1.

（a）$O(wnd)$

当d变大的时候，计算复杂度上升

（b）因为能够cut-off的条件是当且仅当对于query vector q, $\delta_i \geq \triangle_a$（$i(x) = x_k$），显然当d增大的时候，这种情况的发生率会越来越低。所以说对于计算复杂度的降低也是越来越可以忽略不计。

（c）考虑（$((x-k)/x)^d$，底数小于1，当指数越来越大的时候，越来越小。

也就是说中心部分体积越来越小，边缘部分体积越来越大。

（d） 不能，因为当维度升高的时候，大部分点都会被直接丢弃掉（根据（c）中的结论).

（e）考虑集中降维的方法，比如PCA或者LDA

2.

(a) $[x1-y1\ x2-y2\ \ldots\ xn-yn]\ [x1-y1\ x2-y2\ \ldots\ xn-yn]^T = \Sigma(x_i-y_i)^2$

(b) $a_{i,j} = (-x_{0,j}+x_{i,j})$ ($i = 1\ldots K$, $j = 1\ldots N$)

$b_i = \Sigma_{j=1..N}(-x_{0,j}^2 + x_{i,j}^2)$

(c)可以在V空间内任选一点作为x0，然后对于构成V的k条直线分别构建x0的镜像点，构成x1-xk

(d)cos相等=>$\Sigma(x_{i,j}*x_j) = \Sigma(x_{0,j}*x_j)$

欧氏距离相等=> $\Sigma(x_{i,j}-x_j)^2 = \Sigma(x_{0,j}-x_j)^2 \Rightarrow \Sigma(x_{i,j}^2+ x_{i,j}*x_j) = \Sigma(x_{0,j}^2+x_{0,j}*x_j)$

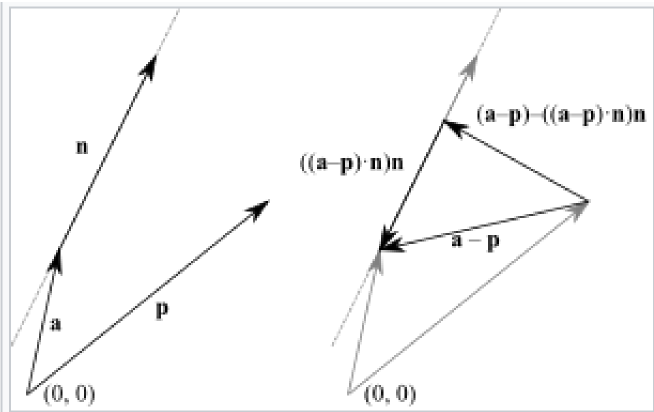
因为x0 xK are unit vectors. 所以说是欧式距离相等和cos相等是等价的。

(e)没有区别，正如（d）中所证明的一样

(f)

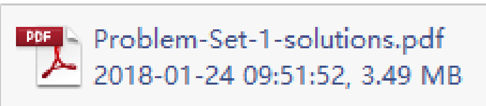
3.

（a）



b-x对应a-p $((x-b)\cdot a)a$是在直线上的投影，垂直部分是$(x-b)-((x-b)\cdot a)a$，经过划归可以得到题目当中的结果。

(b)(c)



