

ФЕДЕРАЛЬНОЕ АГЕНСТВО ПО ОБРАЗОВАНИЮ
ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
КАФЕДРА ИНФОРМАЦИОННЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ
МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ИНДУСТРИАЛЬНЫЙ УНИВЕРСИТЕТ
(ГОУ МГИУ)

КАФЕДРА «ИНФОРМАЦИОННЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ»

ДИПЛОМНАЯ РАБОТА

по специальности «Математическое обеспечение и администрирование
информационных систем»

студента Изория Владислава Фридоновича

на тему «ТРЕХМЕРНАЯ ВИЗУАЛИЗАЦИЯ РЕЗУЛЬТАТОВ
ПОВЕРХНОСТНОГО ЭКГ КАРТИРОВАНИЯ»

Руководитель работы:
доцент, к.ф.-м.н. Белова Ирина Михайловна

Студент-дипломник _____ В.Ф. Изория

Руководитель работы,
доцент, к.ф.-м.н. _____ И.М. Белова

ДОПУСКАЕТСЯ К ЗАЩИТЕ

Зав. кафедрой 36,
доцент, к.ф.-м.н. _____ Е.А. Роганов

МОСКВА 2010

Аннотация

В дипломной работе представлено создание динамической трехмерной модели распределения потенциала на поверхности грудной клетки. Диплом содержит 45 страниц, 24 иллюстрации, 1 таблицу и 5 элементов в списке используемой литературы.

Ключевые слова: ПОВЕРХНОСТНОЕ ЭКГ КАРТИРОВАНИЕ, ТРЕХМЕРНАЯ ВИЗУАЛИЗАЦИЯ, ДВУХМЕРНАЯ ИНТЕРПОЛЯЦИЯ, OPEN GL.

Оглавление

Аннотация	2
Введение	4
Постановка задачи	7
Обзор методов визуализации	7
Трехмерные преобразования объектов.....	7
Проецирование	13
Выбор необходимых преобразований и проекций	30
Интерполяция и текстурирование	31
Интерполяция данных с датчиков	31
Генерация текстуры по полученным данным.....	35
Наложение текстуры на модель	39
Программная реализация и обсуждение результатов	40
Выводы	44
Литература и гиперссылки.....	45

Введение

В настоящее время в кардиологии стало активно использоваться поверхностное ЭКГ-картирование (одномоментная регистрация множественных отведений ЭКГ со всей поверхности грудной клетки). Оно является одним из наиболее информативных методов исследования электрической активности миокарда. Метод позволяет получить максимальную информацию об особенностях электрического поля сердца в любой момент деполяризации и реполяризации желудочков. Автоматическая регистрация распределения электрических потенциалов на поверхности тела осуществляется с помощью сложных компьютерных систем. Униполярные электроды располагаются на передней, задней и боковых поверхностях грудной клетки. Получаемая таким образом, пространственно-временная и амплитудно-временная информация может быть представлена в виде нескольких разновидностей картограмм. В данной работе будут изучаться изоинтегральные картограммы распределения сумм площадей под кривой ЭКГ за любой интересующий период сердечного цикла.

Примеры изопотенциальных карт приведены на рис. 1. Схематическая развертка грудной клетки представлена как цилиндрическая поверхность, разомкнутая по правой задне-подмышечной линии. Для построения карты (в указанный момент времени сердечного цикла) точки, имеющие равный потенциал, соединяются линиями. Точки, имеющие положительный потенциал, создают позитивную зону (незакрашенная область) с максимумом потенциала («+», числовое значение приведено в мкВ), а точки, имеющие отрицательный потенциал, создают негативную зону (закрашенная область) с минимумом потенциала («-», числовое значение приведено в мкВ).

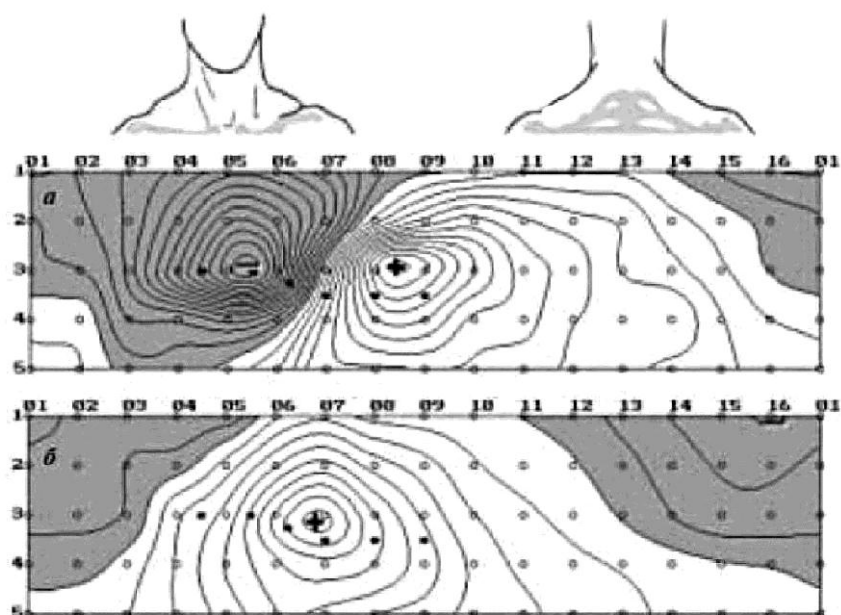


Рис. 1. Типичное распределение потенциалов на изопотенциальных картах процесса деполаризации и реполаризации желудочков

По характеристикам поверхностного ЭКГ-картирования можно оценить следующие параметры: характер распределения потенциалов (дипольность-мультдипольность), расположение экстремумов потенциалов, величина экстремумов, наличие нетипичных положительных и отрицательных зон, направление и скорость движения фронтов возбуждения.

На рис. 1 мы видим изопотенциальную карту на плоскости, однако в действительности она представляет собой трехмерную модель. То есть мы сталкиваемся с ситуацией, когда выбранный метод визуализации убирает одно из измерений. В действительности изопотенциальная карта выглядит так, как показано на рис.2.

Имея такую модель, медицинские работки смогут более наглядно получать данные и сопоставлять их с реальными физическими процессами. Для обеспечения их необходимой поддержкой будет создан интегрированный аппаратно-программный комплекс.

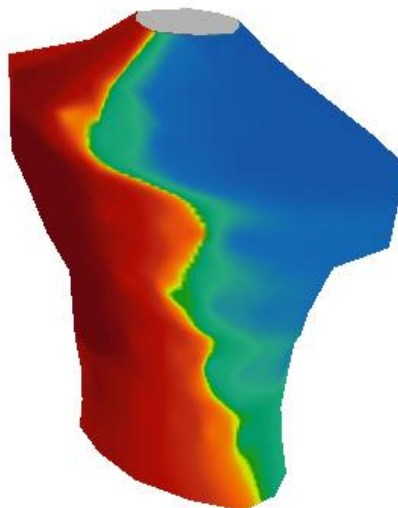


Рис. 2. Типичное распределение потенциалов на трехмерной модели человеческого торса

Аппаратно-программный комплекс для проведения медицинской диагностики должен обладать следующим функционалом:

1. Получение данных с тела пациента;
2. Первичная обработка и сохранение этих данных для последующего использования;
3. Автоматизированный анализ полученных данных на основе ранее полученных данных;
4. Визуализация полученных данных с целью ручного анализа.

На данный момент первые два пункта реализованы в программно аппаратном комплексе Carddiag. Пункт номер 3 был реализован в ходе исследований методов анализа поверхностного ЭКГ картирования в Московском государственном индустриальном университете.

Предлагаемые на данный момент методы визуализации являются ненаглядными и, поэтому, недостаточными для ручного анализа. Метод визуализации будет являться достаточным в том случае, когда отображаемые данные могут быть с легкостью интерпретированы, а так же легко сопоставлены с реальными физическими процессами.

Постановка задачи

На данный момент отсутствует программное обеспечение для трехмерной визуализации. Все решения на данный момент представляют собой двумерную визуализацию. Целью же данной работы будет создание полноценного программного обеспечения для трехмерной визуализации поверхностного ЭКГ картирования.

Обзор методов визуализации

Вначале рассмотрим различные инструменты для трехмерной визуализации. К таким инструментам относятся трехмерное проецирование и преобразования трехмерных объектов.

Трехмерные преобразования объектов

После построения модели нам необходимо произвести преобразования над нашей трехмерной моделью, таким образом, чтобы её вид соответствовал желаемому. Суть проблемы заключается в том, что одну и ту же модель пользователь системы может захотеть рассмотреть модель под разными углами, а так же с разным приближением.

Для решения данной задачи будем использовать механизм трехмерных преобразований, основанный на использовании матрицы трансформации^[4].

Рассмотрим для начала модель, состоящую из одной точки в трехмерном пространстве. Эта точка будет определена 3 координатами:

$$a = \begin{pmatrix} x_a \\ y_a \\ z_a \end{pmatrix}$$

Каждая из координат определяет положение точки вдоль осей координат. Определим так же вектор и опорную точку. Назовем данную пару элементов ось вращения:

$$c = \begin{pmatrix} x_c \\ y_c \\ z_c \end{pmatrix}; d = \begin{pmatrix} x_d \\ y_d \\ z_d \end{pmatrix};$$
$$\vec{M} = \vec{cd}, c - \text{опорная точка}$$

Рассмотрим задачу вращения точки a вокруг оси \vec{M} . Для этого вначале рассмотрим задачу вращения в двухмерном пространстве. Простейшим

случаем будет являться вращение точки на угол α относительно центра координат. При этом мы получим точку, координаты которой могут быть получены по формуле:

$$a' = \begin{pmatrix} x_a * \cos \alpha - y_a * \sin \alpha \\ x_a * \sin \alpha + y_a * \cos \alpha \end{pmatrix}$$

Подобное выражение удобнее записывать в матричной форме, то есть:

$$a' = \begin{pmatrix} x_{a'} \\ y_{a'} \end{pmatrix} = \begin{pmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{pmatrix} \cdot \begin{pmatrix} x_a \\ y_a \end{pmatrix}$$

Однако если мы хотим произвести преобразование относительно точки, отличной от начала координат, то нас необходимо будет проделать 3 операции:

1. Переместить точки a и a' таким образом, чтобы точка a' совпала с началом координат;
2. Произвести вращение вокруг начала координат;
3. Сделать преобразование, обратное к первому.

Таким образом, нам необходимо определить операцию переноса. Для этого введем понятие матрицы преобразование в однородных координатах:

$$(T) = \begin{pmatrix} a & b & m \\ c & d & n \\ p & q & s \end{pmatrix}, \text{ где}$$

a, b, c, d – коэффициенты исходной матрицы преобразования;

m, n – преобразование переноса;

p, q – коэффициенты проецирования (нами использоваться не будут);

s – коэффициент глобального масштабирования.

Мы можем записать преобразование переноса в начало координат следующим образом:

$$a^0 = \begin{pmatrix} 1 & 0 & -x_c \\ 0 & 1 & -y_c \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x_a \\ y_a \\ 1 \end{pmatrix}$$

Аналогичным образом определим перенос в исходную точку:

$$a' = \begin{pmatrix} 1 & 0 & x_c \\ 0 & 1 & y_c \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x_{a^{0'}} \\ y_{a^{0'}} \\ 1 \end{pmatrix}$$

Таким образом, поворот точки a вокруг c точки в двумерном пространстве мы можем записать в виде:

$$a' = \begin{pmatrix} 1 & 0 & x_c \\ 0 & 1 & y_c \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & -x_c \\ 0 & 1 & -y_c \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x_a \\ y_a \\ 1 \end{pmatrix}$$

Обобщим данную задачу для многоугольников. Для того, чтобы повернуть многоугольник вокруг произвольной точки необходимо каждую его вершину подвергнуть трансформации, описанной выше.

Следующим обобщением будет переход в трехмерное пространство. В трехмерном пространстве мы не можем вращать однозначно вокруг точки, зная только один угол, поэтому в трехмерном пространстве мы будем вращать точку (а вследствие обобщения – всю фигуру целиком).

Рассмотрим в начале поворот точки (точнее её радиус-вектора) вокруг оси, проходящей через начало координат. Данное преобразование показано на рисунке 3.

