**109590004 呂育瑋** Machine Vision HW#3

這次作業實作以Quad-tree資料結構做影像處理，對圖像做灰階化與二值化後以不同的Quad-tree Layer輸出影像。

步驟如下：

對圖像做灰階化後，依題目要求的門檻值對影像作二值化，計算圖像最大Layer數為log2(圖像邊長)，由Layer 1 到最大Layer數依序輸出圖像。

vector<uchar> getBinaryImageValueList(const Point2i& pointBegin, const Point2i& pointEnd)

依照選取的區塊取得二值化圖像中的值，並回傳這些值的數列，用作區塊單色、非單色判斷依據。

uchar getMergeColor(const vector<uchar> pixels)

由 getBinaryImageValueList() 回傳的數列，將數列中的值加總後的平均值，判斷應該回傳0, 128, 255其中之一，如果加總平均值為0或255，回傳加總平均值(即0或者255)；否則回傳128(表示區塊內非單色組成)。

void updateQuadTreeImage(const Point2i& pointBegin, const Point2i& pointEnd, const uchar& color)

依照指定的區塊為圖像填上 color 這個顏色，color由 getMergeColor() 決定。

void updateQuadTreeRecursively(int treeHeight, const Point2i pointBegin, const Point2i pointEnd)

以遞迴函式建構Quad-tree，treeHeight 作為 Layer 的判斷依據；pointBegin 與 pointEnd 為圖像起始與終點座標，在

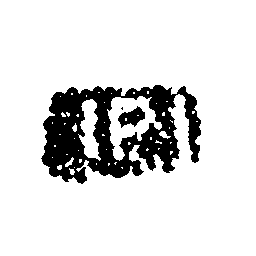
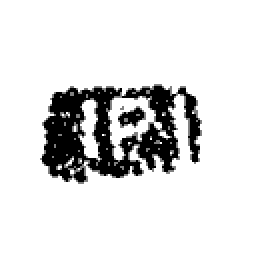
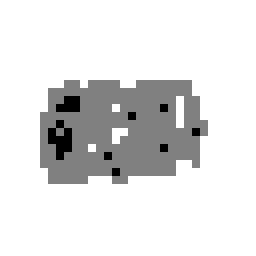
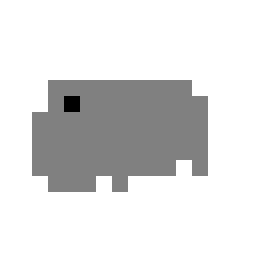
每次遞迴先執行 getBinaryImageValueList() 並使用 getMergeColor() 取得區塊應被填滿的顏色mergeColor：

如果mergeColor為灰色且treeHeight 不為 0，將treeHeight – 1、區塊切割成四等分繼續遞迴。

如果 mergeColor為單色，或者treeHeight = 0代表Quad-tree的深度以達到目標Layer數，呼叫 updateQuadTreeImage() 將該區塊塗上mergeColor的顏色，並且結束遞迴。

void createQuadTreeImage(int layer)

呼叫updateQuadTreeRecursively()遞迴，treeHeight為Layer數、pointBegin = (0, 0)、pointEnd = (圖長, 圖寬)，結束所有遞迴後就能取得指定Layer數的Quad-tree image。



Layer\_1

Layer\_2

Layer\_3

Layer\_4

Layer\_5

Layer\_6

Layer\_7

Layer\_8



Layer\_1

Layer\_2

Layer\_3

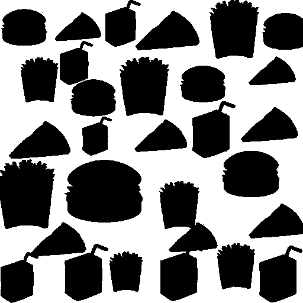
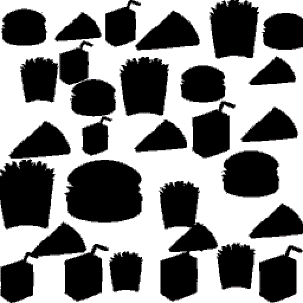
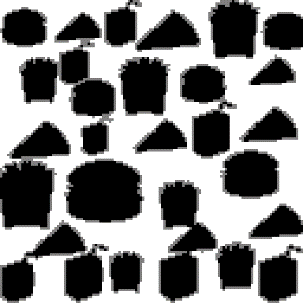
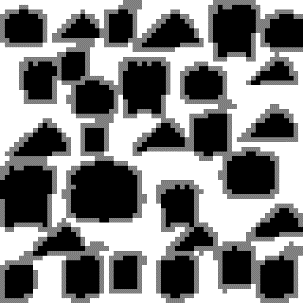
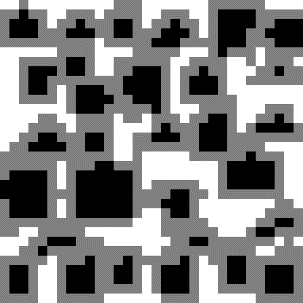
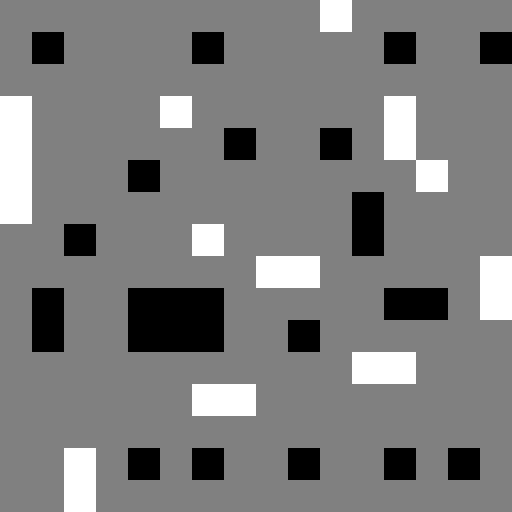
Layer\_4

Layer\_5

Layer\_6

Layer\_7

Layer\_8



Layer\_9

Layer\_1

Layer\_2

Layer\_3

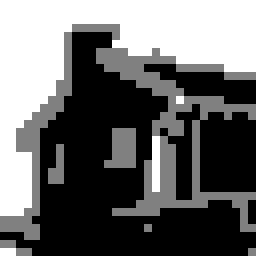
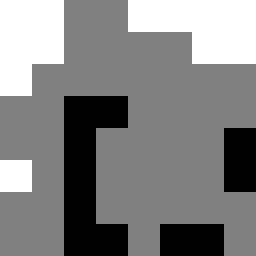
Layer\_4

Layer\_5

Layer\_6

Layer\_7

Layer\_8



Layer\_1

Layer\_2

Layer\_3

Layer\_4

Layer\_5

Layer\_6

Layer\_7

Layer\_8

Layer\_9