(d)首先根据直线直线正交的关系，可以确定a_other，然后再用类似的方法在a确定的情况下，考虑使得所有点的距离之和最短，求得b_other

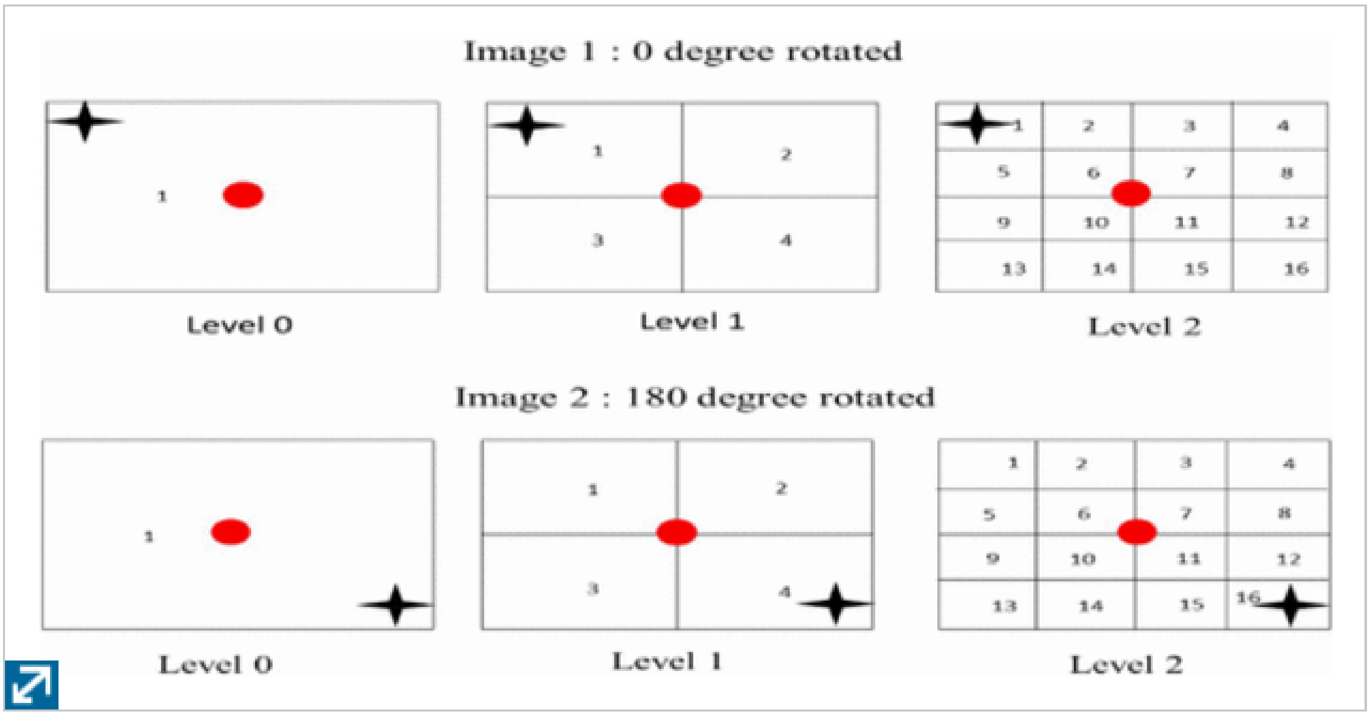
4.

(a)


Starbucks: An Illustrated History
STARBUCKS
COFFEE
STARBUCKS
COFFEE
2011


Starbucks: An Illustrated History
STARBUCKS
COFFEE
STARBUCKS
COFFEE
2011

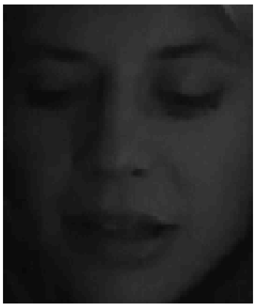
(b)
(c)

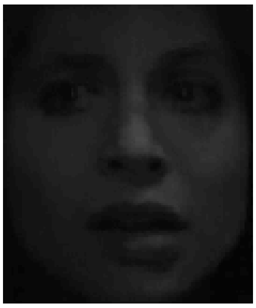


都不能保证，对于SPM，如上图所示，如果旋转180度，会落到完全不同的区域内

5.