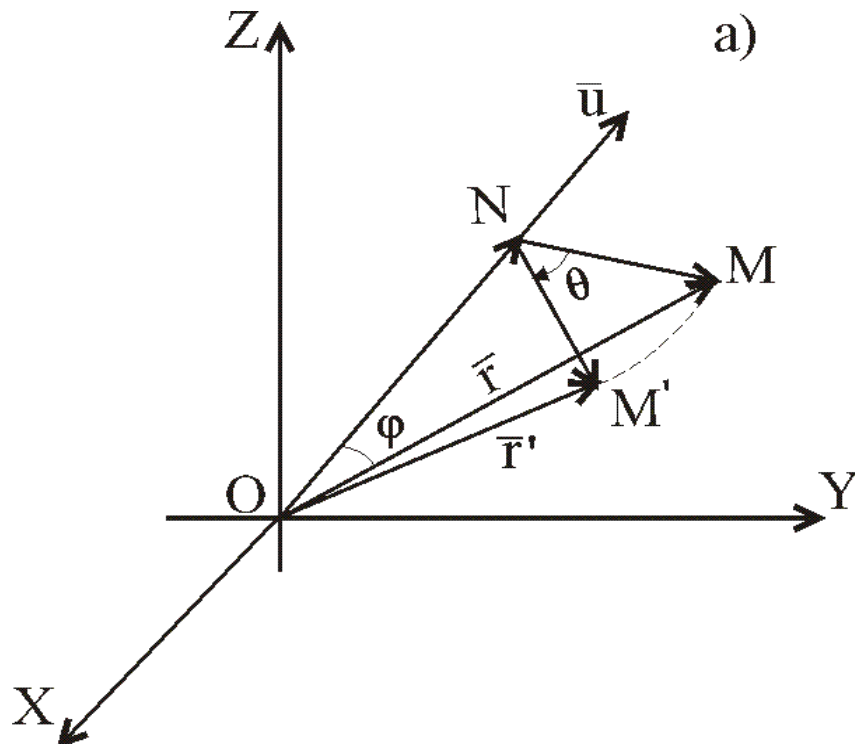


Рис. 3. Поворот точки относительно оси в трехмерном пространстве

Мы будем вращать вектор \vec{r} относительно оси ON, направляющим вектором которой является единичный вектор \vec{u} . Для начала введем матрицу преобразования в однородных координатах для трехмерного пространства:

$$(T) = \begin{pmatrix} a & b & c & m \\ d & e & f & n \\ g & i & j & l \\ p & q & r & s \end{pmatrix}$$

Преобразования переноса и масштабирования будут выполняться в точности по аналогии с двухмерным случаем.

Рассмотрим операцию поворота.

Если вектор \vec{r} повернут вокруг оси на угол θ , то вектор \vec{ON} остается без изменений, а вектор \vec{r} занимает положение $\vec{NM'}$. Выразим векторы \vec{ON} , \vec{NM} и \vec{NP} через известные векторы \vec{r} , $\vec{r'}$, \vec{u} .

Из рисунка 3 следует, что $\vec{ON} = |\vec{ON}| \cdot \vec{u}$. Модуль вектора $|\vec{ON}| = |\vec{r}| \cos \varphi = |\vec{r}| |\vec{u}| \cos \varphi = (\vec{r} \cdot \vec{u})$ – скалярное произведение векторов. Следовательно, $\vec{ON} = (\vec{r} \cdot \vec{u}) \cdot \vec{u}$. Поскольку $\vec{ON} + \vec{NM} = \vec{OM}$, то $\vec{NM} = \vec{OM} - \vec{ON} = \vec{r} - (\vec{r} \cdot \vec{u}) \cdot \vec{u}$.

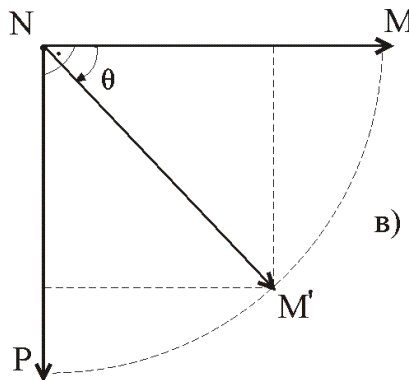


Рис. 4. Поворот точки в плоскости MNP

Из рисунка следует, что $|\vec{NM}| = |\vec{NM'}| = |\vec{NP}|$. Вектор \vec{NP} перпендикулярен плоскости векторов \vec{r} и \vec{u} , следовательно $|\vec{NP}| = |\vec{u} \times \vec{r}|$ – векторное произведение векторов. При этом

$$|\vec{u} \times \vec{r}| = |\vec{r}| |\vec{u}| \sin \varphi = |\vec{r}| \sin \varphi = |\vec{NP}|.$$

Следовательно,

$$|\overline{NM}| = |\overline{NM'}| = |\overline{NP}| = |\overline{u} \times \overline{r}| = |\overline{r} - (\overline{r} \cdot \overline{u}) \cdot \overline{u}|.$$

Разложив вектор $|\overline{NM'}|$ на векторы $|\overline{NM}|$ и $|\overline{NP}|$, получим

$$\overline{NM'} = \cos \Theta (\overline{r} - (\overline{r} \cdot \overline{u}) \cdot \overline{u}) + \sin \Theta (\overline{u} \times \overline{r}).$$

Обозначая полученный в результате вращения вектор $\overline{OM'}$ через $\overline{r'}$, получим $\overline{r'} = \overline{OM'} = \overline{ON} + \overline{NM'} = (\overline{r} \cdot \overline{u}) \cdot \overline{u} + \cos \Theta (\overline{r} - (\overline{r} \cdot \overline{u}) \cdot \overline{u}) + \sin \Theta (\overline{u} \times \overline{r})$.

Заметим, что если векторы обозначены через матрицы столбцы

$$\overline{a} = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix}, \overline{b} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix},$$

то $a^T = (a_1, a_2, a_3)$ - матрицы строка. Согласно определению произведения матриц, скалярное произведение $(\overline{a} \cdot \overline{b}) = a_1 b_1 + a_2 b_2 + a_3 b_3$ может быть записано в виде $(\overline{a} \cdot \overline{b}) = (a_1, a_2, a_3) \cdot \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}$, следовательно произведение $a^T \cdot \overline{b} = (\overline{a} \cdot \overline{b})$. Учитывая это замечание и свойства смешанного произведения, можно доказать, что:

$$(a^T) \cdot (\overline{b}) \cdot \overline{c} = (\overline{c}) \cdot (a^T) \cdot \overline{b}.$$

С учетом сделанного замечания, мы можем произвести следующие преобразования:

$$(\overline{r} \cdot \overline{u}) \cdot \overline{u} = (\overline{u}) \cdot (u^T) \cdot \overline{r}.$$

Векторное произведение $(\overline{u} \times \overline{r})$ так же целесообразно заменить соответствующим матричным произведением.

$$\overline{u} \times \overline{r} = (V) \cdot \overline{r} = \begin{pmatrix} 0 & -u_3 & u_2 \\ u_3 & 0 & -u_1 \\ -u_2 & u_1 & 0 \end{pmatrix} \cdot \overline{r}$$

Используя полученные выражения можно записать окончательное выражение для $\overline{r'}$:

$$\bar{r}' = (\bar{u}) \cdot (u^T) \cdot \bar{r} + \cos \Theta (\bar{u}) \cdot (u^T) \cdot \bar{r} + \sin \Theta (V) \cdot \bar{r}, \text{ или}$$

$$\bar{r}' = (T) \cdot \bar{r}, \text{ где}$$

$$(T) = (\bar{u}) \cdot (u^T) + \cos \Theta (\bar{u}) \cdot (u^T) + \sin \Theta (V).$$

В развернутом виде матрица преобразования поворота вектора \bar{r} на угол Θ вокруг произвольной оси, определяемой направлением вектора \bar{u} выглядит следующим образом:

$$(T) = \begin{pmatrix} u_1^2 + \cos \Theta (1 - u_1^2) & u_1 u_2 (1 - \cos \Theta) - u_3 \sin \Theta & u_1 u_3 (1 - \cos \Theta) + u_2 \sin \Theta & 0 \\ u_2 u_1 (1 - \cos \Theta) + u_3 \sin \Theta & u_2^2 + \cos \Theta (1 - u_2^2) & u_2 u_3 (1 - \cos \Theta) - u_1 \sin \Theta & 0 \\ u_3 u_1 (1 - \cos \Theta) - u_2 \sin \Theta & u_3 u_2 (1 - \cos \Theta) + u_1 \sin \Theta & u_3^2 + \cos \Theta (1 - u_3^2) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Далее рассмотрим построение композиций трехмерных преобразований. Путем объединения трехмерных преобразований, рассмотренных выше, можно получить другие различные сложные преобразования объектов. Так как в общем случае перемножение матриц является некоммутативной операцией, то важен порядок этого перемножения. Правильный порядок определяется положением конкретной матрицы преобразования относительно матрицы координатного вектора. С учетом того, что матрица координатного вектора записывается в виде матрицы-столбца, эта матрица в любой композиции преобразований будет последним сомножителем. Матрица, ближайшая к матрице координатного вектора, является матрицей первого преобразования (T_1). Матрица, записанная перед ней, является матрицей второго преобразования (T_2) и тд.

$$(X') = (T) \cdot (X) = (T_n) \cdot (T_{n-1}) \cdots (T_i) \cdots (T_2) \cdot (T_1) \cdot (X)$$

где (T_i) - произвольная комбинация матриц масштабирования, сдвига, вращения, отражения, переноса, перспективного преобразования и проецирования.

С учетом того, что перспективные преобразования искажают геометрические объекты, а преобразования проецирования приводят к потере информации, эти матрицы должны быть расположены первой и второй (по порядку записи формулы).

Используя композицию преобразований определим трехмерные вращения вокруг произвольной оси. Если ось, вокруг которой осуществляется вращение объекта, не проходит через начало координат, а проходит,

например, через точку $P(m, n, l)$, то композиция трехмерных преобразований должна включать следующие три матрицы:

$$(T_1) = \begin{pmatrix} 1 & 0 & 0 & -m \\ 0 & 1 & 0 & -n \\ 0 & 0 & 1 & -l \\ 0 & 0 & 0 & 1 \end{pmatrix} - \begin{array}{l} \text{матрица линейного переноса} \\ \text{оси вращения из точки } P(m, n, l) \\ \text{в начало координат} \end{array}$$

$$(T_2) = (T) - \begin{array}{l} \text{матрица вращения объекта} \\ \text{вокруг оси, проходящей через} \\ \text{начало координат} \end{array}$$

$$(T_3) = \begin{pmatrix} 1 & 0 & 0 & m \\ 0 & 1 & 0 & n \\ 0 & 0 & 1 & l \\ 0 & 0 & 0 & 1 \end{pmatrix} - \begin{array}{l} \text{матрица обратного переноса} \\ \text{оси вращения из начала координат} \\ \text{в точку } P(m, n, l) \end{array}$$

В итоге трехмерное вращение объекта вокруг произвольной оси можно осуществить с использованием следующих преобразований:

$$(X') = (T) \cdot (X) = (T_3) \cdot (T_2) \cdot (T_1) \cdot (X)$$

где (X) - матрица координат исходного объекта, а (X') - матрица координат преобразованного объекта.

Проецирование

Важнейшим критерием при построении трехмерной модели является выбор подходящей проекции. Именно проекция определяет, каким образом трехмерная фигура отображается на нашем мониторе. Неверный выбор проекции может свести на нет все усилия по построению трехмерной модели, так как она может выглядеть не объемной, а плоской.

В плоской проекции каждая точка объекта проецируется определенным образом на так называемую плоскость проекции, а её образ называется точкой проекции. Плоские геометрические проекции объектов образуются пересечением прямых, называемых линиями проекции или проекторами с плоскостью проекции. Проекторы – это прямые, проходящие через некоторую произвольную точку, называемую центром проекции, и каждую точку объекта. Если центр проекции расположен в конечной точке трехмерного пространства, то получается перспективная или центральная проекция. Если же центр расположен в бесконечности, то все линии

проекции параллельны между собой, и в результате получается параллельная проекция.

Существует большое количество различных проекций трехмерных объектов на плоскость^[3]. Формально их можно выстроить в иерархию, показанную на рисунке 3.



Рассмотрим в начале параллельные проекции. В плоской параллельной проекции все точки объекта проецируются параллельно некоторому фиксированному направлению \vec{u} (единичный вектор). Данная проекция изображена на рисунке 4.

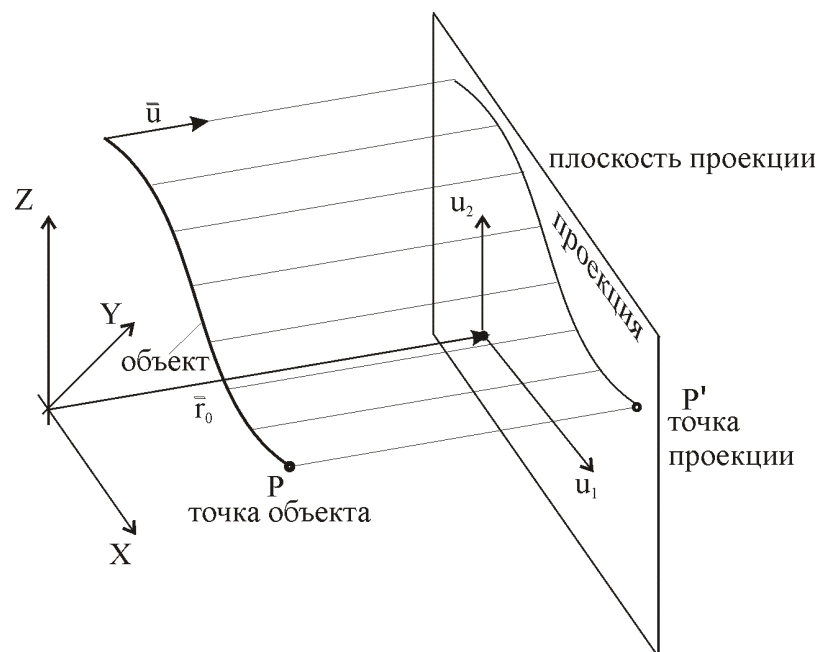


Рис. 5. Параллельная проекция

Рассмотрим некоторую точку объекта P , определенную соответствующим радиус-вектором \bar{r} в пространственной системе координат объекта. Зададим фиксированное направление линии проекции \bar{u}_3 из этой точки на плоскость проекции. Направление осей координат на плоскости проекции задаются направлением единичных векторов \bar{u}_1 и \bar{u}_2 соответственно для осей X' и Y' .

