

Департамент образования администрации г. Перми

МБОУ «Лицей №1» г. Перми

Направление: математическое моделирование

***Описание положения объекта в
трехмерном пространстве по двумерному
изображению***

Выполнили:

Наговицына Галина Викторовна, 201кл

Евсеева Екатерина Сергеевна, 201 кл

Научный руководитель:

К. ф.-м. н., доцент ПНИПУ,

Шабрыкина Наталья Сергеевна

Пермь - 2018

Содержание

Abstract.....	3
Описание работы.....	4
Технология захвата движения.....	5
<i>Пример маркеров, изготовленных самостоятельно</i>	6
Постановка задачи.....	7
Обработка экспериментальных данных	8
<i>Прыжок тур</i>	8
<i>Проблемы, обнаруженные при обработке данных</i>	10
<i>Продолжение расчетов с учетом возникшей проблемы</i>	11
<i>Результаты</i>	12
<i>Эксперимент с маятником</i>	13
<i>Результаты</i>	14
<i>Эксперимент с рукой человека</i>	17
<i>Результаты</i>	18
Заключение.....	19
Список используемых источников	20
Приложение 1. Таблицы с координатами маркеров	21
Приложение 2. Таблица с углами отклонения	22
Приложение 3. Программа расчёта координат маркеров	23
Приложение 4. Результаты работы программы расчёта координат маркеров	26
Приложение 5. Программа расчёта координат маркеров	27
Приложение 6. Таблицы с координатами маятника	30
Приложение 7. Таблицы с координатами маркеров	31

Abstract

The research is devoted to the problem of 3D-image creation on the basis of 2D-object.

The main subject of the research is a two-dimensional image, which is the initial one for 3D-image creation.

The aim of the research is to create a mathematical model of a three-dimensional object using data obtained from two-dimensional image.

To achieve the aim it was necessary to accomplish the following tasks:

- to study theoretical literature;
- to analyze the main laws of motion;
- to make an experiment;
- to analyze the results of the experiment;
- to develop a programme, which is capable of calculating the third coordinates of the object;
- to draw a conclusion

During the experiment it was found out that the real sizes of components of the subject, the sizes of these components in each chosen period of time should be known for reconstruction of three-dimensional model. Angles of deviation from the horizontal plane should be also calculated.

It should be noted that reconstruction of 3D-model is possible only for perfect multi-linked models.

In conclusion, it should be noted that the objectives set in the beginning of the research have been accomplished, namely: all the necessary calculations have been done, the model of 3D-object has been created.

Описание работы

Технология захвата движения (motion capture) активно используется во многих сферах: от игровой и киноиндустрии как метод анимации персонажей и объектов до применения в науке. Чаще всего эта технология используется крупными коммерческими или государственными организациями, у которых имеется достаточный бюджет, а следовательно и доступ к ней, так как она требует больших материальных затрат. Высокая стоимость систем существенно ограничивает возможность использования захвата движения небольшими организациями или же отдельными людьми.

Если простейшие маркеры, используемые при захвате движения, еще можно изготовить самостоятельно, то для считывания данных с маркеров требуется сложная система камер, обеспечивающая видимость каждого маркера, уменьшающая вероятность их перекрытия и помогающая создать модель в 3D, стоимость которой очень велика, и самостоятельно такое большое количество камер практически невозможно синхронизировать. В этом случае можно использовать вообще одну камеру, но тогда появляется проблема с построением модели, так как полученное изображение будет двумерным. Двумерные модели используются, но не так часто, например, при исследовании походки человека. Основной недостаток 2D-изображения состоит в том, что по нему трудно представить, как объект изменяется в пространстве, и у него нет преимуществ трехмерной модели, а именно большей информативности, широкого диапазона движений.

Целью данной работы является нахождение способа полученную на опыте с использованием захвата движения двумерную модель преобразовать в трехмерную. Объектом исследования будет являться модель движения человека.

В работе будет использоваться математическая модель - идеальная научная знаковая формальная модель, описанная на языке математики и исследование которой проводится с использованием тех или иных математических методов.

Процесс построения и изучения математических моделей называется математическим моделированием [1]. Все естественные и общественные науки, использующие математический аппарат, по сути, занимаются математическим моделированием: заменяют объект исследования его математической моделью и затем изучают последнюю. Математические модели позволяют нам описать и изучить различные явления.

Технология захвата движения

Началом индустрии motion capture с некоторой натяжкой можно считать 1915 год [2]. Тогда новатор анимации Макс Флейшер изобрел технику ротоскопирования. Сперва он снимал живого человека, а потом фильм проецировался на стекло и художник кадр за кадром все обрисовывал. Следующий скачок технология сделала в 70-х, благодаря врачам-кинезиологам, в чьи обязанности входит изучение и лечение опорно-двигательного аппарата. Том Калверт, профессор кинезиологии и компьютерных наук предложил использовать потенциометры, прикрепляя их к различным суставам на человеческом теле, затем данные приборов считывались и с помощью компьютера оцифровывались движения.

Со временем технология развивалась, появились системы, основанные на светодиодных маркерах и светоотражающих элементах, и сейчас без её использования тяжело представить любую крупнобюджетную компьютерную игру или фильм, так же как и некоторые сложные научные исследования.

Как правило, процесс захвата выглядит так: на особой съемочной площадке устанавливаются камеры, считывающие круглые маркеры с костюмов актеров. Результаты накладываются на трехмерные модели.

Разновидностей систем захвата движения сегодня достаточно много. Самые популярные - маркерные. Среди них есть пассивные, когда шарики просто отражают посланный им инфракрасный свет, и активные, где вместо светоотражающих маркеров используются светодиоды с собственными процессорами и радиосинхронизацией [3]. Еще применяют магнитные маркеры, гироскопические и даже целые механические костюмы. Есть и вовсе безмаркерные системы, где используются дорогие и сложно устроенные камеры. У каждой системы есть преимущества и недостатки. Пассивные, например, очень точны, но если маркер чем-нибудь перекрывается и не попадает в объектив камеры, сигнал не считывается, а у магнитных любые помехи сказываются на качестве захвата.

Пример маркеров, изготовленных самостоятельно

Примером самодельных маркеров являются маркеры, использованные Елизаветой Лядовой в ее эксперименте [4, 5], подробнее о котором будет рассказано ниже. Маркеры являются оптически активными и изготовлены из теннисных шариков и светодиодов. Для крепления маркеров на человеке были использованы две системы: с применением спортивных суппортов (рис. 1) и с использованием эластичного жакета болеро (рис. 2).



Рис. 1



Рис. 2

При использовании обеих систем возможно смещение маркеров при движении, что может сильно повлиять на результаты некоторых расчетов.

Постановка задачи

При описании движения человека часто используются многозвенные модели, в которых части тела представляют собой недеформируемые твердые тела. Как частный случай можно использовать модель, состоящую из следующих частей: шара (голова), усеченных конусов (корпус и ноги) и четырех стержней (плечо и предплечье правой и левой руки) (рис. 3). Стержни прикреплены к корпусу и друг к другу с помощью идеальных шаровых шарниров. Голова по отношению к корпусу неподвижна.

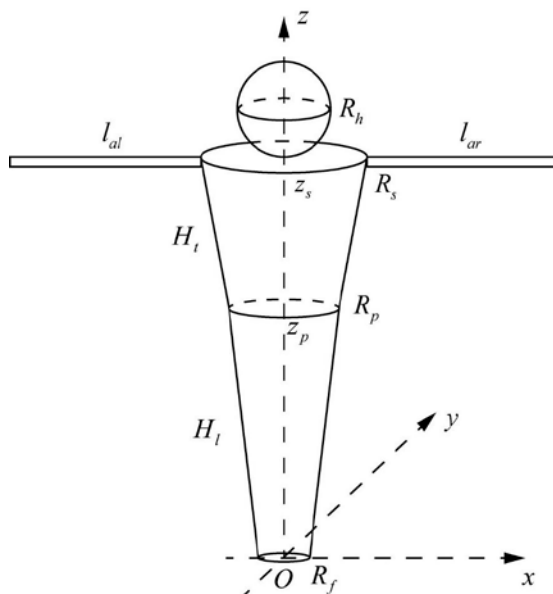


Рис. 3. Модель тела человека

По двумерному изображению движения можно определить координаты ключевых частей, их длину, по изменению которой можно вычислить углы отклонения этих частей от начального положения.

