本周工作：

1. 整理文档内容。
2. 二维观察流水线与坐标系的建立。

通常，在构造和显示一个场景的过程中会使用几个不同的笛卡儿参照系。对于复杂的场景，每个图形对象根据自己的形状在各自的坐标系统中构造。这些坐标系称为建模坐标系或局部坐标系。对于不是很复杂的场景，对象的各部分可以直接在世界坐标系中建立，从而跳过建模坐标和建模变换两步。本系统的图形对象都为四边形，因此数据库直接提供摄像头在世界坐标系中的配准坐标点即可。

在输出设备上观察一个二维或三维的场景，要将场景的世界坐标描述经各种处理变换到一个或多个输出设备参照系来显示，这个过程称为观察流水线。世界坐标系位置首先转换到与我们要对场景进行的观察所对应的观察坐标系，该转换依据观察者的位置和方向而进行。接着，观察坐标系依照世界窗口对场景进行裁剪，再变换到视口中。最后，视口中的图形经扫描转换到光栅系统的刷新缓存中进行显示。显示设备的坐标系称为设备坐标系或对视频监视器而言称为屏幕坐标系。



完整二维观察流水线

在此基础上，我们系统中的显示设备还有电视墙，因此我们需要建立拼接屏坐标系和单元屏坐标系，并完成视口到拼接屏坐标系的转换。

为了便于场景建模和图形的观察与显示，引入以下的坐标系：

1. 世界坐标系 (背景图坐标系)

世界坐标用于为特定图形世界建立模型，为了描述图形，必须首先确定一个世界坐标系，接着通过给出世界坐标系中的位置等几何描述来定义图形中的对象，数据库提供的摄像头标定参数是本系统中的图形对象，摄像头标定点坐标是基于世界坐标系的。本系统以背景图为基准建立二维笛卡儿坐标系，以背景左上角为原点，以背景图的像素大小为坐标系单位。

在世界坐标系中指定一个矩形区域，称为**世界窗口**，世界窗口用于表示场景中要显示的部分，也称为裁剪窗口，所有在此区域外的场景都将被裁去，只有在剪裁窗口内的场景才会显示在屏幕上。



1. 观察坐标系

根据需要观察的角度和位置确定建立二维笛卡尔直角坐标系，称为观察坐标系，观察坐标系通常与世界坐标窗口的位置和方向一致。

观察坐标系在要转换到视口中之前，需要对世界窗口外的场景进行裁剪。



1. 工作站设备坐标系

设备坐标系是二维笛卡尔直角坐标系，原点位于屏幕左上角，基本单位为像素。

设备坐标系是在其上进行绘制的物理设备（如屏幕或纸张）所使用的坐标系，我们这里以屏幕作为设备坐标系。显示器等图形输出设备自身都有一个坐标系统，屏幕上的位置使用与帧缓存中的像素位置相对应的整数屏幕坐标进行描述。

在设备坐标系上指定一个对应的矩形区域，称为**视口**。图形对象在世界窗口内的部分将被映射到显示窗口中指定位置的视口中，世界窗口选择要显示什么内容，而视口指定在输出设备的什么位置进行显示。如果窗口和视口高度宽度比不同，将会出现非均匀缩放变换，我们这里设定世界坐标窗口和视口比例相同，以保证显示效果。