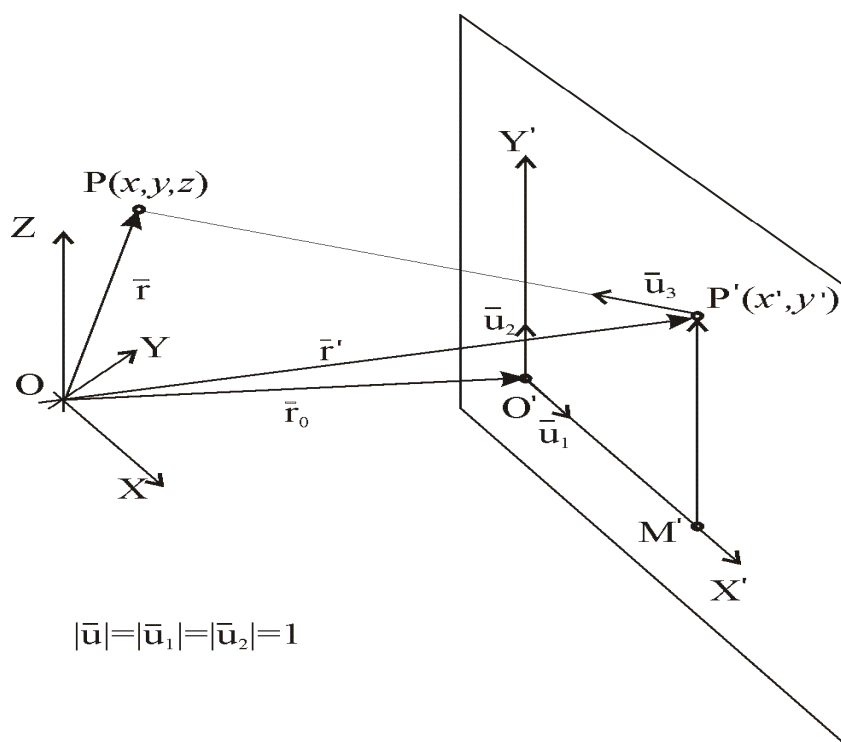


Рис. 6. Построение параллельной проекции

Если длина отрезка PP' равна z' , а направление единичного вектора $\overline{u_3}$ определено так, как показано на рисунке 5, то вектор $\overline{PP'} = -z'\overline{u_3}$, следовательно, с одной стороны: $\overline{OP'} = \overline{OP} + \overline{PP'}$ или $\overline{r'} = \overline{r} - z' \cdot \overline{u_3}$.

Выразив $\overline{r'}$ так же через $\overline{u_1}$ и $\overline{u_2}$ мы получим, что

$$\begin{aligned}\overline{r'} &= \overline{r} - z' \cdot \overline{u_3} = \overline{r_0} + x' \cdot \overline{u_1} + y' \cdot \overline{u_2}, \text{ или} \\ \overline{r} - \overline{r_0} &= x' \cdot \overline{u_1} + y' \cdot \overline{u_2} + z' \cdot \overline{u_3}.\end{aligned}$$

Умножим левую и правую часть полученного уравнения скалярно на векторное произведение $[\overline{u_1} \times \overline{u_2}]$. Поскольку в результате векторного произведения получается вектор, перпендикулярный плоскости двух векторов-сомножителей, то по свойству скалярных произведений взаимноперпендикулярных векторов получим:

$$(\overline{u_1} \cdot [\overline{u_1} \times \overline{u_2}]) = (\overline{u_2} \cdot [\overline{u_1} \times \overline{u_2}]) = 0.$$

Разрешив полученное после умножения уравнение относительно z' , будем иметь:

$$z' = \frac{((\overline{r} - \overline{r_0}) \cdot [\overline{u_1} \times \overline{u_2}])}{(\overline{u_3} \cdot [\overline{u_1} \times \overline{u_2}])}$$

Аналогично, умножая уравнение на $[\overline{u_2} \times \overline{u_3}]$ и $[\overline{u_1} \times \overline{u_3}]$ мы получим

$$\begin{aligned}x' &= \frac{((\overline{r} - \overline{r_0}) \cdot [\overline{u_2} \times \overline{u_3}])}{(\overline{u_1} \cdot [\overline{u_2} \times \overline{u_3}])} = \frac{((\overline{r} - \overline{r_0}) \cdot [\overline{u_2} \times \overline{u_3}])}{(\overline{u_3} \cdot [\overline{u_1} \times \overline{u_2}])} \\ y' &= \frac{((\overline{r} - \overline{r_0}) \cdot [\overline{u_1} \times \overline{u_3}])}{(\overline{u_2} \cdot [\overline{u_1} \times \overline{u_3}])} = \frac{((\overline{r} - \overline{r_0}) \cdot [\overline{u_1} \times \overline{u_3}])}{(\overline{u_3} \cdot [\overline{u_1} \times \overline{u_2}])}\end{aligned}$$

Перестановка векторов-сомножителей в смешанных произведениях знаменателей полученных уравнений выполнены, исходя из свойств смешанного произведения, с целью унификации этих произведений. Эти формулы определяют координаты точки проекции P' на плоскости проекции.

Если линии проекции перпендикулярны плоскости проекции, то

$$\overline{u_3} = [\overline{u_1} \times \overline{u_2}], \quad \overline{u_1} = [\overline{u_2} \times \overline{u_3}], \quad \overline{u_2} = [\overline{u_3} \times \overline{u_1}]$$

В этом случае полученные нами формулы для x' , y' и z' существенно изменятся:

$$x' = ((\bar{r} - \bar{r}_0) \cdot \bar{u}_1) = (x - x_0)u_{1x} + (y - y_0)u_{1y} + (z - z_0)u_{1z}$$

$$y' = ((\bar{r} - \bar{r}_0) \cdot \bar{u}_2) = (x - x_0)u_{2x} + (y - y_0)u_{2y} + (z - z_0)u_{2z}$$

$$z' = ((\bar{r} - \bar{r}_0) \cdot \bar{u}_3) = (x - x_0)u_{3x} + (y - y_0)u_{3y} + (z - z_0)u_{3z}$$

Стоит отметить, что для получения проекции заданного объекта информация относительно z' не нужна, но, располагая этой информацией, при необходимости можно восстановить любую точку объекта.

Полученный набор уравнений можно записать используя матричную форму и однородные координаты.

$$(R') = \begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = (A) \cdot (T) \cdot (R) =$$

$$= \begin{pmatrix} u_{1x} & u_{1y} & u_{1z} & 0 \\ u_{2x} & u_{2y} & u_{2z} & 0 \\ u_{3x} & u_{3y} & u_{3z} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 & -x_0 \\ 0 & 1 & 0 & -y_0 \\ 0 & 0 & 1 & -z_0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

Таким образом, процесс параллельного проецирования эквивалентен переносу плоскости проекции и объекта до тех пор, пока плоскость проекции не пройдет через начало координат (матрица Т), с последующим их вращением (матрица А) до тех пор, пока оси координат плоскости проекции не совпадут с осями ОХ и ОУ системы координат объекта. Чтобы восстановить точку объекта Р необходимо выполнить обратные преобразования.

Рассмотрим различные типы плоских параллельных проекций. Начнем с ортогональной, или, как её ещё называют, ортографической проекции. При построении этой проекции линии проекции перпендикулярны плоскости проекции. В этом случае точно отображаются размер и форма одной из плоских граней объекта на плоскость проекции, параллельную этой грани. Как правило, ортографические проекции – это параллельные проекции на одну из координатных плоскостей (X=0; Y=0; Z=0). Например, матрица преобразования для получения проекции объекта на плоскость Z=0 имеет вид:

$$(T) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

В результате подобного преобразования z-координата координатного вектора станет равной нулю. Аналогичным образом можно записать матрицы преобразования для получения проекций на другие координатные плоскости.

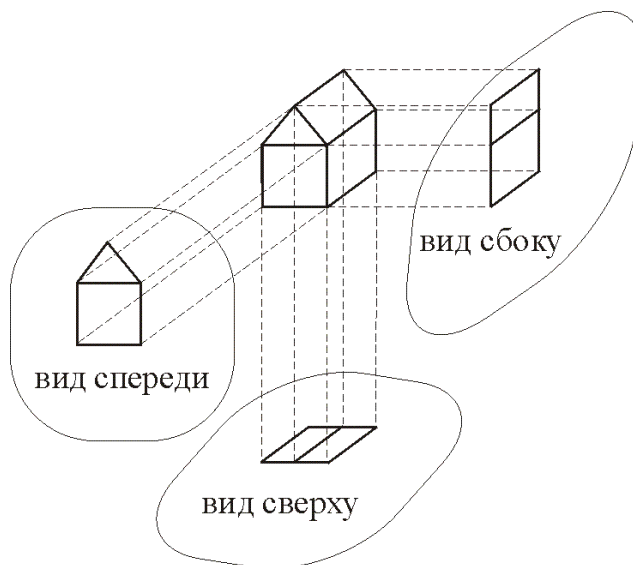


Рис. 7. Проекции на координатные плоскости.

Наиболее широко используемыми ортографическими проекциями являются вид спереди (фронтальная проекция), вид сверху (план) и вид сбоку (боковая проекция), для каждой из которых плоскость проекции расположена параллельно соответствующей главной грани объекта, как показано на рисунке 6. В инженерной графике для описания машиностроительных деталей, агрегатов, архитектурных сооружений и т. п. плоскости проекций, как правило, перпендикулярны главным координатным осям. По этим проекциям можно легко измерять углы и расстояния соответствующих элементов объекта.

Далее рассмотрим класс аксонометрических проекций. В аксонометрических проекциях плоскость проекции может иметь любую ориентацию относительно объекта. Если плоскость проекции ориентирована симметрично по отношению к трем главным плоским пересекающимся граням объекта, то образуется изометрическая проекция (изо – равно, метрия – измерение). Если плоскость проекции ориентирована симметрично

по отношению к двум главным граням объекта, то образуется диметрическая проекция. Самый общий случай аксонометрической проекции – триметрическая проекция. Поскольку в случае аксонометрических проекций грань объекта не параллельна плоскости проекции, то аксонометрические проекции не показывают истинную форму объекта, так как искажаются углы, но при этом проецировании сохраняется параллельность прямых, которые одинаково искажаются (укорачиваются). Коэффициент искажения равен отношению длины проекции отрезка к его истинной длине.

Триметрическая проекция строится произвольными поворотами объекта вокруг любых координатных осей, совершаемыми в произвольном порядке, с последующим проецированием на плоскость $Z=0$. Как правило, триметрические проекции строятся с помощью поворота вокруг оси Y , затем вокруг оси X и параллельного проецирования на плоскость $Z=0$. На рисунке показаны различные проекции одного и того же объекта с различными углами поворота вокруг осей.

