**计算机图形学实验**

**姓 名：白文强**

**学 号：20191060064**

**专 业：计算机科学与技术**

**教 师：钱文华**

实验六 二维图形几何变换实验

时间：2020年11月20日

地点：信息学院2202机房

1、实验内容

使用opengl，对二维几何图形进行几何变换

2、实验目的

验证二维几何变换实验，平移、比例、旋转、错切、对称变换，其中三种。

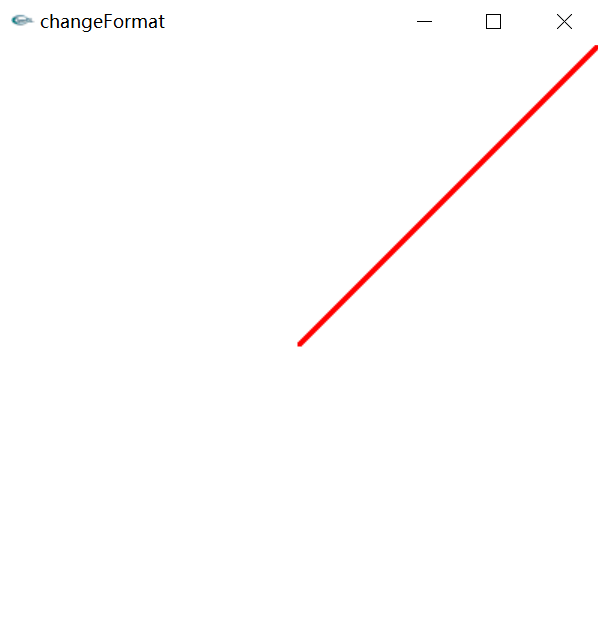
3、实验代码

1. #include <windows.h>
2. #include <GL/glut.h>
3. #include <stdlib.h>
4. #include <math.h>
5. #include <stdio.h>
6. #define PI 3.1415926535
8. **void** init(**void**) {
9. glClearColor(1.0, 1.0, 1.0, 1.0);
10. gluOrtho2D(-400, 400.0, -400, 400.0);
11. }
13. **void** translationTransformation(**int** \*x, **int** \* y, **int** xSpan, **int** ySpan)//平移变换
14. {
15. **int** transformat[3][3] = { 1, 0, 0,
16. 0, 1, 0,
17. xSpan, ySpan, 1
18. };
19. **int** Location[1][3] = {\*x, \*y, 1};
20. **int** nextLocation[1][3] = { 0 };
21. **for** (**int** i = 0; i < 1; i++)
22. {
23. **for** (**int** j = 0; j < 3; j++)
24. {
25. **for** (**int** k = 0; k < 3; k++)
26. {
27. nextLocation[i][j] += Location[i][k] \* transformat[k][j];
28. }
29. }
30. }
31. \*x = nextLocation[0][0] \* nextLocation[0][2];
32. \*y = nextLocation[0][1]\*nextLocation[0][2];
33. }
35. **void** scaleTransformation(**int** \*x, **int** \* y, **float** xScale, **float** yScale)//比例变换
36. {
37. **float** transformat[3][3] = { xScale, 0, 0,
38. 0, yScale, 0,
39. 0, 0, 1 };
40. **float** Location[1][3] = { \*x, \*y, 1 };
41. **float** nextLocation[1][3] = { 0 };
42. **for** (**int** i = 0; i < 1; i++)
43. {
44. **for** (**int** j = 0; j < 3; j++)
45. {
46. **for** (**int** k = 0; k < 3; k++)
47. {
48. nextLocation[i][j] += Location[i][k] \* transformat[k][j];
49. }
50. }
51. }
52. \*x = nextLocation[0][0] \* nextLocation[0][2];
53. \*y = nextLocation[0][1] \* nextLocation[0][2];
54. }
56. **void** rotateTransformation(**int** \*x, **int** \* y, **float** theta)//旋转变换
57. {
58. **float** transformat[3][3] = { cos(theta), sin(theta), 0,
59. -sin(theta), cos(theta), 0,
60. 0, 0, 1 };
61. **float** Location[1][3] = { \*x, \*y, 1 };
62. **float** nextLocation[1][3] = { 0 };
63. **for** (**int** i = 0; i < 1; i++)
64. {
65. **for** (**int** j = 0; j < 3; j++)
66. {
67. **for** (**int** k = 0; k < 3; k++)
68. {
69. nextLocation[i][j] += Location[i][k] \* transformat[k][j];
70. }
71. }
72. }
73. \*x = nextLocation[0][0] \* nextLocation[0][2];
74. \*y = nextLocation[0][1] \* nextLocation[0][2];
75. }
76. **void** setPixel(**int** x, **int** y)
77. {
78. glBegin(GL\_POINTS);
79. rotateTransformation(&x, &y, 0);
80. glVertex2i(x, y);
81. glEnd();
82. }