Чтобы было возможно восстановить трехмерную модель движения, должен выполняться перечень определенных условий:

1. Известны начальные размеры исследуемых объектов.
2. Известен стереотип движения.
3. Присутствует хотя бы одна точка, неподвижная относительно третьей оси координат, получение координат относительно которой является целью исследования.
4. В начальный момент времени вся многозвенная модель находится в плоскости, перпендикулярной оптической оси камеры.

Обработка экспериментальных данных

Прыжок тур

При использовании вышеописанной модели было рассмотрено конкретное движение человека, а именно движение спортсмена при исполнении прыжка тур в фигурном катании (рис. 4), изученное ранее Елизаветой Лядовой. Основное внимание в её работе было сфокусировано на движении корпуса и рук спортсмена.



Рис. 4. Порядок выполнения прыжка тур в один оборот

Прыжок был снят одной камерой сверху (рис. 5), следовательно полученное ей изображение было двумерным.



Рис. 5. Кадр из видео, на котором видны маркеры.

В вышеупомянутом исследовании была разработана программа в математическом пакете Wolfram Mathematica [6], позволяющая по видеосъемке эксперимента определять координаты маркеров. В данном случае маркеры были расположены на суставах рук (запястья, локти, плечи). Были получены координаты данных маркеров в разные моменты времени (Приложение 1). По способу, описанному ранее, можно рассчитать углы отклонения частей рук от горизонтальной плоскости и впоследствии восстановить трехмерную модель движения. Основное внимание сконцентрировано на изучении отклонения рук относительно плечевых суставов, поэтому изменение положения тела относительно земной поверхности не влияет на результаты исследования.

На рис. 6 схематично изображено положение частей рук и их проекции, где А и D – плечевые суставы, AD – ширина плеч, В и Е – локти, С и F – запястья. Штрихами отмечены проекции на горизонтальную плоскость.

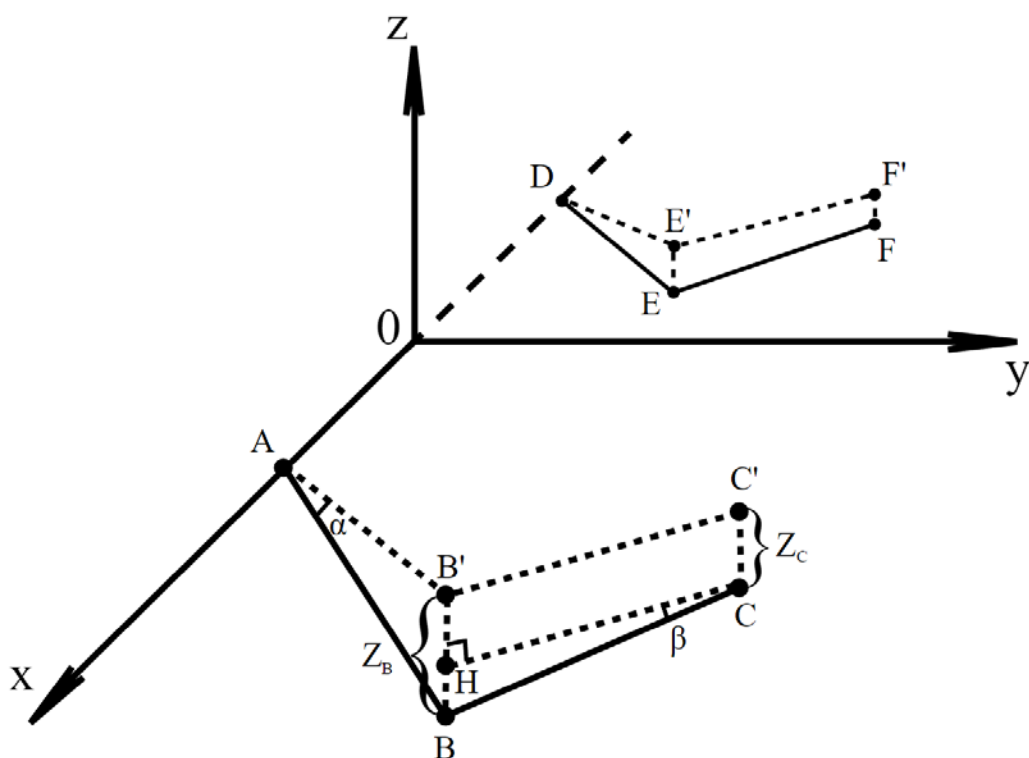


Рис. 6. Расчет координат на ось Z

Расчет длины отрезков (плеч, предплечий и ширины плеч) на изображении:

$$d' = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (1.1),$$

где d' - длина отрезков в пикселях.

Был произведен перевод полученных длин отрезков из пикселей в метры:

$$l' = d' \times \frac{l_{AD}}{d'_{0AD}} \quad (1.2),$$

где l' - реальная длина отрезка в метрах, d'_{0AD} – ширина плеч в пикселях в начальный момент времени, l_{AD} - ширина плеч в метрах.

После перевода длин из пикселей в метры можно посчитать косинусы углов отклонения:

$$\cos \alpha = \frac{l'}{l} \quad (1.3),$$

где α - угол отклонения частей рук от плоскости съемки.

Проблемы, обнаруженные при обработке данных

На этапе вычисления косинусов углов отклонения сегментов руки было обнаружено, что многие их значения больше единицы, то есть выходят за область значений данной функции (рис. 7). Это является следствием того, что длины сегментов в начальном положении были меньше расстояний между маркерами во время прыжка.

Правое предплечье		Правое плечо		Плечи	
cos α	α	cos α	α	cos α	α
1,197078964	#ЧИСЛО!	0,941323093	19,72504697	1	0
1,213066648	#ЧИСЛО!	0,958816697	16,50061288	0,991106788	7,64696903
1,217279626	#ЧИСЛО!	0,966715506	14,82417848	0,997026108	4,419854066
1,210759492	#ЧИСЛО!	0,968002176	14,53324999	1,025875008	#ЧИСЛО!
1,184558207	#ЧИСЛО!	0,960483585	16,16095491	1,071269634	#ЧИСЛО!
1,140428545	#ЧИСЛО!	0,942822397	19,46892485	1,130520853	#ЧИСЛО!
1,099942348	#ЧИСЛО!	0,895963321	26,36756533	1,207719323	#ЧИСЛО!
1,047147636	#ЧИСЛО!	0,834080498	33,47978564	1,287530648	#ЧИСЛО!
1,05992551	#ЧИСЛО!	0,78589991	38,19601426	1,338562396	#ЧИСЛО!
1,107265514	#ЧИСЛО!	0,816126066	35,3011426	1,375506735	#ЧИСЛО!
1,178461138	#ЧИСЛО!	0,919479265	23,14992769	1,40204694	#ЧИСЛО!
1,201967246	#ЧИСЛО!	1,026504364	#ЧИСЛО!	1,409105931	#ЧИСЛО!
1,200161227	#ЧИСЛО!	1,091732229	#ЧИСЛО!	1,402271169	#ЧИСЛО!
1,19569668	#ЧИСЛО!	1,107876621	#ЧИСЛО!	1,359563742	#ЧИСЛО!
1,209460892	#ЧИСЛО!	1,120054988	#ЧИСЛО!	1,344417738	#ЧИСЛО!
1,219471456	#ЧИСЛО!	1,101248129	#ЧИСЛО!	1,356637448	#ЧИСЛО!
1,228200275	#ЧИСЛО!	1,069079056	#ЧИСЛО!	1,421904171	#ЧИСЛО!
1,228143045	#ЧИСЛО!	1,005810277	#ЧИСЛО!	1,423468073	#ЧИСЛО!
1,242474847	#ЧИСЛО!	0,955353174	17,18549861	1,413554216	#ЧИСЛО!
1,248129814	#ЧИСЛО!	0,936987309	20,44838248	1,400933834	#ЧИСЛО!