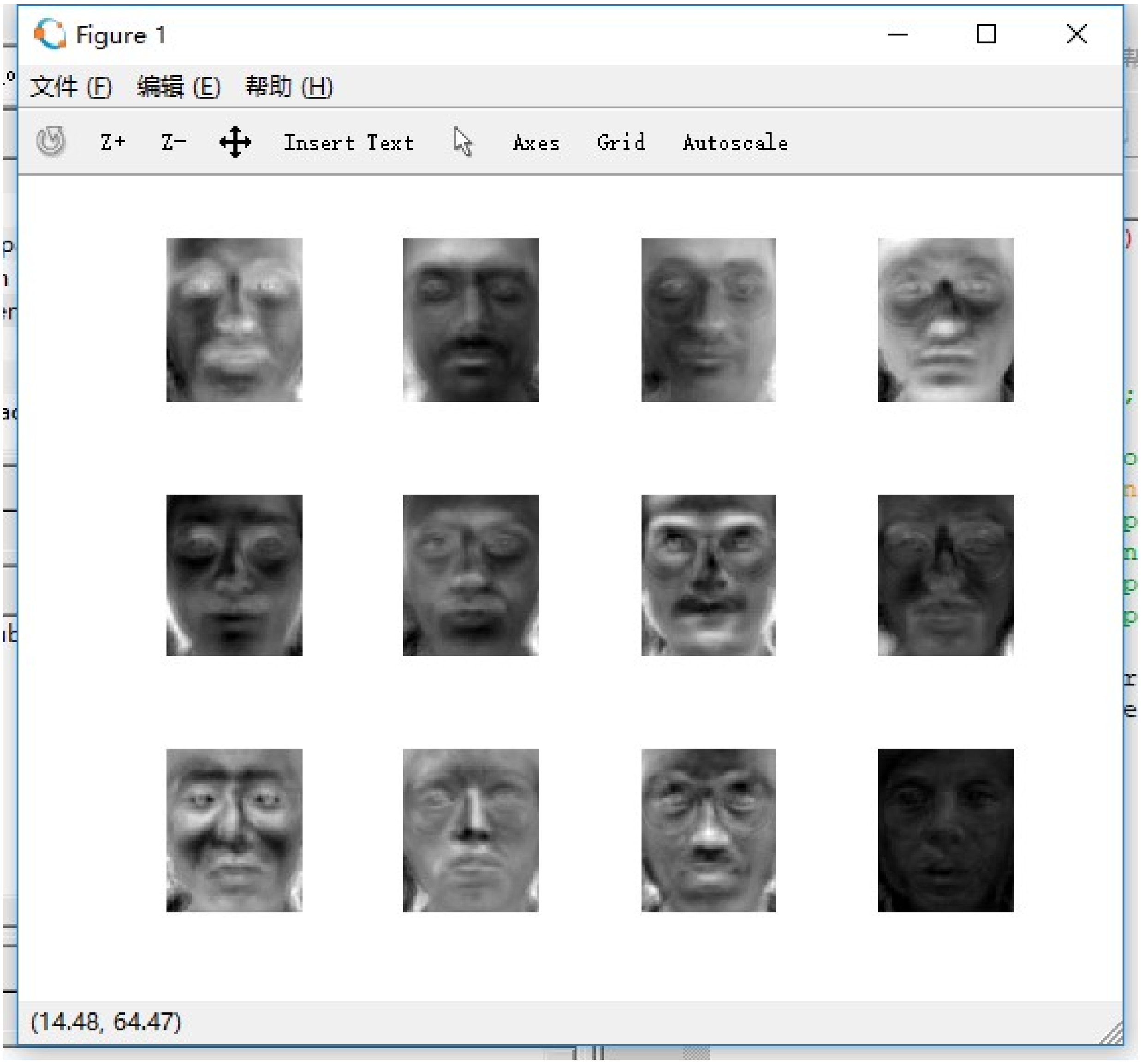
5.1





5.2

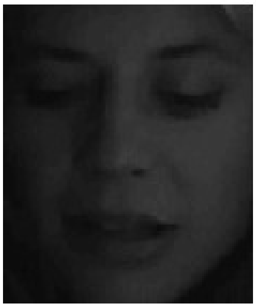
前12个主成分

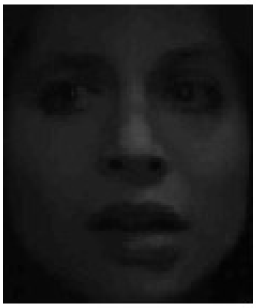


测试数据的主成分

相似图和原始测试图

（b）





5.3

（a）失败 用Sarah Connor, higher brightness.