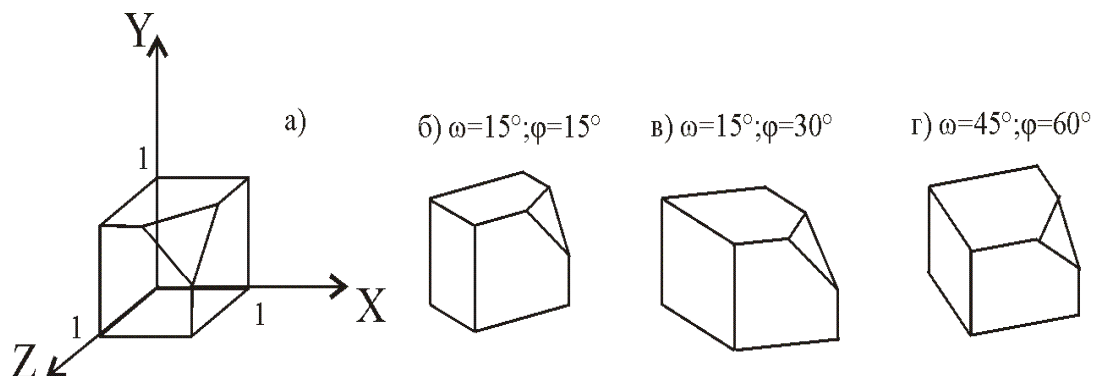


Рис. 8. Триметрические проекции

Для любой конкретной триметрической проекции коэффициенты искажения вычисляются с помощью применения общей матрицы преобразования поворота к единичным векторам вдоль главных координатных осей.

$$(R^*) = (T) \cdot (R) = \begin{pmatrix} a & b & c & 0 \\ d & e & f & 0 \\ g & i & j & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix} = \begin{pmatrix} a & b & c \\ d & e & f \\ g & i & j \\ 1 & 1 & 1 \end{pmatrix}$$

После проецирования на плоскость $Z=0$ получим:

$$(R') = (T) \cdot (R^*) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} a & b & c \\ d & e & f \\ g & i & j \\ 1 & 1 & 1 \end{pmatrix} = \begin{pmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$$

Квадрат длины подвергнутого преобразованию единичного вектора вдоль соответствующей оси, то есть квадрат коэффициента искажения будет равен:

$$d_x^2 = a^2 + d^2$$

$$d_y^2 = b^2 + e^2$$

$$d_z^2 = c^2 + f^2$$

Диметрическая проекция – это триметрическая проекция с двумя одинаковыми коэффициентами искажения, т.е. две из трех осей во время проецирования одинаково уменьшаются. Чтобы получить необходимые соотношения для диметрических проекций, рассмотрим вращение объекта первоначально относительно оси Y на угол ω , а затем вокруг оси X на угол φ :

$$(R') = \begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = (T_x) \cdot (T_y) \cdot (R) =$$

$$= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \varphi & -\sin \varphi & 0 \\ 0 & \sin \varphi & \cos \varphi & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} \cos \omega & 0 & \sin \omega & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \omega & 0 & \cos \omega & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

Или после перемножения матриц (T_x) и (T_y) :

$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} \cos \omega & 0 & \sin \omega & 0 \\ \sin \omega \sin \varphi & \cos \varphi & -\cos \omega \sin \varphi & 0 \\ -\sin \omega \cos \varphi & \sin \varphi & \cos \omega \cos \varphi & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

При помощи полученной матрицы преобразования единичные векторы $\bar{i}, \bar{j}, \bar{k}$ по координатным осям X, Y и Z преобразуются к виду

$$(R^*) = \begin{pmatrix} \cos \omega & 0 & \sin \omega & 0 \\ \sin \omega \sin \varphi & \cos \varphi & -\cos \omega \sin \varphi & 0 \\ -\sin \omega \cos \varphi & \sin \varphi & \cos \omega \cos \varphi & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix} =$$

$$= \begin{pmatrix} \cos \omega & 0 & \sin \omega \\ \sin \omega \sin \varphi & \cos \varphi & -\cos \omega \sin \varphi \\ -\sin \omega \cos \varphi & \sin \varphi & \cos \omega \cos \varphi \\ 1 & 1 & 1 \end{pmatrix}$$

Квадраты длин преобразованных единичных векторов $\bar{i}, \bar{j}, \bar{k}$ (коэффициенты искажения) вдоль соответствующих осей координат будут равны (при проецировании на плоскость $Z=0$):

$$d_x^2 = \cos^2 \omega + (\sin \omega \sin \varphi)^2$$

$$d_y^2 = \cos^2 \varphi$$

$$d_z^2 = \sin^2 \omega + (\cos \omega \sin \varphi)^2$$

Приравнявая, например, коэффициенты искажения по осям x и y (могут быть выбраны любые две оси из трех), получим:

$$\cos^2 \omega + (\sin \omega \sin \varphi)^2 = \cos^2 \varphi, \text{ или}$$

$$\sin^2 \omega = \frac{\sin^2 \varphi}{1 - \sin^2 \varphi}$$

Фиксируя коэффициент искажения d_z вдоль третьей оси z и объединяя полученные уравнения получим:

$$d_z^2 = \sin^2 \omega + (\cos \omega \sin \varphi)^2 = \frac{\sin^2 \varphi}{1 - \sin^2 \varphi} + \left(1 - \frac{\sin^2 \varphi}{1 - \sin^2 \varphi}\right) \sin^2 \varphi$$

Разрешая полученное уравнение относительно $\sin \varphi$, запишем биквадратное уравнение в виде:

$$2 \sin^4 \varphi - (2 + d_z^2) \sin^2 \varphi + d_z^2 = 0$$

Решив это уравнение относительно $u = \sin^2 \varphi$, найдем, что $(\sin^2 \varphi)_1 = 1$ (решение не имеет смысла, так как обращает знаменатель в 0) и $(\sin^2 \varphi)_2 = \frac{d_z^2}{2}$. Следовательно, $\sin \varphi = \pm \frac{d_z}{\sqrt{2}}$ или (получим так же уравнение для ω):

$$\varphi = \arcsin\left(\pm \frac{d_z}{\sqrt{2}}\right)$$

$$\omega = \arcsin\left(\pm \frac{d_z}{\sqrt{2 - d_z^2}}\right)$$

На основании полученных формул можно сделать вывод, что диапазон коэффициента искажения равен $0 \leq d \leq 1$, а так же, что каждый коэффициент искажения d порождает четыре возможные диметрические проекции.

Диметрические проекции для различных значений коэффициента искажения d_z (положительные вращения вокруг осей) для объекта, изображенного на рисунке:

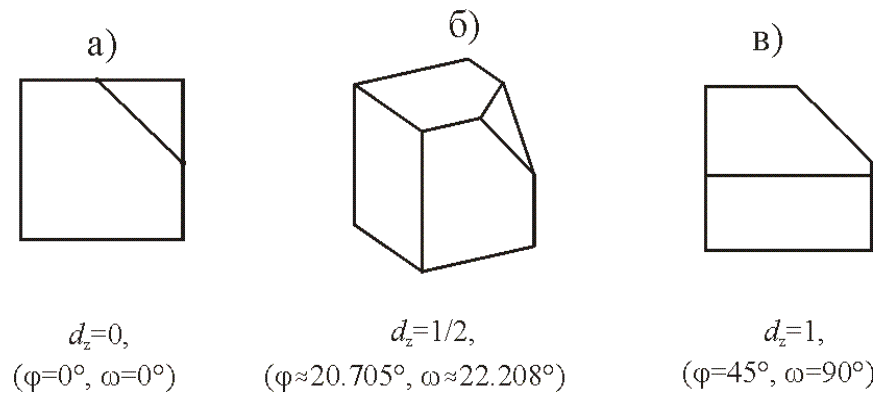


Рис. 9. Диметрические проекции

Таким образом, общая матрица преобразования для получения соответствующей проекции в случае $d_z = \frac{1}{2}$ запишется в виде ($Z = 0$):

$$(T) = \begin{pmatrix} 0.926 & 0 & 0.378 & 0 \\ 0.134 & 0.935 & -0.327 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Рассмотрим далее изометрические проекции. В изометрической проекции все три коэффициента искажения равны. Приравняв полученные ранее уравнения для коэффициентов искажения получим:

$$\sin^2 \omega = \frac{\sin^2 \varphi}{1 - \sin^2 \varphi}$$

$$\sin^2 \omega = \frac{1 - 2 \sin^2 \varphi}{1 - \sin^2 \varphi}$$

Затем найдем, что $\sin^2 \varphi = \frac{1}{3}$, или $\sin \varphi = \sqrt{\frac{1}{3}}$, а $\varphi \approx \pm 35.26^\circ$. И аналогично, что $\sin^2 \omega = \frac{1}{2}$ и $\omega = \pm 45^\circ$.

Таким образом установлено, что может существовать четыре изометрические проекции объекта ($\varphi < 0, \omega > 0$; $\varphi < 0, \omega < 0$; $\varphi > 0, \omega > 0$; $\varphi > 0, \omega < 0$). Коэффициент искажения для изометрической проекции равен:

$$d = d_x = d_y = d_z = \sqrt{\cos^2 \varphi} = \sqrt{\frac{2}{3}} \approx 0.8165$$

Следовательно, изометрическая проекция есть частный случай диметрической проекции с коэффициентом искажения $d_z \approx 0.8165$.

На рисунке показана одна из возможных изометрических проекций объекта для случая ($\varphi < 0, \omega < 0$), а так же изометрическая проекция единичных векторов, определяющих направление осей в пространственной декартовой системе координат. Следует заметить, что окружность в аксонометрической проекции изображается в эллипс.

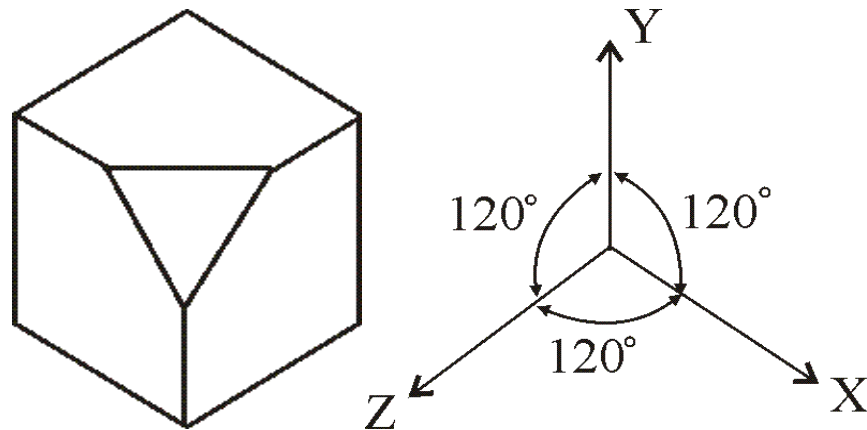


Рис. 10. Изометрическая проекция

Косоугольная проекция является параллельной проекцией общего вида. При построении косоугольной проекции не накладываются никакие ограничения на угол между линиями проекции и плоскостью проекции. В косоугольной проекции сохраняются углы между прямыми на объекте, расположенными в плоскости, параллельной плоскости проекций. Окружность в такой плоскости проецируется в окружность и при этом на

изображении присутствует более чем одна грань объекта. На рис. показан процесс построения косоугольной проекции. Видно, что нормаль к проекционной плоскости и направление проецирования не совпадают.

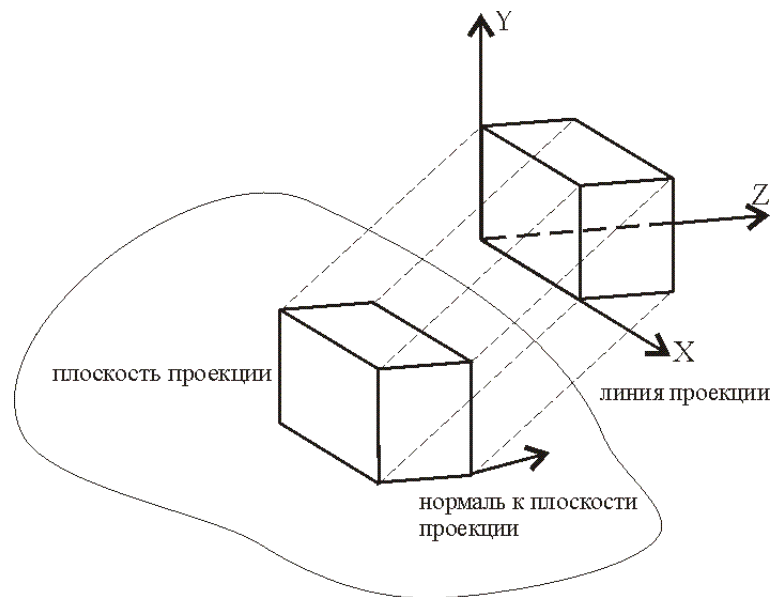


Рис. 11. Косоугольная проекция

Особый интерес среди косоугольных проекций представляют две проекции – *кавалье* и *кабине*.

В проекции кавалье направление проецирования составляет с плоскостью проецирования угол 45° . В этой проекции коэффициенты искажения для всех трех главных направлений одинаковы, поэтому проекцию кавалье еще называют горизонтальной косоугольной изометрией (военная перспектива). На рисунке показана проекция кавалье единичного куба на плоскость $Z=0$.

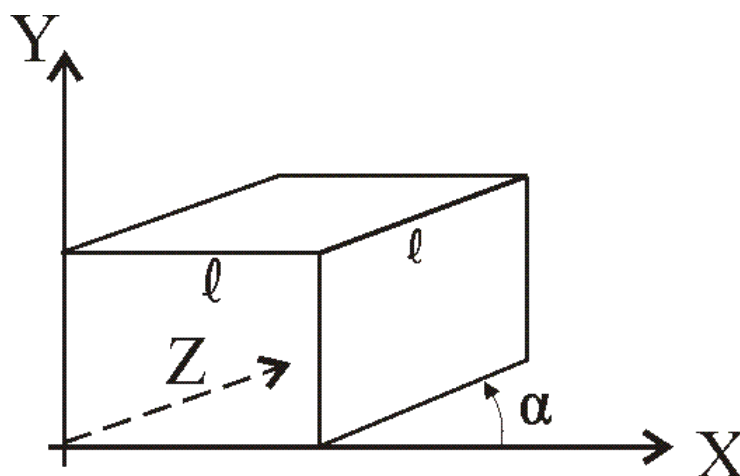


Рис. 12. Проекция кавалье

Уходящие как бы вглубь линии являются проекциями тех ребер куба, которые перпендикулярны плоскости $Z=0$. Они расположены под углом α к горизонтали. Обычно этот угол составляет 30° или 45° . Очевидно, что результат данного проецирования выглядит неестественно утолщенным.

