеры в реальном масштабе времени. Представлен в явной форме переход от координат проекции объекта на плоскость изображения к его реальным координатам в глобальной системе отсчета, что является исходным условием для расчета инверсной кинематики в задаче захвата распознанного объекта рукой робота.

### Литература

1. Venkataraman S.T. Iberall T. Dextrous Robot Hands /Springer Verlag, 1990, 345 p.
2. Geoffrey Taylor, Lindsay Kleeman, Visual Perception and Robotic Manipulation: 3D Object Recognition, Tracking and Hand-Eye Coordination (Springer Tracts in Advanced Robotics), 2006, 218 p.
3. Dan Marghitu, Kinematic chains and machine components design, Elsevier Inc. 2005, 790 p.
4. Josephs, Harold. Dynamics of mechanical systems / by Harold Josephs and Ronald L. Huston, by CRC Press LLC, 2002, 777 p.
5. Fundamentals of Robotic Mechanical Systems, theory, methods, and algorithms, Springer Verlag, 2003, 545 p.
6. Edward Tsang, Foundations of Constraint Satisfaction (Computation in Cognitive Science), 1993, 300 p.
7. D.T. Pham, D. Karaboga, Intelligent Optimisation Techniques, Springer, 2000, 302 p.
8. Rafael C. Gonzalez, Richard E. Woods, Steven L. Eddins, Digital Image Processing Using MATLAB, Prentice Hall; 2003, 782 p.
9. <http://research.microsoft.com/~viola/>
10. <http://sourceforge.net/projects/opencvlibrary/>
11. Yi Ma, S. Soatto, J Kosecka, S. Shankar Sastry, An Invitation to 3-D Vision, Springer, 2005, 552 p.