Рис. 7

Причин этого может быть несколько, например:

1. Маркеры не были закреплены неподвижно
2. Возможные ошибки при обработке раскадровки видео в мат. пакете Wolfram Mathematica (см. стр. 6)

Продолжение расчетов с учетом возникшей проблемы

Перевод длин из пикселей в метры был выполнен иначе:

$$l' = d' \times \frac{l_{AD}}{d'_{\max AD}} \quad (1.4),$$

где $d'_{\max AD}$ - максимальная ширина плеч в пикселях.

При таком переводе значения косинусов углов отклонения не будут выходить за область значений функции.

В дальнейшем численные данные используются лишь для отработки метода, так как результаты расчётов не будут соответствовать реальной ситуации.

После перевода длин из пикселей в метры можно посчитать косинусы углов отклонения (Приложение 2), были получены углы отклонения и их тангенсы.

Формула для расчёта координаты локтя:

$$Z_B = AB' \operatorname{tg} \alpha \quad (1.5),$$

где Z_B - координата локтя по оси Z.

Формула для расчёта координаты запястья:

$$Z_C = Z_B - C'B' \operatorname{tg} \beta \quad (1.6),$$

где Z_C - координата запястья по оси Z.

Для расчёта координат в третьей плоскости была написана программа в среде CodeBlocks на языке программирования C++ [7] (Текст программы приведен в приложении 3), благодаря чему исчезла необходимость вручную производить данные расчеты. Для работы программы необходимы лишь координаты по осям OX и OY и исходные размеры ключевых отрезков. Результаты работы программы сохранялись в файл (Приложение 4).

Результаты

В процессе расчетов выяснилось, что хотя проведенный Елизаветой эксперимент верен для её исследования, для нашей работы он не подходит. Численные значения использовались только для обработки метода. В полученных данных с реальностью совпадает стереотип движения.

На рис. 8 изображены построенные по полученным результатам трехмерные графики движения маркеров: а) левого и б) правого плеча; в) левого и г) правого локтя; д) левого и е) правого запястья.

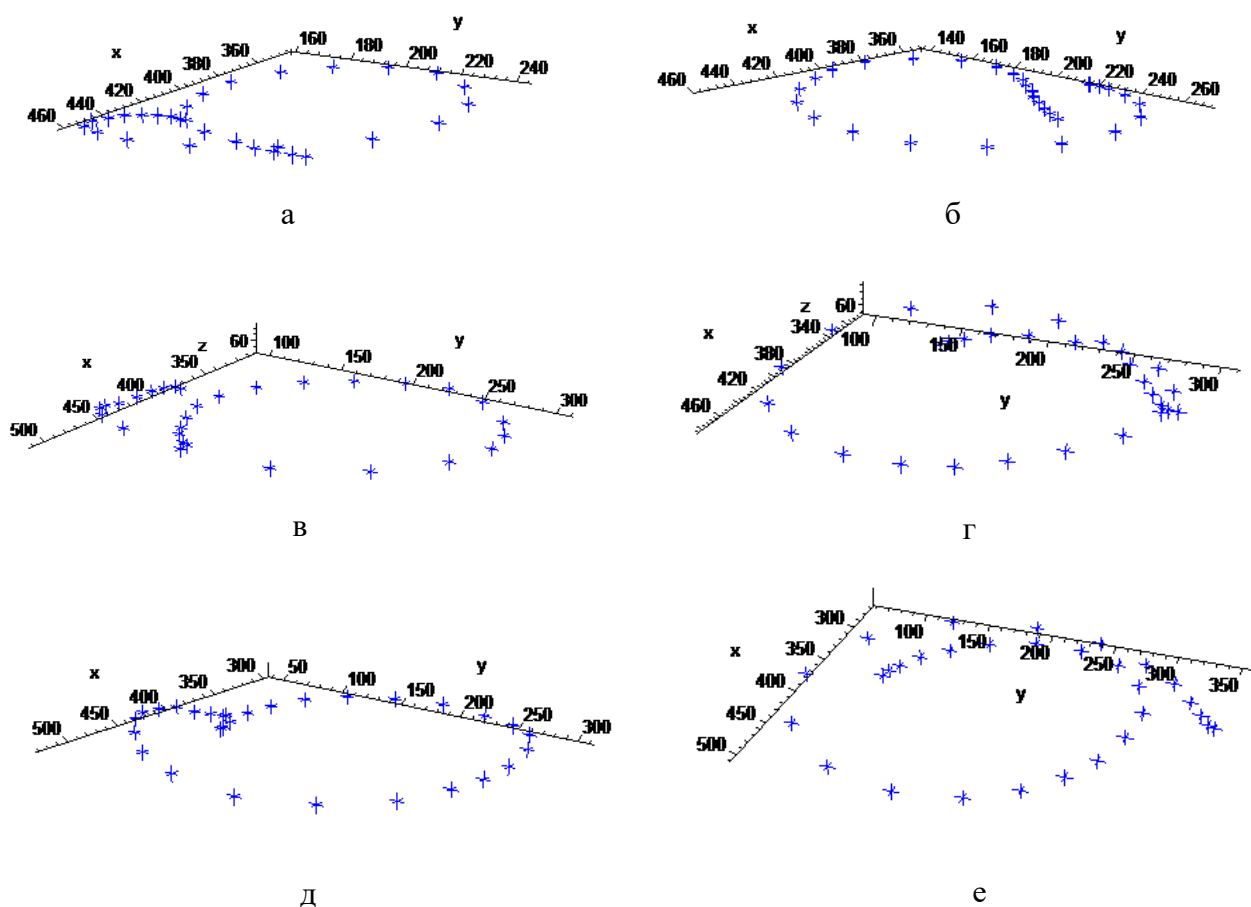


Рис. 8

Эксперимент с маятником

Было решено провести эксперимент с простым, однозвенным объектом для отработки метода, учитывая проблемы, возникшие в предыдущем эксперименте.

К неподвижному креплению с помощью нерастяжимой нити был прикреплен шар (рис. 9), размеры обоих объектов известны. Под шаром была расположена квадратная разметка размерами 12 x 12 см. В состоянии покоя шар находился над центром квадрата. Шар отклоняли от положения равновесия в угол квадрата. По видеозаписи был определен период колебания. Рассматривалось движение маятника за полпериода от одного амплитудного смещения к другому. Для съемки эксперимента использовалась цифровая фотокамера смартфона LG Optimus G.



Рис. 9. Маятник в положении равновесия.

Было произведено несколько опытов, и был выбран самый удачный. Для полученного видео выполнялась раскадровка в программе Free Video to JPG Converter. Дальнейшая обработка получившихся изображений выполнялась в математическом пакете Wolfram Mathematica. Программа определяла координаты неподвижного крепления и координаты шара в разные моменты времени по вертикальной и горизонтальной осям. Получившиеся значения сохранялись в файл.

Для расчёта координат в третьей плоскости была использована немного видоизмененная вышеописанная программа на языке C++ (Текст программы приведен в приложении 5). Результаты работы программы сохранялись в файл.

