

Nom i Cognoms: \_\_\_\_\_ **Una Possible Solució** \_\_\_\_\_

1) Què és el *SKIZ* (*Skeleton by Influence Zone*) d'una imatge binària?

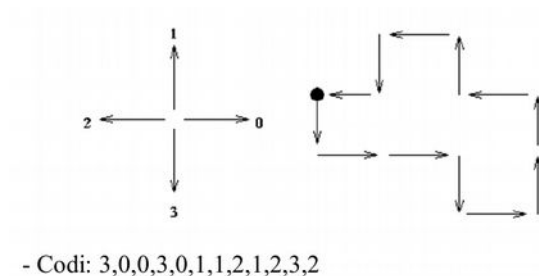
És el conjunt de píxels del fons que equidisten de dues (o més) formes

Com l'implementaries?

Mitjançant un watershed sobre la transformada de distància

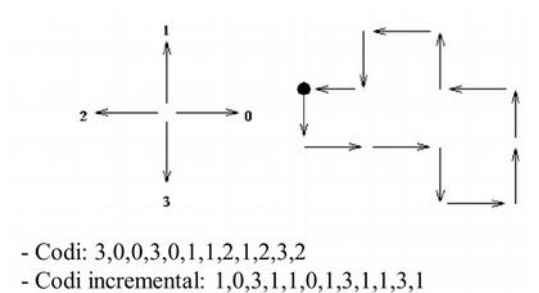
2) Com es pot descriure el contorn d'una figura usant codis de cadena?

Assignant un codi numèric a cada segment del contorn en funció de la seva orientació



Com faries que els codis de cadena fossin invariants a la rotació?

Fent una codificació incremental en funció de la variació de la orientació



3) Cita els inconvenients de reconèixer objectes mitjançant *template matching*

No és invariant a la rotació, ni al canvi d'escala, ni a canvis d'il.luminació, ni de perspectiva, ni a oclusions.

Probar tots els templates resultants de totes les possibles transformacions és inviable.

4) Avantatges i inconvenients dels histogrames de color enfront dels *HoGs* per a descriure un objecte.

Els histogrames de color funcionen bé si les imatges tenen constància de color. Són molt sensibles a la il.luminació.

Els histogrames d'orientació són més robustos a canvis d'il.luminació, però no funcionaran bé si l'objecte pateix deformacions.

Que és més convenient: dimensionar els histogrames d'orientació entre  $-180^\circ$  i  $180^\circ$ , o entre  $0^\circ$  i  $180^\circ$  ?. Proposa dos exemples que mostrin quan convé fer-ho d'una forma o de l'altra.

Depèn de l'aplicació.

Un gradient amb orientació  $\alpha$ , i un gradient amb orientació  $\alpha+180^\circ$ , tenen la mateixa direcció. Un representa una transició clar $\rightarrow$ fosc, mentre que l'altre representa una transició fosc $\rightarrow$ clar.

Per tant, si desconexem la lluminositat de l'objecte respecte del fons, és absurd treballar amb signe, doncs el mateix objecte es descriuria diferent sobre un fons clar que sobre un fons fosc.

5) Per a calcular una certa matriu de transformació  $T$ , volem usar l'algorisme RANSAC. Calculem el número d'iteracions necessàries usant la fórmula:

$$N = \log(1 - p) / \log(1 - (1 - e)^f)$$

I obtenim que calen  $N = 10000$  iteracions per a trobar la matriu  $T$  amb els paràmetres corresponents.

Malgrat això, executem RANSAC i l'algorisme s'atura a la iteració  $N=37$ . Com s'explica això?

Perquè l'algorisme RANSAC recalcula  $N$  en cada iteració en funció del  $n^\circ$  d'inliers detectats.

Quina seguretat tenim de que la  $T$  trobada sigui la matriu de transformació correcta?

$T$  és la matriu de transformació correcta amb una probabilitat  $p$ .

---