1. 拼接屏设备坐标系

拼接屏设备坐标系是二维笛卡尔直角坐标系，原点位于左上角，基本单位为像素。

拼接屏设备总分辨率大小为7680\*3240，边界为矩阵中横坐标对应的垂直线段，纵坐标对应的水平线段。

视口宽度设为1600像素，高度设为675像素，与拼接屏设备坐标系分辨率的比为1:4.8。



1. 单元屏设备坐标系

拼接屏由12块单元屏组成，每一块单元屏都有自己的设备坐标系，显示边界为对应垂直线段，对应水平线段，为二维笛卡尔直角坐标系，原点位于左上角，基本单位为像素。

1. 二维几何变换与二维坐标系间变换
2. 基本二维几何变换
   1. 二维平移矩阵

使用齐次坐标方法，坐标位置的二维平移可表示为下面的矩阵乘法。



该运算可简写为



在平移参数没有被混淆的情况下，我们可以简单地使用来表示平移矩阵。

* 1. 二维旋转矩阵

类似的，围绕坐标系原点的二维旋转变换方程可以表示成矩阵形式：



可简写为



我们可以简单地把旋转矩阵写成。绕任意基准点的旋转要经过一系列的变换操作来完成。

* 1. 二维缩放矩阵

最后，相对于坐标原点的二维缩放变换可以表示为矩阵形式：



可简写为



多数情况下，我们可以将缩放矩阵写成。以任一参考点为中心的缩放变换通过一系列变换操作来处理。

1. 二维坐标系间变换

程序可设置参数：

世界窗口初始宽w

世界窗口初始高h

缩放因子s

旋转因子r

平移因子t

* 1. 二维观察变换

二维世界坐标系场景描述到设备坐标系的映射称为二维观察变换。二维观察变换的一般方法是在世界坐标系中指定一个观察坐标系统，以该系统为参考通过选定方向和位置来制定矩形剪裁窗口。

选择世界坐标系其中某点作为二维观察坐标系的原点。变换的第一步是将观察坐标系原点移动到与世界坐标系原点重合。接着，旋转观察坐标系使其与世界坐标系重合。

可以计算出由观察坐标系变换到与世界坐标系重合所需要的平移矩阵和旋转矩阵。即：



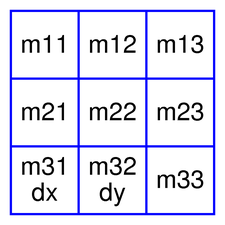
现在假设在世界坐标系中有个点，他在观察坐标系中的位置可由下式得出：





Qt中矩阵变换。

### Basic Matrix Operations



A [QTransform](qthelp://org.qt-project.qtgui.550/qtgui/qtransform.html) object contains a 3 x 3 matrix.

The m31 (dx) and m32 (dy) elements specify horizontal and vertical translation.

The m11 and m22 elements specify horizontal and vertical scaling.

The m21and m12 elements specify horizontal and vertical *shearing*.

And finally, the m13 and m23 elements specify horizontal and vertical projection, with m33 as an additional projection factor.

[QTransform](qthelp://org.qt-project.qtgui.550/qtgui/qtransform.html) transforms a point in the plane to another point using the following formulas:

x' = m11\*x + m21\*y + dx

y' = m22\*y + m12\*x + dy

if (is not affine) {

w' = m13\*x + m23\*y + m33

x' /= w'

y' /= w'

} 

The point *(x, y)* is the original point, and *(x', y')* is the transformed point. *(x', y')* can be transformed back to *(x, y)* by performing the same operation on the [inverted](qthelp://org.qt-project.qtgui.550/qtgui/qtransform.html#inverted)() matrix.

Translation is the simplest transformation. Setting dx and dy will move the coordinate system dx units along the X axis and dy units along the Y axis. Scaling can be done by setting m11 and m22. For example, setting m11 to 2 and m22 to 1.5 will double the height and increase the width by 50%. The identity matrix has m11, m22, and m33 set to 1 (all others are set to 0) mapping a point to itself. Shearing is controlled by m12 and m21. Setting these elements to values different from zero will twist the coordinate system. Rotation is achieved by setting both the shearing factors and the scaling factors. Perspective transformation is achieved by setting both the projection factors and the scaling factors.

无变换。

QTransform(type=TxNone,

11=1 12=0 13=0

21=0 22=1 23=0

31=0 32=0 33=1)

P(100,100)世界坐标系变为观察坐标系：QPoint(100,100)



放大S=1.05。

QTransform(type=TxScale,

11=1.05 12=0 13=0

21=0 22=1.05 23=0

31=0 32=0 33=1)

P(100,100)世界坐标系变为观察坐标系：QPoint(105,105)

[100 100 1] \* [1.05 0 0; 0 1.05 0;0 0 1] =[105;105;1]