Формулы, использованные в программе, аналогичны формулам, использованным в первом эксперименте, расчет координат маятника по третьей оси координат выполнялся по формуле:

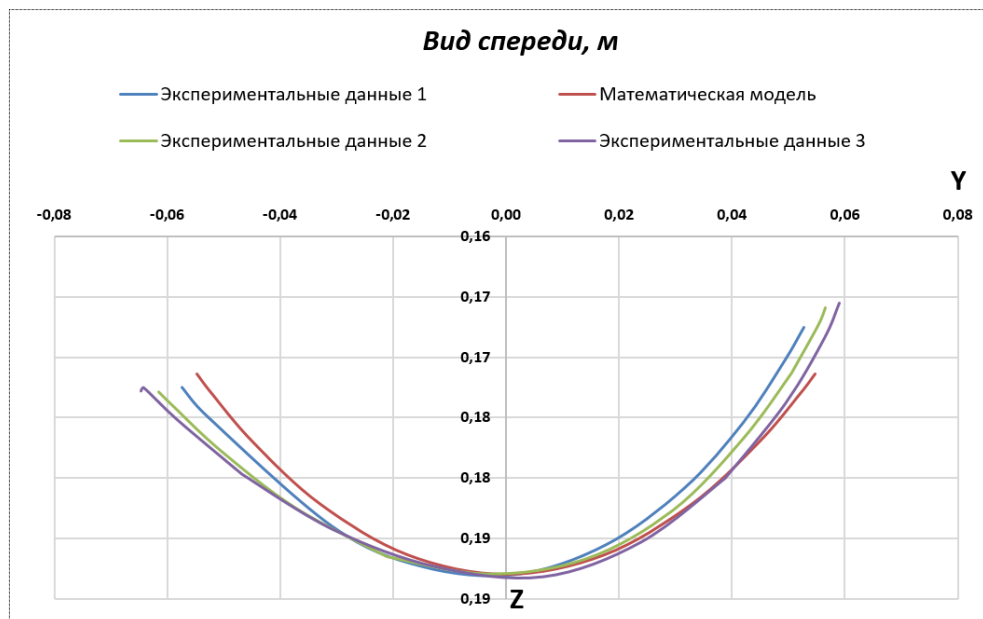
$$x = d' \operatorname{tg} \alpha \quad (2.1),$$

где x - координата запястья по третьей оси, d' - длина проекций в пикселях.

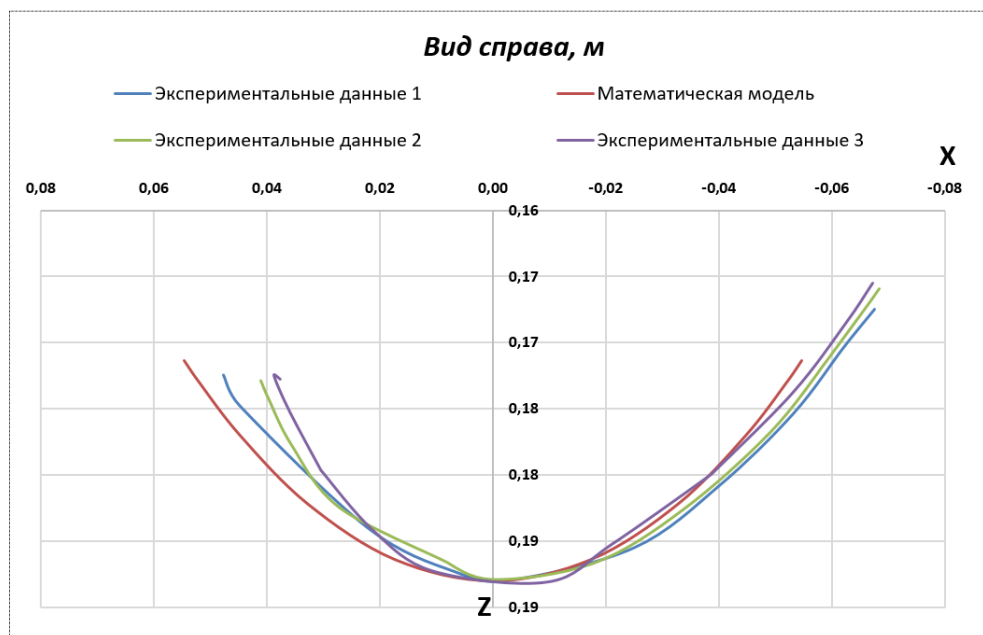
Результаты работы обеих программ приведены в приложении 6.

Результаты

На рис. 10 изображены графики движения маятника в различных координатных плоскостях, построенные по данным нескольких экспериментов. Также была аналитически рассчитана математическая модель движения маятника для проверки адекватности полученных при экспериментах результатов.



а



б

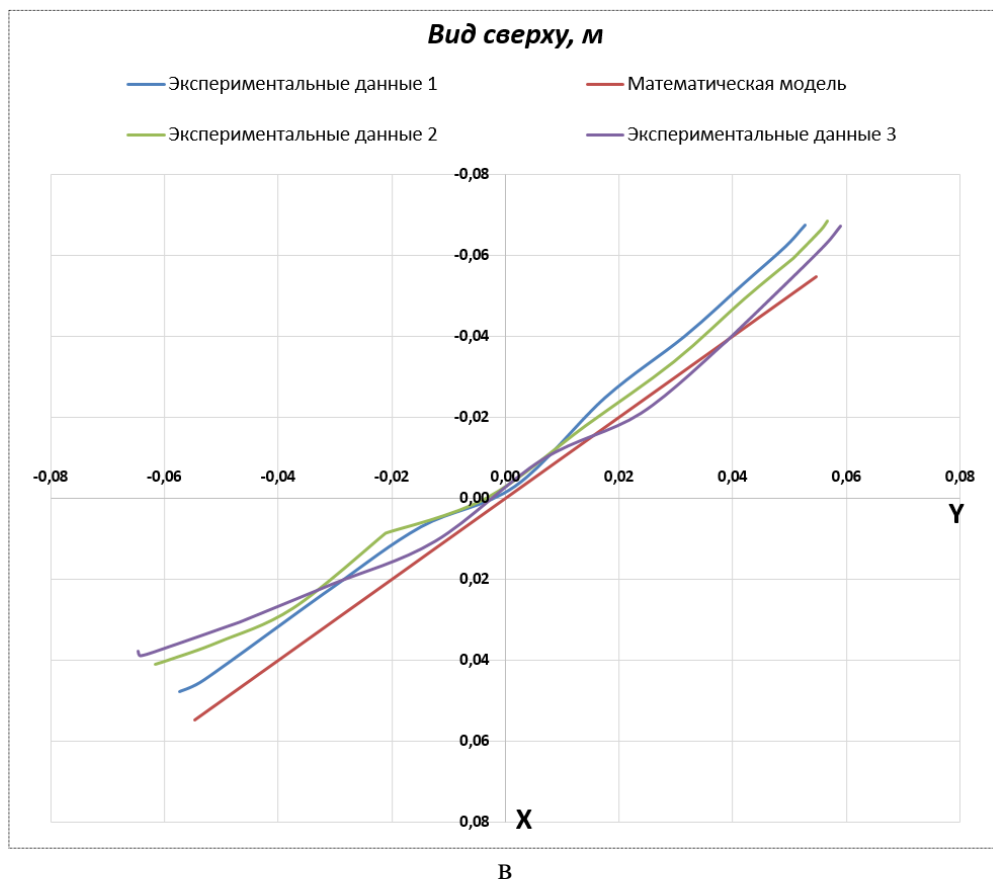


Рис. 10

Можно заметить, что графики, построенные по данным экспериментов, в среднем соответствуют графику математической модели. Возникновение погрешностей обуславливается: затухающим движением маятника, недостатками работы мат. пакета Wolfram Mathematica, некоторыми возможными недостатками постановки эксперимента.

На рис. 11 изображен трехмерный график движения маятника.

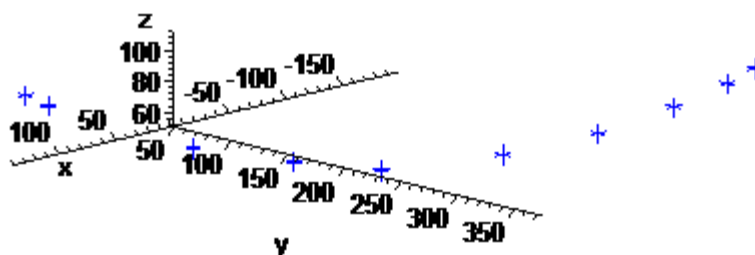


Рис. 11

Координаты в метрах были сопоставлены с разметкой. Так как маятник отклонялся в угол квадрата, расстояния до осей Ox и Oy в начальном и конечном положении должны быть одинаковыми и равны 6 см. В результате эксперимента были получены величины, близкие к этой (см. приложение 6).

Период колебаний можно рассчитать по формуле:

$$T = 2\pi \sqrt{\frac{l}{g}} \quad (2.2),$$

где l – длина подвеса.

Период, полученный экспериментально, равен 0,8 секундам, что сопоставимо с аналитически полученным периодом для данного мат. маятника, равным 0,85 секундам.