Проекция кабина имеет направление проецирования, которое составляет с проекционной плоскостью угол $\arccotg\left(\frac{1}{2}\right) \approx 63.43^\circ$. При этом отрезки, перпендикулярные проекционной плоскости, после проецирования составляют $\frac{1}{2}$ их действительной длины.

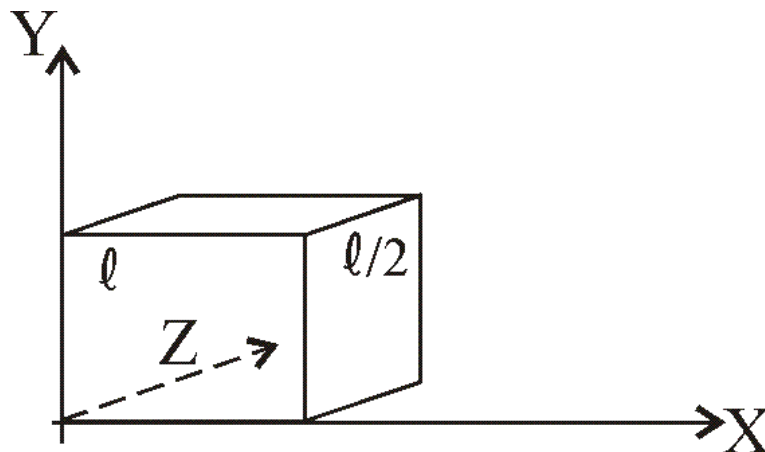


Рис. 13. Проекция кабина

Таким образом у этих проекций коэффициент искажения d для ребер, перпендикулярных плоскости проекции, равен $\frac{1}{2}$. В отечественной литературе проекция кабина называется фронтальной косоугольной диметрией (кабинетная проекция). Проекция кабина являются более реалистичными, чем проекции кавалье, поскольку укорачивание боковых граней с коэффициентом $\frac{1}{2}$ больше согласуется с нашим визуальным опытом.

Чтобы построить матрицу преобразования для косоугольной проекции, рассмотрим единичный вектор вдоль оси Z (показан на рисунке). Как показано на рис. при косоугольном проецировании нормаль к плоскости проекции составляет определенный угол с линиями проецирования.

Угол β – угол между линией проекции PP' и плоскостью проекции $Z = 0$. $\overline{OP'}$ – проекция единичного вектора \overline{OP} на плоскость $Z = 0$ в заданной косоугольной системе координат. Угол α – угол между проекцией вектора и

горизонтальной осью X. Обозначим длину спроецированного единичного вектора $|\overline{OP'}| = d$. По условиям решаемой задачи d – коэффициент искажения. Из рисунка следует, что $\operatorname{tg} \beta = \frac{|\overline{OP'}|}{|\overline{OP}|} = \frac{d}{1} = d$, т. е. $\beta = \operatorname{arctg} d$.

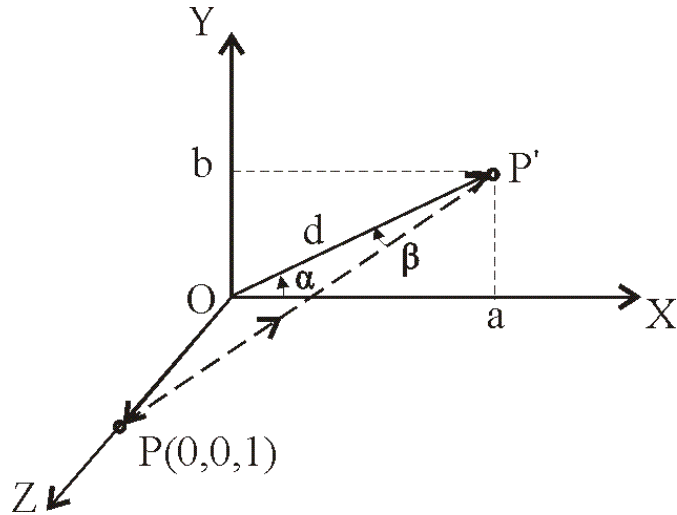


Рис. 14. Построение косоугольной проекции

Проекцию OP' можно получить из OP с помощью преобразований сдвига вектора \overline{OP} в направлениях X и Y. Для этого необходимо использовать матрицу:

$$(T_1) = \begin{pmatrix} 1 & 0 & a & 0 \\ 0 & 0 & b & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

При проецировании на плоскость $Z = 0$ результирующая матрица примет вид:

$$(T_2) = \begin{pmatrix} 1 & 0 & a & 0 \\ 0 & 0 & b & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Поскольку из иллюстрации следует, что

$$\begin{cases} a = d \cos \alpha \\ b = d \sin \alpha \end{cases}$$

Окончательная матрица преобразований запишется следующим образом:

$$(T) = \begin{pmatrix} 1 & 0 & d \cos \alpha & 0 \\ 0 & 0 & d \sin \alpha & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

При заданном значении коэффициента искажения d можно получить различные проекции единичного вектора на плоскость $Z = 0$.

При $d = 0 \rightarrow \beta = 90^\circ$, т.е. переход к ортографической проекции.

Если $d = 1$, в этом случае не подвергаются искажению ребра, перпендикулярные плоскости проекции, а $\beta = \text{arctg } 1 = 45^\circ$, т.е. имеет место проекция кавалье. Проекция кабина получается при $d = \frac{1}{2}$, т.е.

$\beta = \text{arctg } \frac{1}{2} \approx 63.43^\circ$. В матрице преобразования угол α является свободным параметром. Как отмечалось выше, наиболее часто значения α выбирают равными 30° и 45° .

Перспективные (центральные) проекции представляют собой преобразования из трехмерного пространства в двухмерное. При этом используется понятие *центра проекции* (точка Q). Проекция объекта строится при помощи прямых проецирующих лучей, называемых *линиями проекции* или *проекторами*. Линии проекции выходят из центра проекции, проходят через каждую точку объекта и, пересекая плоскость проекции, образуют проекцию объекта на эту плоскость. На рисунке показаны два возможных варианта расположения центра проекции, плоскости проекции и объекта.

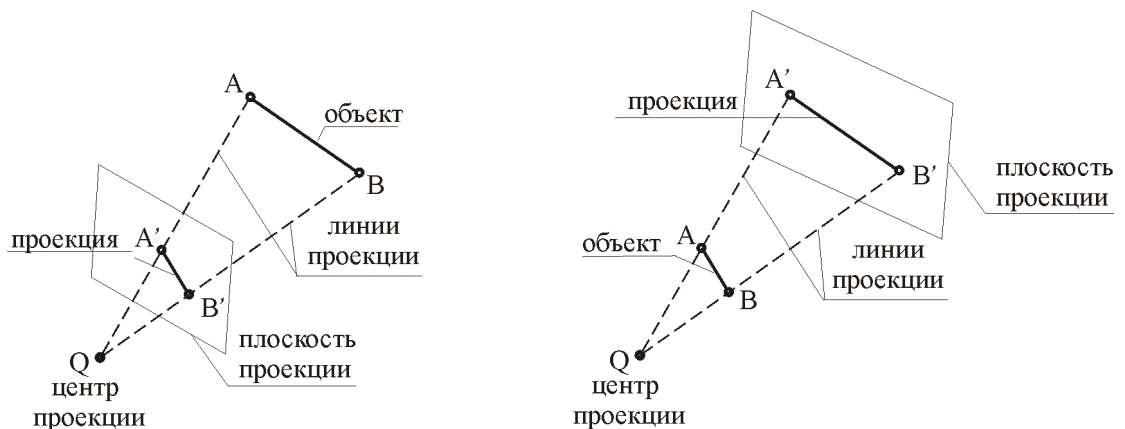


Рис. 15. Перспективные проекции

Рассмотрим некоторую точку объекта P (определенную вектором \vec{r}) в пространственной системе координат и при заданном центре проекции Q (вектор \vec{r}_Q).

Найдем центральную проекцию этой точки. В рассматриваемом случае плоскость проекции расположена между центром проекции и объектом. Зададим фиксированное направление из точки P в точку Q с помощью единичного вектора \vec{u}_3 .

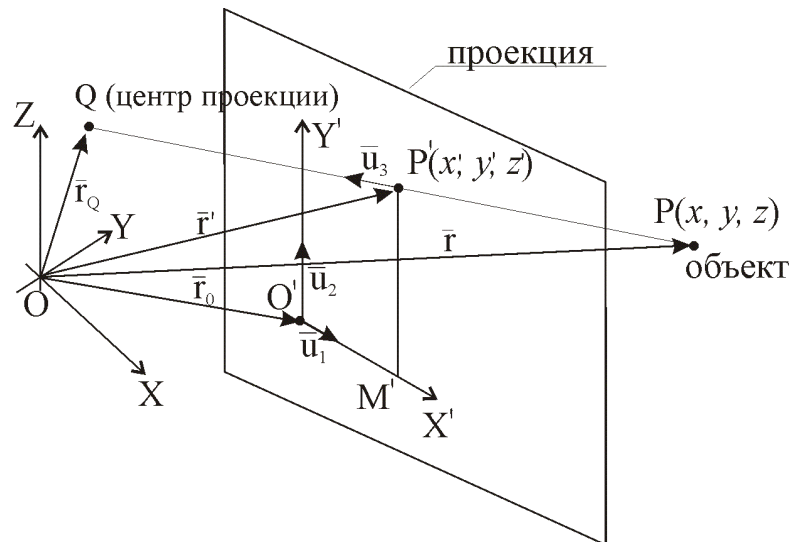


Рис. 16. Построение перспективной проекции

Для фиксации плоскости проекции в пространстве зададим вектор \vec{r}_0 , поместив его начало в пространственную систему координат, а конец в выбранную точку на плоскости проекции. Таким образом определена точка O' на плоскости проекции. Направление осей координат на плоскости проекции определяется единичными векторами \vec{u}_1 и \vec{u}_2 соответственно для осей X' и Y' .

Точки Q , P и P' лежат на одной прямой, следовательно, для некоторого значения z' :

$$\vec{r}' = z'\vec{r} + (1 - z')\vec{r}_Q$$

С другой стороны из рисунка следует, что

$$\vec{r}' = \vec{OO'} + \vec{OM'} + \vec{M'P'}, \text{ или } \vec{r}' = \vec{r}_0 + x'\vec{u}_1 + y'\vec{u}_2$$

Объединив полученные уравнения, получим

$$\vec{r}' = \vec{r}_0 + x'\vec{u}_1 + y'\vec{u}_2 = z'\vec{r} + (1 - z')\vec{r}_Q$$

Умножив скалярно последнее уравнение на векторы $[\overline{u_2} \times (\overline{r} - \overline{r_Q})]$, $[\overline{u_1} \times (\overline{r} - \overline{r_Q})]$ и $[\overline{u_3} \times \overline{u_2}]$, после преобразований (с учетом свойств смешанного произведения векторов) будем иметь:

$$x' = \frac{(\overline{r} - \overline{r_Q}) \cdot [\overline{u_2} \times (\overline{r_0} - \overline{r_Q})]}{(\overline{r} - \overline{r_Q}) \cdot [\overline{u_1} \times \overline{u_2}]}$$

$$y' = \frac{(\overline{r} - \overline{r_Q}) \cdot [\overline{u_1} \times (\overline{r_0} - \overline{r_Q})]}{(\overline{r} - \overline{r_Q}) \cdot [\overline{u_2} \times \overline{u_1}]}$$

$$z' = \frac{(\overline{r_0} - \overline{r_Q}) \cdot [\overline{u_1} \times \overline{u_2}]}{(\overline{r} - \overline{r_Q}) \cdot [\overline{u_1} \times \overline{u_2}]}$$