顺时针旋转 =12°。

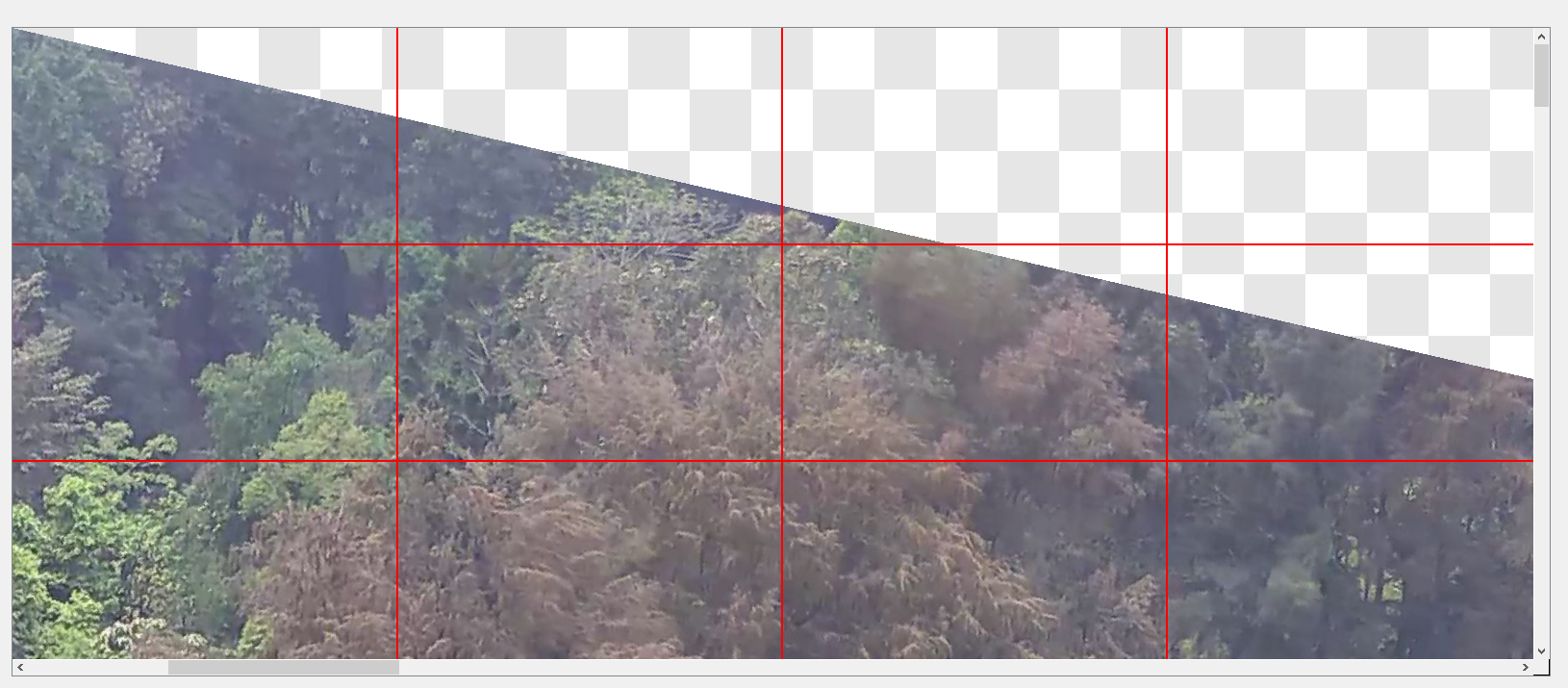
QTransform(type=TxRotate,

11=0.978148 12=0.207912 13=0

21=-0.207912 22=0.978148 23=0

31=0 32=0 33=1)

P(100,100)世界坐标系变为观察坐标系：QPoint(77,119)

[100 100 1] \* [0.978148 0.207912 0;-0.207912 0.978148 0;0 0 1] =[77.0236 118.6060 1.0000]