Можно проверить угол, под которым отклоняется шар (рис. 12):

$$\tan \alpha = \frac{d'_{12}}{d'_{02}} \quad (2.3)$$

$$\alpha = \tan^{-1} \alpha \quad (2.4)$$

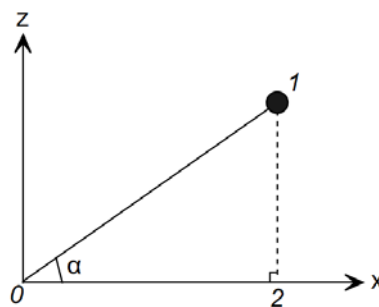


Рис. 12

Угол α равен примерно 52 градусам, что близко к углу по условию эксперимента, равному 45 градусам. Разность этих углов является причиной разницы между аналитической моделью исследуемого колебания и экспериментальной.

Эксперимент с рукой человека

После отработки метода расчета третьей координаты на примере простого, однозвенного объекта, было решено вернуться к идеальной многозвенной модели человека и ее движению. В качестве объекта исследования была выбрана рука человека, модель которой представляет собой два недеформируемых стержня, соединенных между собой идеальным шарниром.

На руке были расположены три маркера из чёрной клеящейся ленты: на плече, локтевом суставе и запястье (рис. 13).



Рис. 13

Рукой было выполнено произвольное движение. Снятое видео было раскадровано, потом обработано в мат. пакете Wolfram Mathematica для определения координат маркеров в плоскости съемки. После этого полученные координаты были заведены в ранее написанную программу на C++, предназначенную для расчета координаты маркеров в трехмерном пространстве по формулам, соответствующим формулам 1.1-1.6, где перевод длин сегментов был произведен по длине предплечья.

Результаты работы обеих программ приведены в приложении 7.

Результаты

На рис. 14 представлены трёхмерные графики изученного произвольного движения руки.

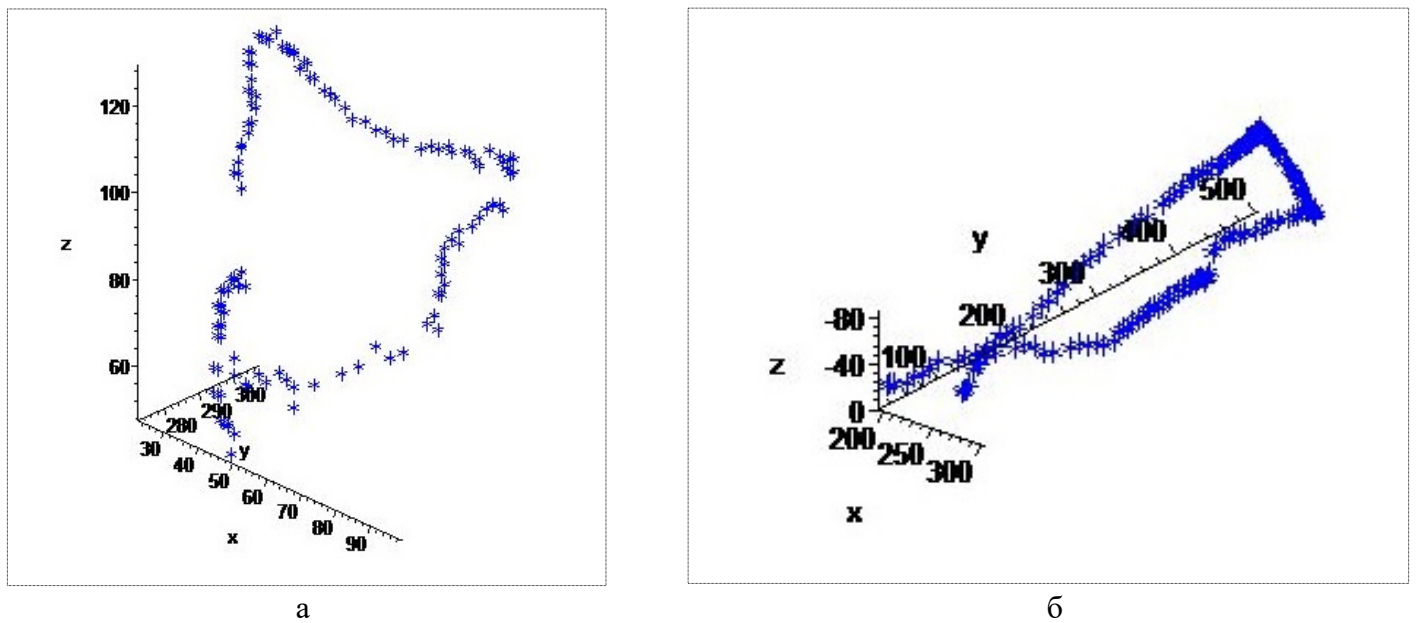


Рис. 14. Движение: а) локтя и б) запястья

По графику видно, что движение представляло собой поднятие и опускание согнутой в локте руки с выведением локтя вперед и заведением запястья назад.

Заключение

В данной работе был получен способ описания положения объекта в трехмерном пространстве по двумерному изображению. Способ был опробован на серии экспериментов, включающих в себя описание положения маятника при выведении его из состояния равновесия и описание положения руки человека при выполнении произвольного движения.

Также в процессе работы были получены требования и рекомендации к проведению экспериментов, выполнение которых обязательно или желательно для использования метода:

1. Маркеры должны контрастировать с объектами съемки, обладать правильной формой (желательно круглой) и крепиться неподвижно
2. Объект должен двигаться медленно и/или частота съемки должна быть достаточно велика, чтобы избежать смазывания маркеров на кадре
3. Исследуемый объект должен быть хорошо освещен

Список используемых источников

- 1) Математическая модель. https://ru.wikipedia.org/wiki/Математическая_модель, февраль 2018
- 2) Технологии захвата движения. <https://www.youtube.com/watch?v=6H2DKTj49MI>, ноябрь 2017
- 3) Захват движения. https://ru.wikipedia.org/wiki/Захват_движения, ноябрь 2017
- 4) Лядова Е.А., Шабрыкина Н.С. Математическое моделирование движения человека при исполнении прыжков в фигурном катании на примере прыжка тур // Сборник тезисов работ участников XI Всероссийской конференции обучающихся «Национальное достояние России» и II Всероссийского молодежного форума «АПК – молодежь, наука, инновации», Москва, 29-31 марта 2017 г. / Под ред. А.А. Румянцева, Е.А.Румянцевой – М.: НС «ИНТЕГРАЦИЯ», 2017. – с. 201-202.
- 5) Лядова Е.А., Шабрыкина Н.С., Анферов С.Д. Математическое моделирование движения человека при исполнении прыжков в фигурном катании на примере прыжка тур // Материалы XXVI Всероссийской школы-конференции молодых ученых и студентов "Математическое моделирование в естественных науках", г. Пермь, 4-7 октября 2017 г. – Пермь, Изд-во ПНИПУ, 2017. – с. 48-53.
- 6) Зеленица А.М. Виртуальный учебник Wolfram Language (Mathematica) / Пер. А.М. Зеленицей. — 2-е изд. — Украина, Киев, 2012-2014.
- 7) Страуструп, Бьёрн. Язык программирования C++ / Бьёрн Страуструп - 4-е изд., перераб. и доп. - Addison–Wesley, 2013. - 1136 с.

Приложение 1. Таблицы с координатами маркеров

Таблица 1. Фрагмент сводной таблицы с координатами маркеров (в пикселях) правого запястья, правого локтя и правого плеча.