Обычно начало координат на плоскости проекции (точка O') выбирается таким образом, чтобы прямая, соединяющая точку центра проекции с точкой O' была перпендикулярна плоскости проекции. Если расстояние от точки Q до плоскости проекции равно d_0 , то из рисунка следует, что:

$$\overline{u_3} = \overline{u_1} \times \overline{u_2}$$

$$\overline{r_Q} = \overline{r_0} + d_0 \cdot \overline{u_3}$$

Подставив полученные выражения в уравнения для x' , y' и z' получим:

$$x' = \frac{d_0(\overline{r} - \overline{r_0}) \cdot \overline{u_1}}{(\overline{r} - \overline{r_0}) \cdot \overline{u_3} - d}$$

$$y' = \frac{d_0(\overline{r} - \overline{r_0}) \cdot \overline{u_2}}{(\overline{r} - \overline{r_0}) \cdot \overline{u_3} - d}$$

$$z' = -\frac{d_0}{(\overline{r} - \overline{r_0}) \cdot \overline{u_3} - d}$$

Пологая $\omega' = d_0 - (\overline{r} - \overline{r_0}) \cdot \overline{u}$, полученные уравнения можно представить в следующем виде:

$$\overline{r'} = (V) \cdot (A) \cdot (T) \cdot \overline{r}$$

$$\text{где } \overline{r'} = \begin{pmatrix} x' \omega' \\ y' \omega' \\ z' \omega' \\ \omega' \end{pmatrix}, \overline{r} = \begin{pmatrix} x \omega \\ y \omega \\ z \omega \\ \omega \end{pmatrix}$$

$$(V) = \begin{pmatrix} d_0 & 0 & 0 & 0 \\ 0 & d_0 & 0 & 0 \\ 0 & 0 & 0 & d_0 \\ 0 & 0 & -1 & d_0 \end{pmatrix} - \text{матрица проективного преобразования}$$

$$(A) = \begin{pmatrix} u_{1x} & u_{1y} & u_{1z} & 0 \\ u_{2x} & u_{2y} & u_{2z} & 0 \\ u_{3x} & u_{3y} & u_{3z} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} - \begin{array}{l} \text{матрица вращения осей плоскости} \\ \text{проекции до совпадения этих осей с} \\ \text{осями } OX \text{ и } OY \end{array}$$

$$(T) = \begin{pmatrix} 1 & 0 & 0 & -x_0\omega_0 \\ 0 & 1 & 0 & -y_0\omega_0 \\ 0 & 0 & 1 & -z_0\omega_0 \\ 0 & 0 & 0 & \omega_0 \end{pmatrix} - \begin{array}{l} \text{матрица переноса плоскости проекции} \\ \text{в координатную плоскость } XOY \\ \text{пространственной системы координат} \end{array}$$

Если плоскость проекции уже занимает это стандартное положение, то перспективные преобразования определяются по формуле:

$$\vec{r}' = (V) \cdot \vec{r} = \begin{pmatrix} d_0 & 0 & 0 & 0 \\ 0 & d_0 & 0 & 0 \\ 0 & 0 & 0 & d_0 \\ 0 & 0 & -1 & d_0 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} xd_0 \\ yd_0 \\ d_0 \\ d_0 - z \end{pmatrix}, \text{ те}$$

$$x' = \frac{xd_0}{d_0 - z}; y' = \frac{yd_0}{d_0 - z}; z' = \frac{d_0}{d_0 - z}.$$

Центральные проекции параллельных прямых, которые не параллельны плоскости проекции, будут сходиться в *точке схода*. Параллельные прямые пересекаются лишь в бесконечности, поэтому точку схода можно представить как проекцию точки, находящейся в бесконечности. Если совокупность параллельных прямых параллельна одной из главных координатных осей, то их точка схода называется *главной точкой схода*. Центральные проекции классифицируются в зависимости от числа главных точек схода, которыми они обладают, а, следовательно, от числа координатных осей, которые пересекают плоскость проекции.

Выбор необходимых преобразований и проекций

В результате мы получили набор различных трехмерных преобразований, из которых нам понадобятся:

- Перенос (для возможности увеличения-уменьшения модели);

- Поворот вокруг произвольного вектора (для возможности вращения модели).

В подборе проекции мы остановимся на центральной проекции с центром проекции расположенным за объектом, так как этот тип проекции в наибольшей мере соответствует восприятию трехмерных объектов в окружающем нас мире (присутствует перспектива).

Интерполяция и текстурирование

Одной из основных задач визуализации поверхностного ЭКГ картирования является интерполяция данных, поступающих от датчиков, на всю поверхность тела. При этом, датчики могут располагаться произвольным образом, и в независимости от этого, мы должны уметь получать общую картину.

Интерполяция данных с датчиков

Так как показания датчиков изменяются во времени в интервале $t_0 \cdots t_k$, где k – общее количество отсчетов. Рассмотрим задачу в момент времени t_0 . Опишем набор датчиков следующим образом:

$$s_i = \begin{bmatrix} x \\ y \\ v \end{bmatrix}, i \in [1 \cdots m],$$

где m – общее количество датчиков, x – абсцисса положения датчика, y – ордината положения датчика, v – показание датчика в исследуемый момент времени (t_0).

С учетом особенностей современных систем отображения графики полученная полная карта должна представлять собой квадратную матрицу M размерности $l = 2^n$, где n – произвольное натуральное число, не превышающее определенного значения, специфичного для каждой отдельно взятой системы. В связи с этим введем ограничения на координаты положения датчика:

$$1 \leq x \leq l, 1 \leq y \leq l; x, y \in \mathbb{N}.$$

Введя данное ограничение, мы можем определить значения карты в точках, которые соответствуют датчикам, а, точнее, определить значения следующих элементов матрицы:

$$M_{x,y} = s_i^y, \text{ при } x = s_i^x, x = s_i^y, i \in [0 \dots m].$$

После этого мы можем приступить к решению задачи интерполяции. Учтем, что наша карта должна «замыкаться», то есть введем ограничения:

$$M_{1,y} = M_{l,y}, \text{ при } 1 \leq y \leq l,$$

а так же будем требовать от метода интерполяции инвариантность значений элементов матрицы, соответствующих датчикам, а так же оптимальную вычислительную сложность, при условии, что положения датчиков фиксированные, а значения изменяются во времени.

В основе нашего метода будет лежать расчет весов: нам нужно будет определить, насколько показание того или иного датчика влияет на интерполируемый элемент матрицы (а, соответственно, на точку на карте). Возьмем за основу расстояние от интерполируемой точки до каждого отдельно взятого датчика. Так как нам известны координаты s_i^x и s_i^y i -го датчика, а так же координаты x и y интерполируемой точки, то мы можем посчитать отклонение по каждой из координатных осей:

$$\begin{aligned} \Delta x &= |x - s_i^x| \\ \Delta y &= |y - s_i^y| \end{aligned}$$

Однако, приняв во внимание тот факт, что карта должна замыкаться, а так же, ограничения на Δx и Δy , которые следуют из ограничений на x и y :

$$0 \leq \Delta x < l, 0 \leq \Delta y < l; \Delta x, \Delta y \in \mathbb{N},$$

мы можем дополнить нашу матрицу следующими элементами:

$$M_{x-l,y} = M_{x+l,y} = M_{x,y}.$$

Таким образом, из одной точки мы получаем три идентичные, и в качестве итогового расстояния мы можем взять минимальное из расстояний, до каждой из этих точек. Приняв во внимание «наложения» значений датчиков, мы может показать, что

$$M_{s_i^x-l, s_i^y} = M_{s_i^x+l, s_i^y} = M_{s_i^x, s_i^y}.$$

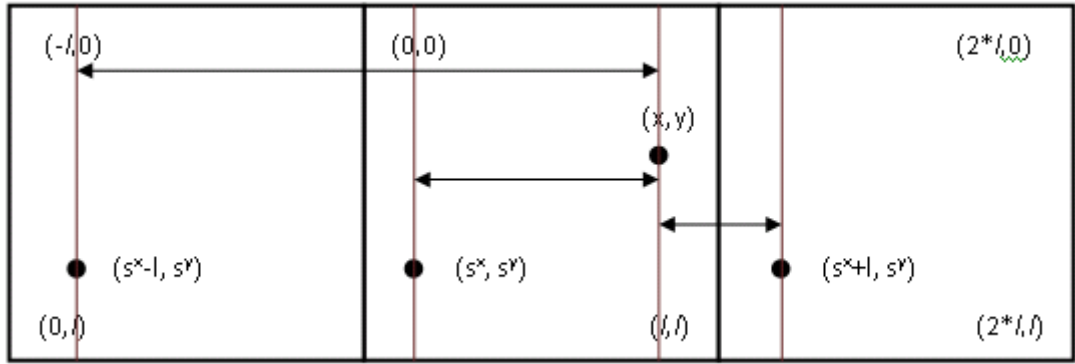


Рис. 17. Определение расстояния по оси x

С учетом вышесказанного, мы можем получить итоговые формулы для расчета расстояний вдоль координатных осей:

$$\Delta x = \min(|x - s_i^x|, |x - (s_i^x - l)|, |x - (s_i^x + l)|) \geq 0$$

$$\Delta y = \min(|y - s_i^y|, |y - (s_i^y - l)|, |y - (s_i^y + l)|) \geq 0$$

Введем функцию $w(\Delta x, \Delta y)$ – определение веса, такую что:

$$w(\Delta x, \Delta y) \geq 0; w(0, 0) = 0.$$

А так же введем набор весов:

$$W_{i,j,k} = w(\Delta x, \Delta y); x = i, y = j, s^{x,y} = s_k^{i,j}.$$

Однако, стоит заметить, что в данном случае *большой* вес означает *малый* вклад в значение, а, соответственно, если вес равен 0 – то вклад значения данного датчика для данной точки наибольший, и при этом никакие другие точки свой вклад не вносят. Тем самым, мы удовлетворили оба условия: условие «замыкания» по оси x и условие инвариантности значений.

Для дальнейшего использования преобразуем наши веса таким образом, чтобы удовлетворялись следующие условия:

1. Большой вес вносит большой вклад;
2. $\forall i, j \in [1 \dots l] \sum_{k=1}^m W_{i,j,k} = 1.$

Выполним для этого следующий набор действий:

1. Нормализация: $\forall i, j \in [1 \dots l] \forall k \in [1 \dots m] W_{i,j,k} = \frac{W_{i,j,k}}{\sum_{k=1}^m W_{i,j,k}};$

$$2. \text{ Инвертирование: } \forall i, j \in [1 \dots l] \forall k \in [1 \dots m] W_{i,j,k} = \begin{cases} \frac{1}{W_{i,j,k}}, & W_{i,j,k} \neq 0; \\ +\infty, & \text{иначе} \end{cases};$$

$$3. \text{ Повторная нормализация, причем если } \exists k \in [1 \dots m] W_{i,j,k} = +\infty, \text{ то } \forall p \in [1 \dots m]: p \neq k W_{i,j,p} = 0; W_{i,j,k} = 1.$$

Таким образом, наиболее близкие датчики будут вносить наибольший вклад в значение точки, а в точках, соответствующих датчикам, вклад будет вносить только он, то есть мы удовлетворили первое условие. К тому же, так как мы в конце выполнили нормализацию, то второе условие так же будет выполнено.

Используя полученную матрицу весов, рассчитаем значения по формуле:

$$\forall x, y \in [1 \dots l] M_{x,y} = \sum_{k=1}^m W_{x,y,k} \cdot M_{s_k^x, s_k^y} = \sum_{k=1}^m W_{x,y,k} \cdot s_k^y.$$

При условии нормализации весов мы можем гарантировать, что значения во всех вычисленных точках будут лежать в промежутке $[\min_k(s_k^y) \dots \max_k(s_k^y)]$. Докажем это утверждение:

Для начала докажем, что значение не превысит максимума. Пусть максимум достигается в некоторой точке t . Тогда значение максимума будет равно s_t^y . Запишем условие, которому удовлетворят данная точка:

$$\max_k(s_k^y) = s_t^y; \forall k \in [1 \dots m], k \neq t: s_k^y \leq s_t^y.$$

В таком случае для любого датчика справедливо уравнение:

$$\forall k \in [1 \dots m]: s_k^y = a_k s_t^y; a_k \leq 1.$$

Следовательно:

$$\forall x, y \in [1 \dots l] M_{x,y} = \sum_{k=1}^m W_{x,y,k} \cdot s_k^y = \sum_{k=1}^m W_{x,y,k} \cdot a_k s_t^y = s_t^y \sum_{k=1}^m W_{x,y,k} \cdot a_k.$$

Рассмотрим каждое из слагаемых по отдельности:

$$a_k \leq 1 \Rightarrow \forall k \in [1 \dots m], \forall x, y \in [1 \dots l]: W_{x,y,k} a_k \leq W_{x,y,k},$$

то есть каждое из отдельно взятых слагаемых *меньше или равно*, чем слагаемые суммы:

$$\forall x, y \in [1 \dots l]: \sum_{k=1}^m W_{x,y,k} = 1.$$

Таким образом, мы получили, что:

$$\forall x, y \in [1 \dots l]: \sum_{k=1}^m W_{x,y,k} \cdot a_k \leq 1.$$

А это означает, что:

$$\forall x, y \in [1 \dots l] M_{x,y} \leq s_t^v.$$

Доказательство того, что $M_{x,y} \geq \min_k(s_k^v)$ производится аналогичным способом.

Генерация текстуры по полученным данным

После того, как мы получим матрицу $M_{x,y}$ нам необходимо отобразить её на экране. Для этого каждому значению мы должны присвоить определенный цвет. По спецификации датчиков, их значения варьируются от -255 до 255. А из вышепоказанного факта следует, что значения всех элементов полученной матрицы будут лежать в этом интервале.

В общем случае цвет точки задается по формуле:

$$\forall x, y \in [1 \dots l]: color_{x,y} = \{ red(M_{x,y}), green(M_{x,y}), blue(M_{x,y}) \},$$

то есть нам необходимо отдельно посчитать каждую из трех компонент цвета. Наиболее гибкий способ – это задать значения компонент цвета для всех возможных значений $M_{x,y}$ (а их конечное число). Однако вследствие того, что в случае необходимости изменения цветовой палитры необходимо будет проделать достаточный объем работ, от использования данного метода было решено отказаться.