向右下平移244，191个单位。

QTransform(type=TxTranslate,

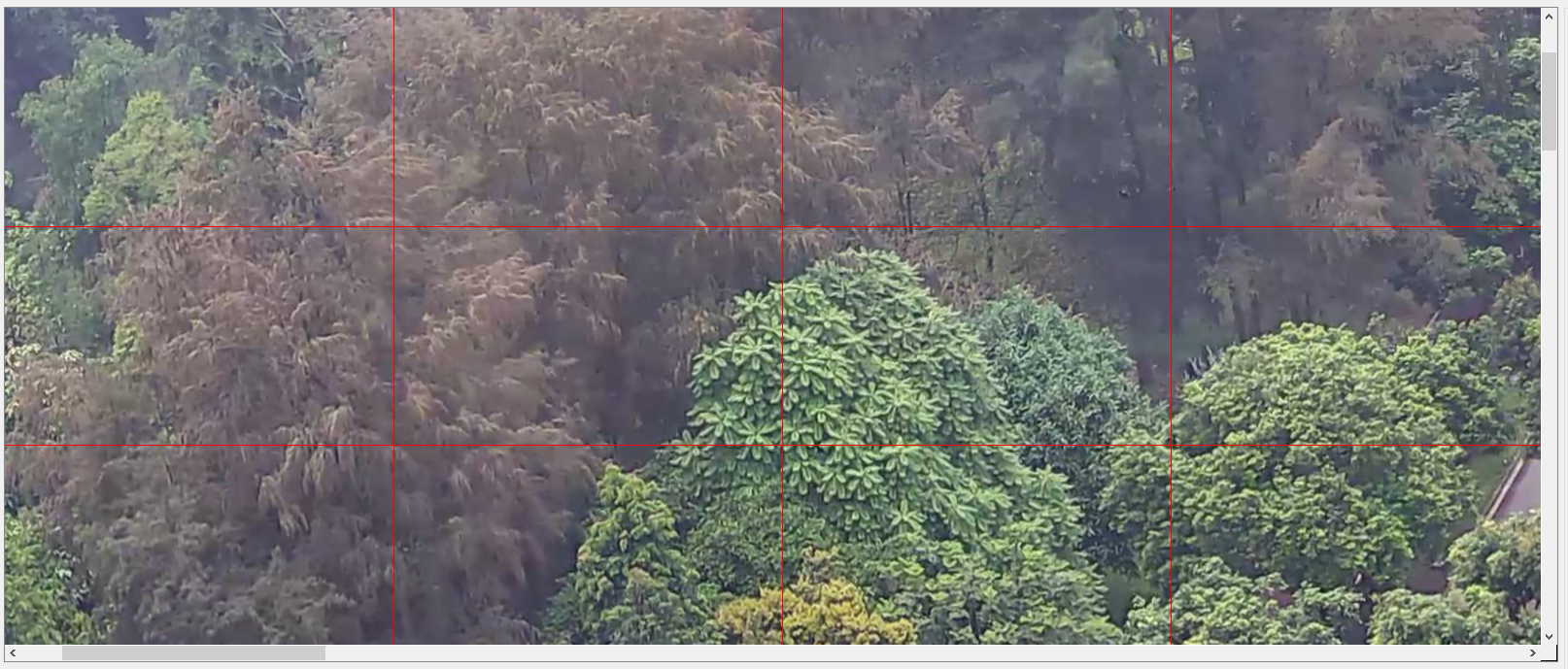
11=1 12=0 13=0

21=0 22=1 23=0

31=-244 32=-191 33=1)

P(100,100)世界坐标系变为观察坐标系：QPoint(-144,-91)

[100 100 1] \* [1 0 0;0 1 0;-244 -191 1] =[-144 -91 1]



复合变换，顺时针旋转45°，放大S=1.05。

顺时针旋转45。

QTransform(type=TxRotate,

11=0.707107 12=0.707107 13=0

21=-0.707107 22=0.707107 23=0

31=0 32=0 33=1)

P(100,100)世界坐标系变为观察坐标系：QPoint(0,141)

[100 100 1] \* [0.707107 0.707107 0;-0.707107 0.707107 0;0 0 1] =[0 141.4214 1.0000]

再放大1.05后。

QTransform(type=TxRotate,

11=0.742462 12=0.742462 13=0

21=-0.742462 22=0.742462 23=0

31=0 32=0 33=1)

R\*S:

[0.707107 0.707107 0;-0.707107 0.707107 0;0 0 1]\* [1.05 0 0; 0 1.05 0;0 0 1] =

0.7425 0.7425 0

-0.7425 0.7425 0

0 0 1.0000

P(100,100)世界坐标系变为观察坐标系：QPoint(0,148)

[100 100 1] \* [0.707107 0.707107 0;-0.707107 0.707107 0;0 0 1] =[0 148.4925 1.0000]