	A	B	C	D	E	F	G
1		P3 (пр.зап.)		P2 (пр.лок.)		P1 (пр.пл.)	
2	Номер кадра	x	y	x	y	x	y
3	1	376,24	357,57	398,12	306,75	400,00	248,14
4	2	372,70	352,57	394,96	301,11	396,30	241,39
5	3	362,13	344,80	388,11	294,89	391,63	234,77
6	4	344,87	332,83	377,17	287,13	386,53	227,56
7	5	322,11	314,89	362,60	278,03	381,58	221,28
8	6	297,95	286,93	345,79	264,79	376,36	214,64
9	7	278,12	246,35	328,90	244,02	370,90	207,25
10	8	271,35	193,57	315,15	214,17	365,00	199,50
11	9	288,99	132,60	313,15	175,22	359,50	191,00
12	10	336,09	83,65	330,85	134,56	355,65	178,95
13	11	400,50	63,75	365,24	105,27	357,00	161,96
14	12	459,84	80,27	406,81	96,84	366,83	146,75
15	13	497,81	123,76	444,45	108,57	383,14	138,02
16	14	508,67	175,45	469,72	136,23	400,80	139,83
17	15	500,39	221,70	481,40	169,13	415,33	146,67
18	16	481,60	256,28	482,95	199,93	429,92	156,41
19	17	460,97	279,38	481,62	226,50	444,44	171,24
20	18	440,24	295,22	474,85	250,23	454,06	191,11
21	19	412,88	306,12	459,87	273,09	458,33	213,59
22	20	379,20	307,85	434,61	291,78	455,00	237,09

Таблица 2. Фрагмент сводной таблицы с координатами маркеров (в пикселях) левого плеча, левого локтя и левого запястья.

H		I		J		K		L		M	
L1 (лев.пл.)		L2 (лев.лок.)				L3 (лев.зап.)					
x	y	x	y	x	y	x	y	x	y	x	y
434,55	180,03	418,93	121,24	394,00	79,64						
433,94	175,73	420,20	117,56	394,77	78,23						
434,64	171,94	422,82	112,38	398,73	70,40						
437,17	167,78	426,78	106,08	403,60	60,45						
440,44	164,47	433,11	99,89	411,11	49,73						
444,62	161,77	442,57	95,06	422,24	42,70						
450,26	160,26	454,83	95,06	436,76	42,34						
456,05	162,38	469,26	102,52	454,25	51,30						
459,31	168,94	486,32	119,23	475,76	70,85						
460,68	180,83	502,81	148,31	502,17	102,28						
457,35	199,29	510,50	191,69	523,58	147,42						
445,61	220,06	499,74	239,93	528,76	201,83						
425,03	236,57	469,11	279,93	512,50	252,10						
399,53	243,65	428,00	301,86	480,87	287,61						
375,59	241,34	391,07	306,78	448,43	303,60						
356,50	229,50	361,75	299,23	420,21	308,81						
344,70	214,18	336,66	284,71	392,26	308,58						
345,54	197,44	317,43	261,48	361,98	303,52						
355,07	182,13	307,36	233,24	333,97	288,19						
370,44	171,54	307,13	203,13	313,86	262,92						

Приложение 2. Таблица с углами отклонения

Таблица 3. Фрагмент сводной таблицы с углами отклонения правого предплечья, правого плеча и их косинусами.

Правое предплечье		Правое плечо	
$\cos\alpha$	α	$\cos\alpha$	α
0,840959475	32,75842303	0,661288518	48,60178329
0,852190977	31,54922216	0,673577943	47,65618355
0,855150635	31,22361977	0,679126932	47,22454392
0,850570178	31,72626071	0,680030831	47,15394768
0,832163523	33,67837015	0,674748948	47,56534235
0,801162012	36,75878997	0,662341794	48,5212832
0,772720069	39,40121897	0,629422843	50,99244626
0,735631277	42,63941489	0,585949565	54,12989925
0,744607856	41,87457219	0,552102239	56,48864477
0,777864664	38,93452034	0,573336404	55,01678739
0,827880274	34,11839735	0,645943019	49,76358333
0,844393541	32,39298311	0,72112918	43,85221351
0,843124795	32,52842366	0,766952382	39,91900132
0,839988408	32,86110447	0,778293973	38,89536255
0,849657899	31,82551997	0,786849392	38,10795057
0,856690416	31,05301057	0,773637393	39,31834324
0,862822495	30,36501491	0,75103831	41,31960024
0,862782291	30,36957149	0,706591385	45,04174657
0,872850519	29,2084029	0,671144785	47,84451852
0,876823188	28,73851078	0,658242589	48,83402005

Приложение 3. Программа расчёта координат маркеров

```
#include <iostream>
#include <cmath>
#include <fstream>
using namespace std;
#define PI 3.14159265
int main()
{   int k;
    float X_Lplecho[30], X_Llok[30], X_Lzap[30], X_Pzap[30], X_Plok[30],
X_Pplecho[30];
    float Y_Lplecho[30], Y_Llok[30], Y_Lzap[30], Y_Pzap[30], Y_Plok[30],
Y_Pplecho[30];
    float Dlina_plecha, Dlina_predpl, Shirina_plech;
    { // ВВОД КООРДИНАТ И НАЧАЛЬНЫХ ДАННЫХ
        // Координаты по оси OX
        ifstream XLplecho; // Считываем координату левого плеча
        XLplecho.open("Исходные координаты и данные//X. Левое плечо.txt");
        for (k=0; k<30; k++)
            XLplecho >> X_Lplecho[k];
        XLplecho.close();
        ifstream XLlok; // Считываем координату левого локтя
        XLlok.open("Исходные координаты и данные//X. Левый локоть.txt");
        for (k=0; k<30; k++)
            XLlok >> X_Llok[k];
        XLlok.close();
        ifstream XLzap; // Считываем координату левого запястья
        XLzap.open("Исходные координаты и данные//X. Левое запястье.txt");
        for (k=0; k<30; k++)
            XLzap >> X_Lzap[k];
        XLzap.close();
        ifstream XPplecho; // Считываем координату правого плеча
        XPplecho.open("Исходные координаты и данные//X. Правое плечо.txt");
        for (k=0; k<30; k++)
            XPplecho >> X_Pplecho[k];
        XPplecho.close();
        ifstream XPlok; // Считываем координату правого локтя
        XPlok.open("Исходные координаты и данные//X. Правый локоть.txt");
        for (k=0; k<30; k++)
            XPlok >> X_Plok[k];
        XPlok.close();
        ifstream XPzap; // Считываем координату правого запястья
        XPzap.open("Исходные координаты и данные//X. Правое запястье.txt");
        for (k=0; k<30; k++)
            XPzap >> X_Pzap[k];
        XPzap.close();
        // Координаты по оси OY
        ifstream YLplecho; // Считываем координату левого плеча
        YLplecho.open("Исходные координаты и данные//Y. Левое плечо.txt");
        for (k=0; k<30; k++)
            YLplecho >> Y_Lplecho[k];
        YLplecho.close();
        ifstream YLlok; // Считываем координату левого локтя
        YLlok.open("Исходные координаты и данные//Y. Левый локоть.txt");
        for (k=0; k<30; k++)
            YLlok >> Y_Llok[k];
        YLlok.close();
        ifstream YLzap; // Считываем координату левого запястья
        YLzap.open("Исходные координаты и данные//Y. Левое запястье.txt");
        for (k=0; k<30; k++)
            YLzap >> Y_Lzap[k];
        YLzap.close();
        ifstream YPplecho; // Считываем координату правого плеча
```

```

Y_Pplecho.open("Исходные координаты и данные//Y. Правое плечо.txt");
for (k=0; k<30; k++)
    Y_Pplecho >> Y_Pplecho[k];
Y_Pplecho.close();
ifstream Y_Plok; // Считываем координату правого локтя
Y_Plok.open("Исходные координаты и данные//Y. Правый локоть.txt");
for (k=0; k<30; k++)
    Y_Plok >> Y_Plok[k];
Y_Plok.close();
ifstream Y_Pzap; // Считываем координату правого запястья
Y_Pzap.open("Исходные координаты и данные//Y. Правое запястье.txt");
for (k=0; k<30; k++)
    Y_Pzap >> Y_Pzap[k];
Y_Pzap.close();
// Ввод нач. данных
ifstream D_linaplecha;
D_linaplecha.open("Исходные координаты и данные//Длина плеча.txt");
D_linaplecha >> D_lina_plecha;
D_linaplecha.close();
ifstream D_linapredpl;
D_linapredpl.open("Исходные координаты и данные//Длина предплечья.txt");
D_linapredpl >> D_lina_predpl;
D_linapredpl.close();
ifstream Shirinaplech;
Shirinaplech.open("Исходные координаты и данные//Ширина плеч.txt");
Shirinaplech >> Shirina_plech;
Shirinaplech.close();
}
float Px_Lpr[30], Px_Lp[30], Px_Pl[30], Px_Pp[30], Px_Ppr[30];
float M_Lpr[30], M_Lp[30], M_Pl[30], M_Pp[30], M_Ppr[30];
{ // ВЫЧИСЛЕНИЕ ДЛИН ОТРЕЗКОВ В ПИКСЕЛЯХ, ПЕРЕВОД ИХ В МЕТРЫ
    // Вычисляем длину в пикселях
    for (k=0; k<30; k++)
        { Px_Lpr[k] = sqrt(pow(X_Lzap[k] - X_Llok[k],2) + pow(Y_Lzap[k] -
Y_Llok[k],2));
        Px_Lp[k] = sqrt(pow(X_Llok[k] - X_Lplecho[k],2) + pow(Y_Llok[k] -
Y_Lplecho[k],2));
        Px_Pl[k] = sqrt(pow(X_Lplecho[k] - X_Pplecho[k],2) + pow(Y_Lplecho[k] -
Y_Pplecho[k],2));
        Px_Pp[k] = sqrt(pow(X_Pplecho[k] - X_Plok[k],2) + pow(Y_Pplecho[k] -
Y_Plok[k],2));
        Px_Ppr[k] = sqrt(pow(X_Plok[k] - X_Pzap[k],2) + pow(Y_Plok[k] -
Y_Pzap[k],2)); }
    // Находим множитель для перевода
    float maxSP = 0, perevod;
    for(k=0;k<30;k++)
        if(Px_Pl[k] > maxSP)
            maxSP = Px_Pl[k];
    perevod = Shirina_plech/maxSP;
    // Переводим пиксели в метры
    for (k=0; k<30; k++)
        { M_Lpr[k] = Px_Lpr[k] * perevod;
        M_Lp[k] = Px_Lp[k] * perevod;
        M_Pl[k] = Px_Pl[k] * perevod;
        M_Pp[k] = Px_Pp[k] * perevod;
        M_Ppr[k] = Px_Ppr[k] * perevod; }
}
float Tg_Lpr[30], Tg_Lp[30], Tg_Pl[30], Tg_Pp[30], Tg_Ppr[30];
{ // ВЫЧИСЛЯЕМ УГЛЫ ОТКЛОНЕНИЯ ОТ ГОРИЗОНТАЛИ
    float Cos_Lpr[30], Cos_Lp[30], Cos_Pl[30], Cos_Pp[30], Cos_Ppr[30];
    float a_Lpr[30], a_Lp[30], a_Pl[30], a_Pp[30], a_Ppr[30];
    //Вычисляем косинус
    for (k=0; k<30; k++)