（b）

用fisher方法，成功识别

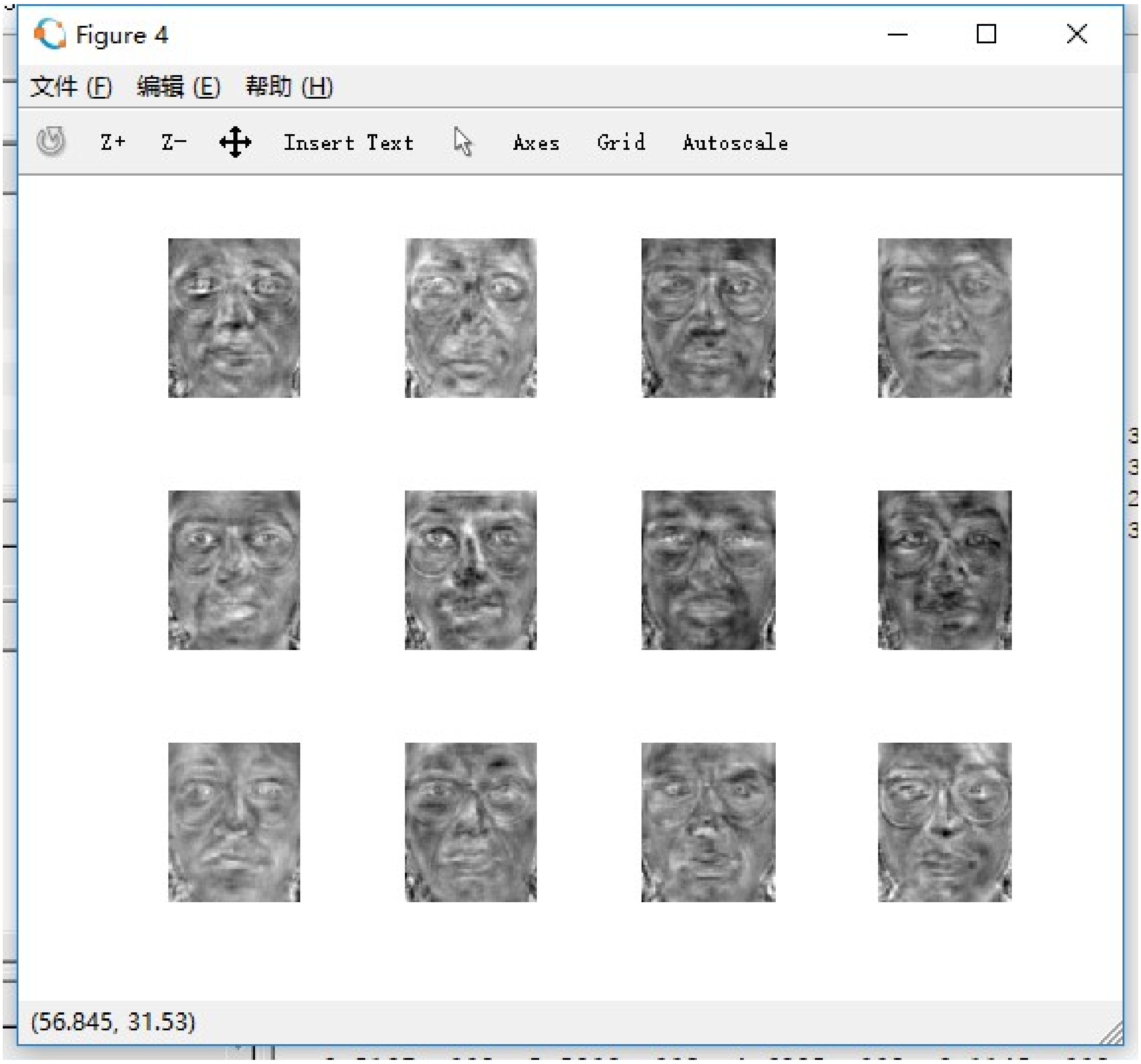


Figure 4
文件 (F)
编辑 (E)
帮助 (H)
Z+
Z-
Insert Text
Axes
Grid
Autoscale
(56.845, 31.53)