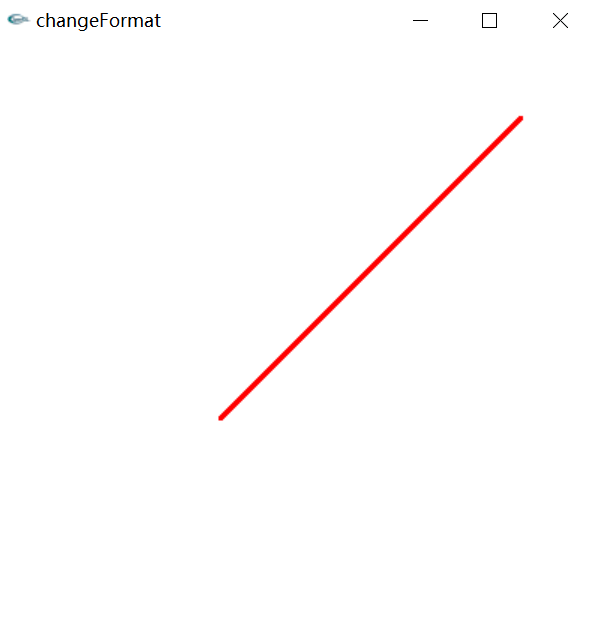
85. **void** lineBresenHam(**int** x0, **int** y0, **int** xEnd, **int** yEnd) {
86. **int** x, y;
87. **int** dx = xEnd - x0;
88. **int** dy = yEnd - y0;
89. //每次都从最左边的点画
90. **if** (x0 > xEnd) {
91. x = xEnd;
92. y = yEnd;
93. xEnd = x0;
94. yEnd = y0;
95. }
96. **else** {
97. x = x0;
98. y = y0;
99. }
100. **float** k = **float**(dy) / dx;       //斜率
101. **float** d = 0;
102. setPixel(x, y);
103. **if** (fabs(dx) >= fabs(dy)) {       //斜率绝对值大于零小于1
104. **int** steps = fabs(dx);
105. **for** (**int** i = 0; i < steps; i++) {
106. x++;
107. d += k;
108. **if** (k >= 0) {
109. **if** (d > 1) {
110. d = d - 1;
111. }
112. **if** (d > 0.5) {
113. y++;
114. }
115. }
116. **else** {
117. **if** (d < -1) {
118. d = d + 1;
119. }
120. **if** (d < -0.5) {
121. y--;
122. }
123. }
124. setPixel(x, y);
125. }
126. }
127. **else** {
128. **int** steps = fabs(dy);
129. **for** (**int** i = 0; i < steps; i++) {
130. d += 1 / k;
131. **if** (k >= 0) {
132. y++;
133. **if** (d > 1) {
134. d = d - 1;
135. }
136. **if** (d > 0.5) {
137. x++;
138. }
139. }
140. **else** {
141. y--;
142. **if** (d < -1) {
143. d = d + 1;
144. }
145. **if** (d < -0.5) {
146. x++;
147. }
148. }
149. setPixel(x, y);
150. }
151. }
152. }
153. **void** lineSegement(**void**) {
154. glClear(GL\_COLOR\_BUFFER\_BIT);   // 清空显示窗口
155. glColor3f(1.0, 0.0, 0.0);       // 指定前景色
156. glPointSize(3.0);               //点的大小
157. lineBresenHam(0, 0, 400, 400);
158. glFlush();
159. }
161. **int** main(**int** argc, **char**\*\* argv) {
162. glutInit(&argc, argv);
163. glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);
164. glutInitWindowPosition(100, 100);
165. glutInitWindowSize(400, 400);
166. glutCreateWindow("changeFormat");
167. init();
168. glutDisplayFunc(lineSegement);
169. glutMainLoop();
170. **return** 0;
171. }

4、实验结果

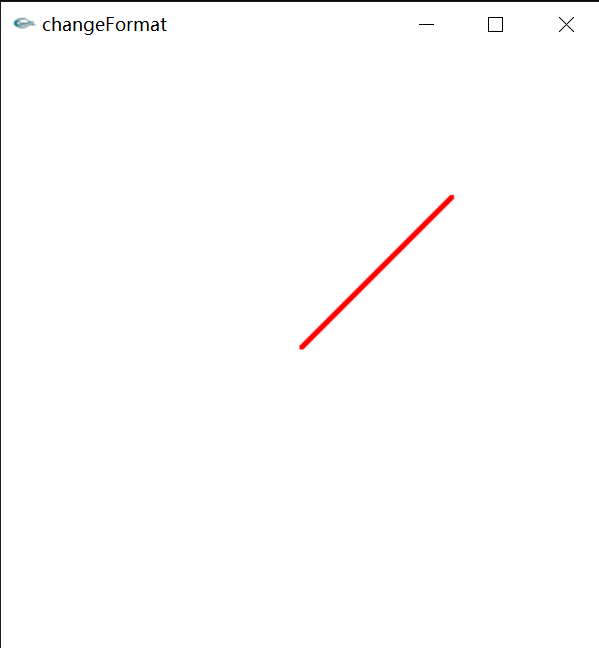
初始图形：



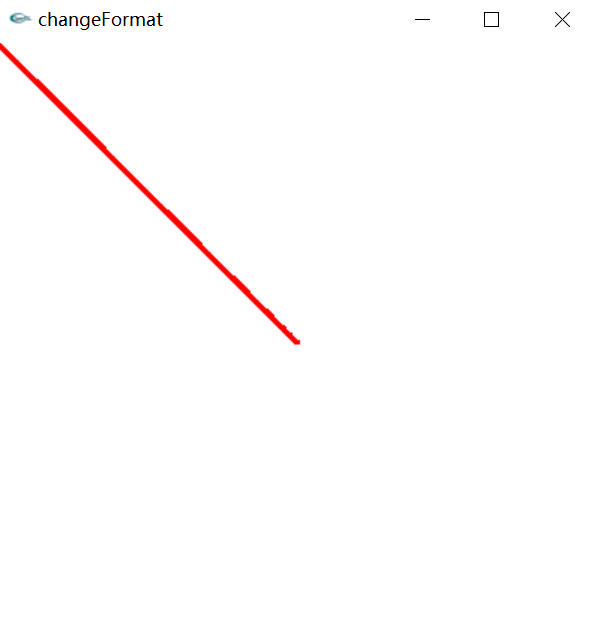
平移：



缩放：



旋转：



5、实验总结

　　通过齐次矩阵对单个点进行变换，这样，对所有的点都进行同样地变换，即可以达到对整个图形进行变换的目的