Для описания цветовой палитры укажем лишь значения компонент в нескольких точках (то есть опишем цвет для определенных значений), а значения компонент для всех остальных значений посчитаем путём линейной интерполяции.

Значение каждой их компонент будем рассчитывать по следующей формуле:

$$P_i(M_{x,y}) = \left\{ P : \max_{val(P) \leq M_{x,y}} val(P) \right\} - \text{нижняя граница интервала}$$

$$P_{i+1}(M_{x,y}) = \left\{ P : \min_{val(P) \geq M_{x,y}} val(P) \right\} - \text{верхняя граница интервала}$$

$$red(M_{x,y}) = red(P_i(M_{x,y})) + \frac{M_{x,y} - val(P_i(M_{x,y}))}{val(P_{i+1}(M_{x,y})) - val(P_i(M_{x,y}))} \cdot (red(P_{i+1}(M_{x,y})) - red(P_i(M_{x,y})));$$

$$green(M_{x,y}) = green(P_i(M_{x,y})) + \frac{M_{x,y} - val(P_i(M_{x,y}))}{val(P_{i+1}(M_{x,y})) - val(P_i(M_{x,y}))} \cdot (green(P_{i+1}(M_{x,y})) - green(P_i(M_{x,y})));$$

$$blue(M_{x,y}) = blue(P_i(M_{x,y})) + \frac{M_{x,y} - val(P_i(M_{x,y}))}{val(P_{i+1}(M_{x,y})) - val(P_i(M_{x,y}))} \cdot (blue(P_{i+1}(M_{x,y})) - blue(P_i(M_{x,y})));$$

Значения $val(P)$, $red(P)$, $green(P)$ и $blue(P)$ берем из таблицы:

Значение $val(P)$	$red(P)$	$green(P)$	$blue(P)$
-255	0	28	131
-127	24	66	155
-64	20	108	180
-32	39	171	125
-16	15	162	115
-8	3	154	104
-4	28	155	31

-2	19	148	19
0	204	227	0
2	207	207	0
4	233	152	0
8	235	122	0
16	237	85	10
32	193	46	20
64	176	14	5
127	147	16	9
256	102	8	15

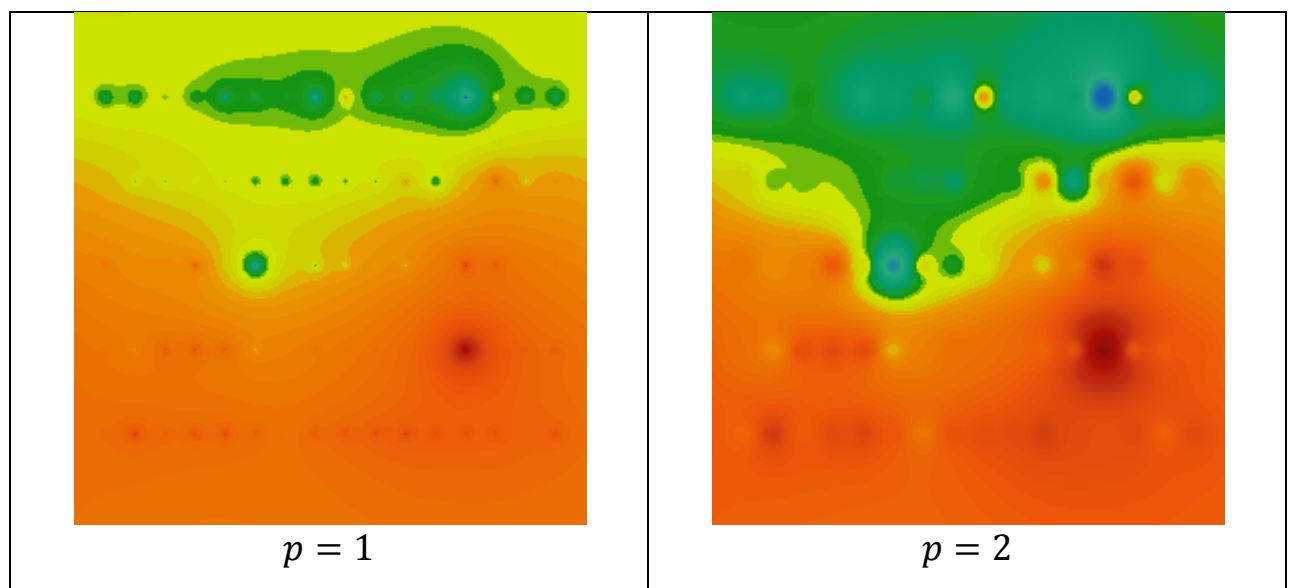
Табл. 1. Список цветов для различных значений показаний датчика

Имея данный инструмент отображения карты, мы можем визуально оценивать подбор тех или иных параметров визуализации. Так, например, нам необходимо подобрать функцию $w(\Delta x, \Delta y)$, удовлетворяющую заданным условиям.

Полученный метод является вариантом метода интерполяции по обратному средневзвешенному. Поэтому будем подбирать функцию $w(\Delta x, \Delta y)$ из класса функций, описанных в этом методе:

$$w(\Delta x, \Delta y) = \left(\sqrt{\Delta x^2 + \Delta y^2} \right)^p, \text{ } p - \text{коэффициент}$$

Рассмотрим карту при различных функциях $w(\Delta x, \Delta y)$:



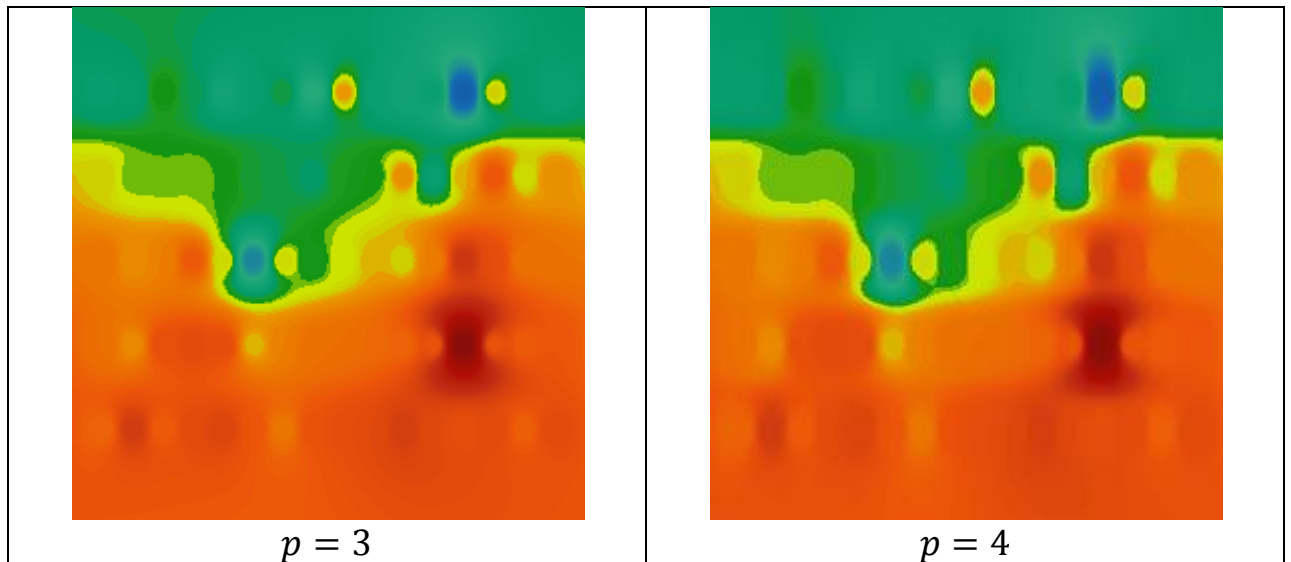


Рис. 18. Интерполяция по обратному средневзвешенному

Отметим, что наиболее гладкая картина наблюдается при использовании функций $w(\Delta x, \Delta y) = (\sqrt{\Delta x^2 + \Delta y^2})^3$ ($p = 3$) и $w(\Delta x, \Delta y) = (\Delta x^2 + \Delta y^2)^2$.

Заметим, что при использовании функции $w(\Delta x, \Delta y) = (\Delta x^2 + \Delta y^2)^2$ «область влияния» каждой из точек уменьшается, а при $w(\Delta x, \Delta y) = \Delta x^2 + \Delta y^2$ она слишком большая, чем и обусловлена «локальность» (в тех точках, где размещены датчики значение равно показанию датчика, а на всей остальной поверхности значение размыто).

У метода интерполяции по обратному средневзвешенному есть улучшенная версия, которая называется метод Шепарда^[5]. В нем используется следующая функция:

$$w(\Delta x, \Delta y) = \left(\frac{R_x \sqrt{\Delta x^2 + \Delta y^2}}{R_x - \sqrt{\Delta x^2 + \Delta y^2}} \right)^2, \quad R_x = \max_{i \in K} \sqrt{\Delta x_i^2 + \Delta y_i^2},$$

K – подмножество известных точек, n самых близких

В данной функции мы рассматриваем не все известные нам точки (в которых расположены датчики), а только n самых близких к рассматриваемой точке, причем этот параметр подбирается перебором. К тому же эта функция обладает следующим свойством: её значение в самой дальней из n самых близких точек устремляется к бесконечности.

Рассмотрим картину, при использовании данной функции:

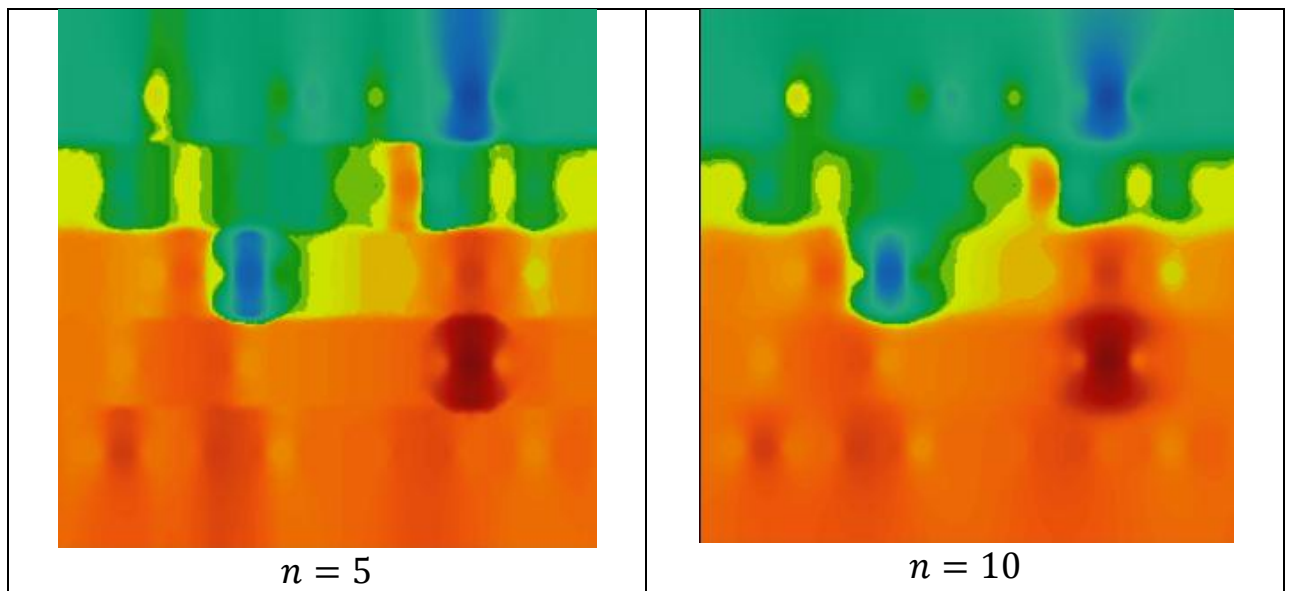


Рис. 19. Интерполяция Шепарда

Мы остановим наш выбор на данной функции со значением параметра $n = 10$, так как при меньших значениях мы наблюдаем сильную «локальность».

К тому же использование данной функции позволило нам сократить время построения каждого кадра, так как теперь для построения требуется всего 10 итераций, вместо 80, так как для расчета каждой точки используется всего 10 известных точек, а коэффициенты остальных равны нулю.

Наложение текстуры на модель

Построим трехмерную модель торса. Для начала модель будет представлять собой набор вершин в трехмерном пространстве, а так же набор ребер, их соединяющих.