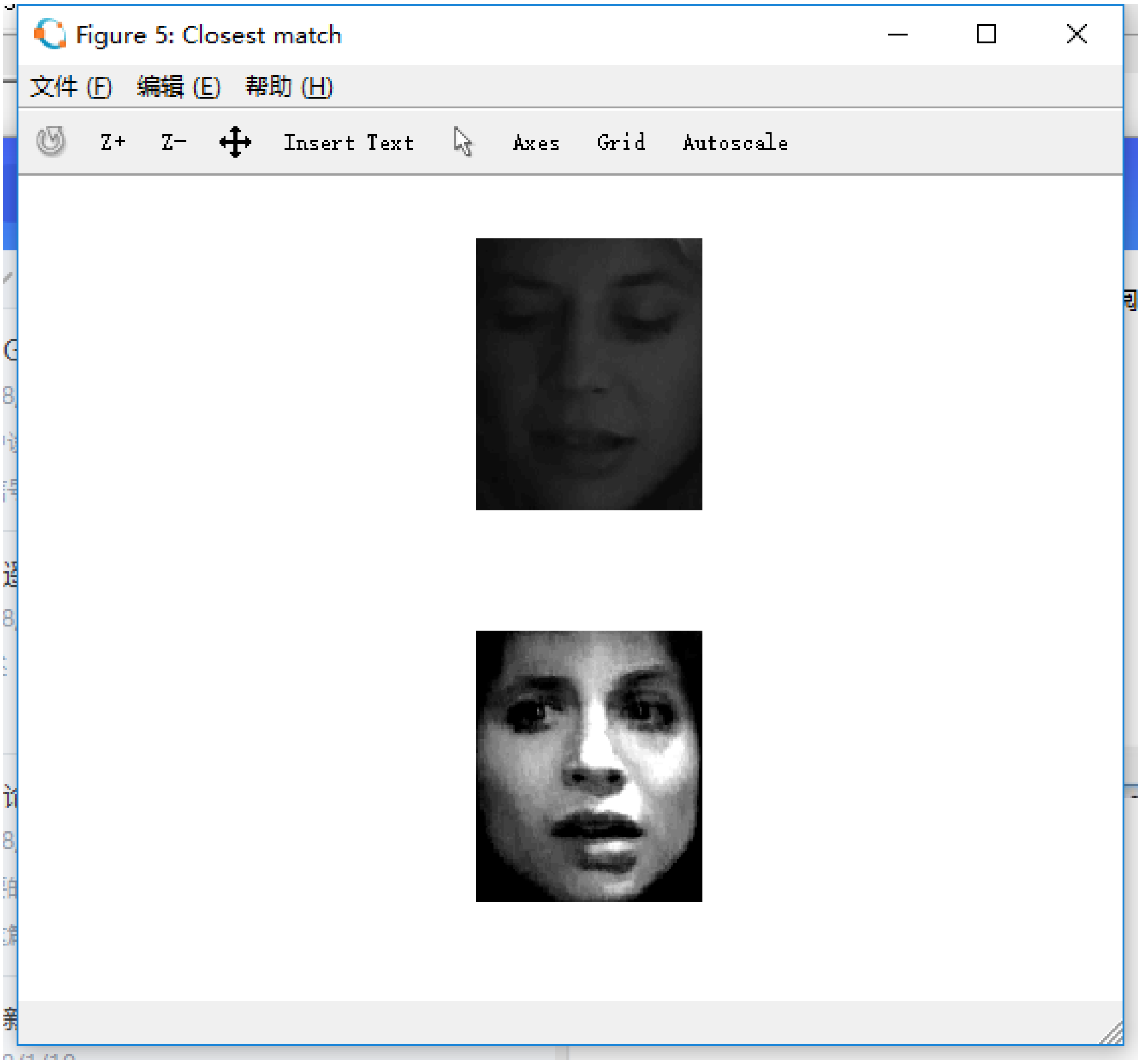


Figure 5: Closest match
文件 (F) 编辑 (E) 帮助 (H)
Z+ Z- Insert Text Axes Grid Autoscale