```



```

    { Cos_Lpr[k] = M_Lpr[k]/Dlina_predpl;
      Cos_Lp[k] = M_Lp[k]/Dlina_plecha;
      Cos_Pl[k] = M_Pl[k]/Shirina_plech;
      Cos_Pp[k] = M_Pp[k]/Dlina_plecha;
      Cos_Ppr[k] = M_Ppr[k]/Dlina_predpl; }
for (k=0; k<30; k++)
  { a_Lpr[k] = acos(Cos_Lpr[k]);
    a_Lp[k] = acos(Cos_Lp[k]);
    a_Pl[k] = acos(Cos_Pl[k]);
    a_Pp[k] = acos(Cos_Pp[k]);
    a_Ppr[k] = acos(Cos_Ppr[k]); }
for (k=0; k<30; k++)
  { Tg_Lpr[k] = tan(a_Lpr[k]);
    Tg_Lp[k] = tan(a_Lp[k]);
    Tg_Pl[k] = tan(a_Pl[k]);
    Tg_Pp[k] = tan(a_Pp[k]);
    Tg_Ppr[k] = tan(a_Ppr[k]); }
}
{ // ВЫЧИСЛЕНИЕ КООРДИНАТ ТОЧЕК В ТРЕТЬЕЙ КООРДИНАТНОЙ ПЛОСКОСТИ
float K_Lp[30];
for (k=0; k<30; k++)
  K_Lp[k] = Tg_Lp[k] * Px_Lp[k];
ofstream lpout; // Выводим координату левого локтя
lpout.open("Результирующие координаты//Координаты левого локтя.txt");
for (k=0; k<30; k++)
  { lpout << K_Lp[k];
    lpout << endl; }
lpout.close();
float R_Lpr[30], K_Lpr[30];
for (k=0; k<30; k++)
  { R_Lpr[k] = Tg_Lpr[k] * Px_Lpr[k];
    K_Lpr[k] = K_Lp[k] - R_Lpr[k]; }
ofstream lprou; // Выводим координату левого запястья
lprou.open("Результирующие координаты//Координаты левого запястья.txt");
for (k=0; k<30; k++)
  { lprou << K_Lpr[k];
    lprou << endl; }
lprou.close();
float K_Pp[30];
for (k=0; k<30; k++)
  K_Pp[k] = Tg_Pp[k] * Px_Pp[k];
ofstream ppout; // Выводим координату правого локтя
ppout.open("Результирующие координаты//Координаты правого локтя.txt");
for (k=0; k<30; k++)
  { ppout << K_Pp[k];
    ppout << endl; }
ppout.close();
float R_Ppr[30], K_Ppr[30];
for (k=0; k<30; k++)
  { R_Ppr[k] = Tg_Ppr[k] * Px_Ppr[k];
    K_Ppr[k] = K_Pp[k] - R_Ppr[k]; }
ofstream pprou; // Выводим координату правого запястья
pprou.open("Результирующие координаты//Координаты правого запястья.txt");
for (k=0; k<30; k++)
  { pprou << K_Ppr[k];
    pprou << endl; }
pprou.close();
}
}

```

Приложение 4. Результаты работы программы расчёта координат маркеров

Таблица 4. Фрагмент итоговой сводной таблицы с координатами маркеров по всем трём осям (в пикселях) правого запястья, правого локтя и правого плеча.

1	A	B	C	D	E	F	G	H	I	J
2	Номер кадра	P3 (Правое запястье)			P2 (Правый локоть)			P1 (Правое плечо)		
3		x	y	z	x	y	z	x	y	z
4	1	376,2407407	357,5740741	30,9207	398,1153846	306,75	66,5227	400	248,1363636	
5	2	372,7037037	352,5740741	31,1194	394,962963	301,1111111	65,5458	396,3	241,3923077	
6	3	362,1296296	344,7962963	30,9867	388,1111111	294,8888889	65,0939	391,625	234,765625	
7	4	344,8684211	332,8333333	30,4202	377,1730769	287,1346154	65,0197	386,5294118	227,5588235	
8	5	322,1111111	314,8888889	28,9652	362,6034483	278,0344828	65,451	381,5769231	221,2846154	
9	6	297,9528302	286,9339623	27,0648	345,7884615	264,7884615	66,4402	376,3636364	214,6363636	
10	7	278,1153846	246,3461538	27,1473	328,9038462	244,0192308	68,911	370,9	207,25	
11	8	271,3536585	193,5731707	27,2937	315,1521739	214,173913	71,8627	365	199,5	
12	9	288,9871795	132,6025641	30,0215	313,1511628	175,2209302	73,9404	359,5	191	
13	10	336,0882353	83,64705882	31,3102	330,8529412	134,5588235	72,6584	355,6489362	178,9468085	
14	11	400,5	63,75	30,7929	365,2428571	105,2714286	67,6981	357	161,9583333	
15	12	459,8428571	80,27142857	26,1901	406,8142857	96,84285714	61,4385	366,8333333	146,754902	
16	13	497,8076923	123,7564103	21,5275	444,452381	108,5714286	56,9072	383,1428571	138,0178571	
17	14	508,6666667	175,452381	19,9819	469,7244898	136,2346939	55,683	400,7982456	139,8333333	
18	15	500,3863636	221,7045455	20,0328	481,3958333	169,125	54,7292	415,328125	146,671875	
19	16	481,5980392	256,2843137	22,2515	482,9528302	199,9339623	56,191	429,921875	156,40625	0
20	17	460,9705882	279,3823529	25,2924	481,6176471	226,5	58,5526	444,4354839	171,2419355	
21	18	440,2446809	295,2234043	29,4881	474,8529412	250,2254902	62,7528	454,0645161	191,1129032	
22	19	412,8829787	306,1170213	33,6343	459,8673469	273,0918367	65,7418	458,328125	213,59375	
23	20	379,2	307,85	35,1243	434,6086957	291,7826087	66,7598	455	237,0862069	