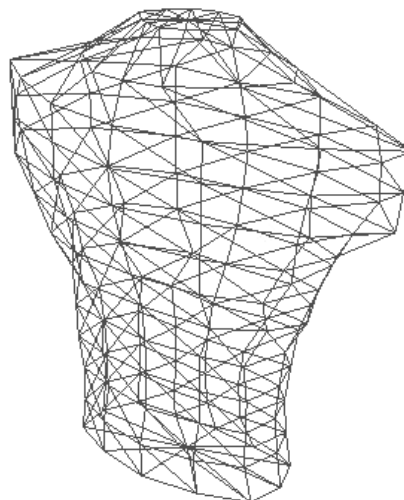


Рис. 20. Трехмерная модель человеческого торса

Полученные ребра можно объединить в треугольники. Таким образом, мы получим набор полигонов. На каждый полигон нам необходимо наложить часть полученной текстуры^[1]. Для этого сопоставим каждую вершину точке на текстуре. Так как полученная нами текстура имеет одинаковые значения на левом и на правом крае, то при наложении она замкнется. В результате мы получим следующую модель:

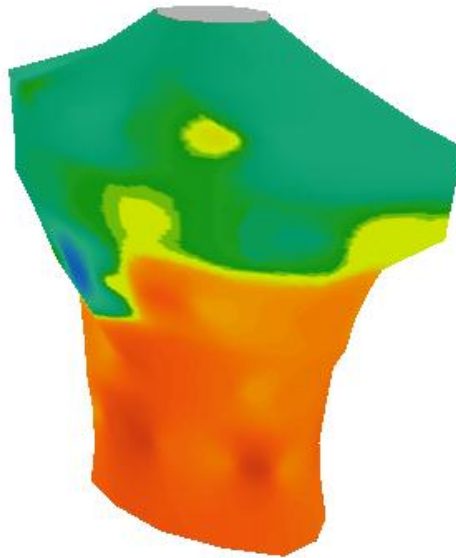


Рис. 21. Трехмерная модель торса с наложенной текстурой

Программная реализация и обсуждение результатов

Перейдем к программной реализации. Помимо непосредственной визуализации программа должна обладать следующими функциями:

1. наличие графического интерфейса пользователя;
2. возможность изменения конфигурации датчиков;
3. чтение данных с файлов импорта аппарата поверхностного ЭКГ картирования.

Для реализации данного функционала мы будем использовать технологию OpenGL^[2]. Выбор обусловлен кроссплатформенностью данного решения, в отличие от технологии DirectX. Так же мы будем использовать библиотеку GLUT, которая предоставляет нам расширение функционала стандартной библиотеки GL.

Начнем с проектирования интерфейса. Нам необходимо отображать различные элементы интерфейса в зависимости от того, что пользователь делает в данный момент. Для этого введем понятие режима работы, который может принимать одно из нижеперечисленных значений:

- простой просмотр;
- визуализация (анимация);
- ввод конфигурации датчиков.

Режим работы может быть дополнен режимом визуализации. Нами были предусмотрены два режима визуализации:

- просмотр карты потенциала;
- просмотр изопотенциальных линий.

Интерфейс в режиме просмотра выглядит следующим образом:

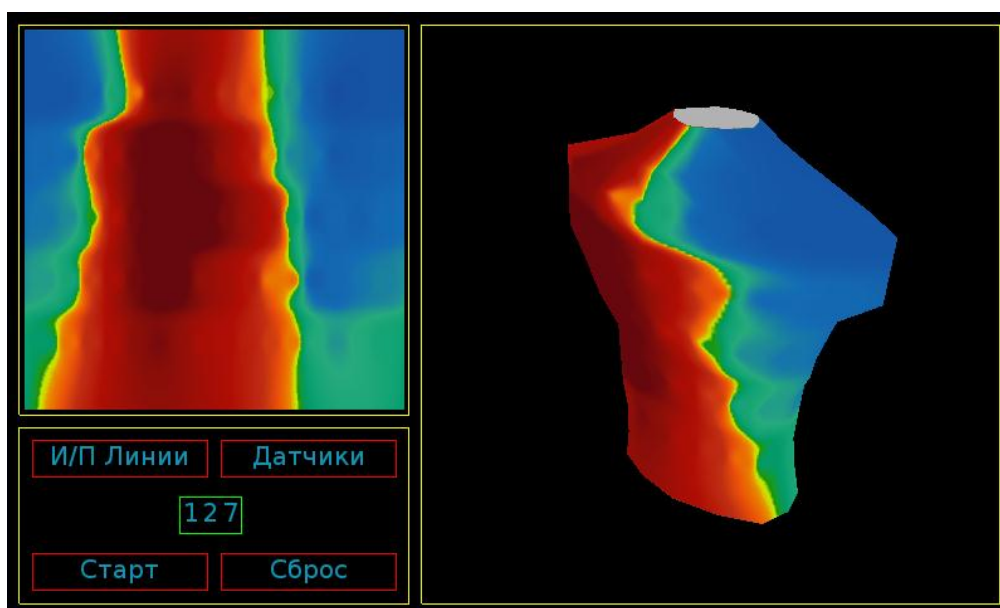


Рис. 22. Режим просмотра

Интерфейс состоит из трех основных частей. Слева вверху расположено окно двухмерной визуализации, слева внизу – кнопки управления и счетчик, показывающий, на каком именно отсчете мы находимся. Каждая из кнопок отвечает за следующие функции:

- И/П Линии – включит отображение изопотенциальных линий;
- Датчики – перейти в режим редактирования датчиков;
- Старт – запустить анимацию;

- Сброс – сбросить счетчик отсчетов.

Справа расположено основное окно визуализации. При этом поддерживается вращение модели при помощи мыши – для этого необходимо зажать левую кнопку и перемещать мышь. При помощи колесика мыши мы можем приближать и отдалять модель. Данные функции продублированы на клавиатуре: стрелки и кнопки «+» и «-» соответственно.

В текущей цветовой схеме красным цветом обозначены места с положительным потенциалом, синим – с отрицательным. Желтый цвет означает потенциал близкий к 0. А оранжевый и голубой – небольшие значение положительного и отрицательного потенциала соответственно.

В режиме отображения изопотенциальных кривых интерфейс выглядит следующим образом:

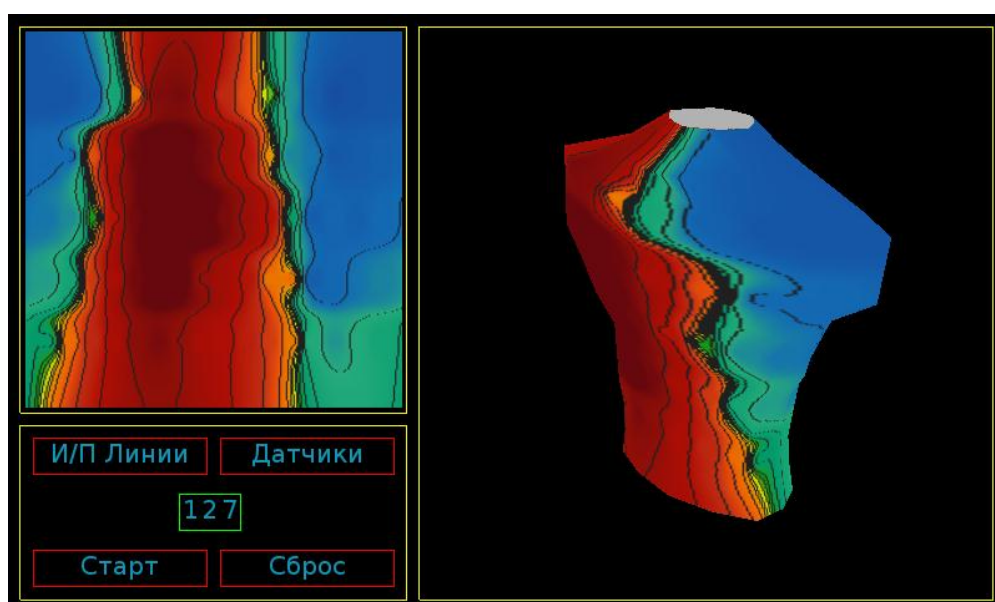


Рис. 23. Изопотенциальные кривые

Каждая изопотенциальная кривая соответствует значению потенциала из первого столбца таблицы 1.

Перейдем к режиму редактирования положения датчиков. В этом режиме интерфейс программы меняется на следующий:

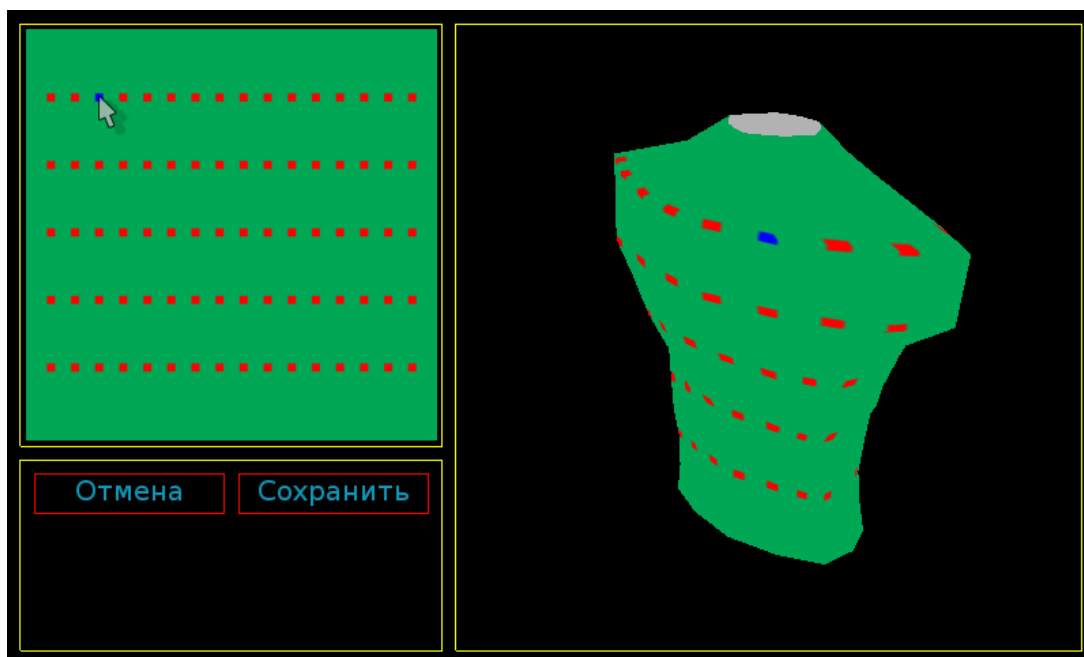


Рис. 24. Редактирование датчиков

В этом режиме мы можем редактировать положения датчиков. Редактирование производится на двухмерной развертке, при этом в реальном времени все изменения отражаются на трехмерной модели, которую мы так же можем вращать и увеличивать.

Доступно два режима редактирования положения датчиков:

- по одному – необходимо навести указатель мыши на датчик, зажать левую кнопку и переместить датчик;
- все сразу – необходимо зажать правую кнопку мыши в любом месте на двухмерной развертке и перетащить его.

После редактирования мы так же можем отменить их, или сохранить. При сохранении будет произведен пересчет текстуры и программа перейдет в режим просмотра.

Таким образом, мы реализовали весь необходимый функционал. Используемые методы позволяют получить трехмерную модель торса, которую можно просматривать под разными углами с разным увеличением, позволяет просматривать изменения показаний кардиодатчиков на разных этапах цикла работы сердца. Полученные таким образом визуальные наблюдения позволяют медицинским работникам уточнять диагноз пациентов.

Выводы

В результате выполнения данной дипломной работы была разработана часть аппаратно-программного комплекса для проведения поверхностного ЭКГ картирования, а именно – приложение для трехмерной визуализации результатов. Аппаратно-программный комплекс на данный момент состоит из следующих частей:

- аппарат поверхностного ЭКГ картирования Cardiac 112.2;
- программа регистрации потенциала на поверхности грудной клетки;
- программа фильтрации шумов;
- программа визуализации.

Полученный аппаратно-программный комплекс позволяет получать данные пациентов, а так же наглядно визуализировать результаты при помощи двухмерной и трехмерной модели визуализации.

Разработанная программа визуализации является полностью настраиваемым. Пользователь может загружать различные входные данные, конфигурации расположения датчиков, а так же загружать различные трехмерные модели и цветовые схемы.

Полученное решение является кроссплатформенным и может работать под операционными системами семейства Microsoft Windows и Linux, при наличии установленных библиотек OpenGL и FreeGLUT.

Данную разработку можно применять при исследовании состояния пациентов и помощи врачу-кардиологу при визуальном исследовании карт.

Литература и гиперссылки

- [1]. Порев В. Н. Компьютерная графика. – СПб.: БХВ-Петербург, 2004. – 432с.
- [2]. Ричард С. Райт-мл, Бенджамин Липчак. OpenGL. Суперкнига, 3-е издание.: Пер с англ. – М.: Издательский дом «Вильямс», 2006. – 1040 с.
- [3]. Д. Роджерс, Дж. Адамс. Математические основы машиной графики: Пер. с англ. – М.: Мир, 2001. – 604 с.
- [4]. Хилл Ф. OpenGL. Программирование компьютерной графики. Для профессионалов. – СПб.: Питер, 2002. – 1088 с.
- [5]. http://www.emsi.com/smshelp/Data_Module/Interpolation/Inverse_Distance_Weighted.htm