Приложение 5. Программа расчёта координат маркеров

```
#include <iostream>
#include <cmath>
#include <fstream>
using namespace std;
#define PI 3.10159265
int main()
{   int k;
    float X_K, X_O[10], XN_K, XN_O;
    float Y_K, Y_O[10], YN_K, YN_O;
    float Dlina;
    { // ВВОД КООРДИНАТ И НАЧАЛЬНЫХ ДАННЫХ
        // Координаты по оси OX
        ifstream XNK_in; // Считываем начальную координату крепления
        XNK_in.open("Исходные координаты и данные//Нач. координата крепления по
X.txt");
        XNK_in >> XN_K;
        XNK_in.close();
        ifstream XNO_in; // Считываем начальную координату шара
        XNO_in.open("Исходные координаты и данные//Нач. координата шара по X.txt");
        XNO_in >> XN_O;
        XNO_in.close();
        ifstream XK_in; // Считываем координату крепления
        XK_in.open("Исходные координаты и данные//Координаты крепления по X.txt");
        XK_in >> X_K;
        XK_in.close();
        ifstream XO_in; // Считываем координату шара
        XO_in.open("Исходные координаты и данные//Координаты шара по X.txt");
        for (k=0; k<10; k++)
            XO_in >> X_O[k];
        XO_in.close();
        // Координаты по оси OY
        ifstream YNK_in; // Считываем начальную координату крепления
        YNK_in.open("Исходные координаты и данные//Нач. координата крепления по
Y.txt");
        YNK_in >> YN_K;
        YNK_in.close();
        ifstream YNO_in; // Считываем начальную координату шара
        YNO_in.open("Исходные координаты и данные//Нач. координата шара по Y.txt");
        YNO_in >> YN_O;
        YNO_in.close();
        ifstream YK_in; // Считываем координату крепления
        YK_in.open("Исходные координаты и данные//Координаты крепления по Y.txt");
        YK_in >> Y_K;
        YK_in.close();
        ifstream YO_in; // Считываем координату шара
        YO_in.open("Исходные координаты и данные//Координаты шара по Y.txt");
        for (k=0; k<10; k++)
            YO_in >> Y_O[k];
        YO_in.close();
        // Ввод нач. данных
        ifstream Dlina_in;
        Dlina_in.open("Исходные координаты и данные//Длина нити.txt");
        Dlina_in >> Dlina;
```

```

    Dlina_in.close();
}
float Px[10], Pxn;
float M[10], MN;
float perevod;
{ // ВЫЧИСЛЕНИЕ ДЛИН ОТРЕЗКОВ В ПИКСЕЛЯХ, ПЕРЕВОД ИХ В МЕТРЫ, (ВЫВОДИМ ДЛИНЫ В
ПИКСЕЛЯХ В ФАЙЛЫ)
    // Вычисляем длину в пикселях
    for (k=0; k<10; k++)
        Px[k] = sqrt(pow(X_O[k] - X_K,2) + pow(Y_O[k] - Y_K,2));
    // Вычисляем начальную длину в пикселях
    Pxn = sqrt(pow(XN_O - XN_K,2) + pow(YN_O - YN_K,2));
    // Находим множитель для перевода
    perevod = Dlina/Pxn;
    // Переводим пиксели в метры
    for (k=0; k<10; k++)
        M[k] = Px[k] * perevod;
    // Выводим длины в файлы
    ofstream Pxout;
    Pxout.open("Длины в пикселях//Px.txt");
    for (k=0; k<10; k++)
        { Pxout << Px[k];
          Pxout << endl; }
    Pxout.close();
}
float Tg[10];
{ // ВЫЧИСЛЯЕМ УГЛЫ ОТКЛОНЕНИЯ ОТ ГОРИЗОНТАЛИ, (ВЫВОДИМ ТАНГЕНСЫ В ФАЙЛЫ)
float Cos[10];
float a[10];
    //Вычисляем косинус
    for (k=0; k<10; k++)
        Cos[k] = M[k]/Dlina;
    for (k=0; k<10; k++)
        a[k] = acos(Cos[k]);
    for (k=0; k<10; k++)
        Tg[k] = tan(a[k]);
    // Выводим тангенсы в файлы
    ofstream Tgout;
    Tgout.open("Тангенсы углов отклонения//Tg.txt");
    for (k=0; k<10; k++)
        { Tgout << Tg[k];
          Tgout << endl; }
    Tgout.close(); }
{ // ВЫЧИСЛЕНИЕ КООРДИНАТ ТОЧЕК В ТРЕТЬЕЙ КООРДИНАТНОЙ ПЛОСКОСТИ
float K[10], K_M[10];
    for (k=0; k<10; k++) // Вычисляем координаты в пикселях
        K[k] = Tg[k] * Px[k];
    for (k=0; k<10; k++) // Вычисляем координаты в метрах
        K_M[k] = K[k] * perevod;
    ofstream out; // Выводим координаты шара в пикселях
    out.open("Результирующие координаты//Координаты шара по Z.txt");
    for (k=0; k<10; k++)
        { out << K[k];
          out << endl; }
    out.close();
}

```

```
ofstream mout; // Выводим координаты шара в метрах
mout.open("Результирующие координаты//Координаты шара по Z в метрах.txt");
for (k=0; k<10; k++)
    { mout << K_M[k];
      mout << endl; }
mout.close();
}
}
```

Приложение 6. Таблицы с координатами маятника

Таблица 5. Фрагмент итоговой сводной таблицы с координатами маятника в пикселях и в метрах.

Номер кадра	Крепление, пикс			Шар, пикс		
	x	y	z	x	y	z
1	0	219	590	-193	370	111
2				-178	360	102
3				-152	339	88
4				-115	309	72
5				-71	269	59
6				-9	224	52
7				20	176	54
8				64	131	63
9				128	67	90
10				136	55	96
Координаты шара по системе координат, привязанной к креплению						
Номер кадра	Шар, пикс			Шар, м		
	x	y	z	x	y	z
1	-193	151	479	-0,0675	0,0527	0,1675
2	-178	141	488	-0,0620	0,0491	0,1704
3	-152	120	502	-0,0531	0,0420	0,1754
4	-115	90	518	-0,0400	0,0315	0,1808
5	-71	50	531	-0,0249	0,0175	0,1855
6	-9	5	538	-0,0030	0,0016	0,1880
7	20	-43	536	0,0070	-0,0149	0,1873
8	64	-88	527	0,0223	-0,0308	0,1840
9	128	-152	500	0,0448	-0,0531	0,1748
10	136	-164	494	0,0477	-0,0574	0,1725

Приложение 7. Таблицы с координатами маркеров

Таблица 6. Фрагмент сводной таблицы с координатами маркеров (в пикселях) плеча, локтя и запястья.

	A	B	C	D	E	F	G	H	I
1		Плечо		Локоть			Запястье		
2	Номер кадра	x	y	x	y	z	x	y	z
3	1	29	546	50	273	54	197	34	-18
4	2	29	547	49	273	49	199	35	-20
5	3	28	546	48	273	55	201	38	-19
6	4	28	547	47	274	55	208	45	-20
7	5	28	547	47	274	55	212	50	-26
8	6	28	547	46	274	55	216	54	-27
9	7	27	547	46	274	55	220	59	-32
10	8	27	547	45	274	55	224	65	-36
11	9	26	547	44	275	60	233	77	-35
12	10	26	547	43	275	60	237	83	-37
13	11	25	547	42	275	60	242	90	-39
14	12	26	547	42	276	65	246	97	-37
15	13	25	547	41	276	65	251	104	-37