

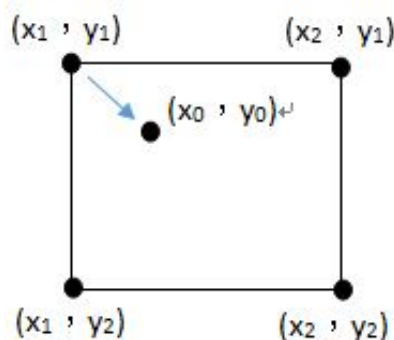
阿洲的程式教學

關於Qt、OpenCV、影像處理演算法

線性內插(Interpolation)

當我們圖像進行幾何轉換時，假使輸出像素映射的地方，不是輸入圖像某個整數像素位置，這時要用整數座標的灰度值進行推斷，這就是插值，這邊介紹幾種插值方式，通常較好的結果也導致較大的計算量。

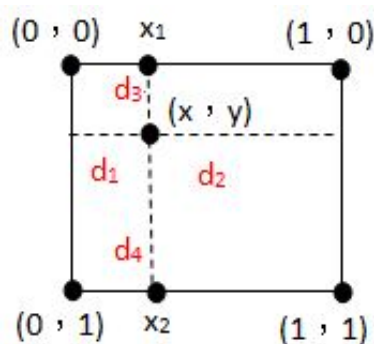
最近插值法(Nearest Neighbor Interpolation)：這是一種最簡單的插值算法，輸出像素的值為輸入圖像離映射點最近的像素值，如下圖假使 (x_0, y_0) 為映射點，則讓此點的強度值為 (x_1, y_1) 的值，這種算法作幾何轉換時，邊緣通常有較嚴重的鋸齒狀。



雙線性插值法(Bilinear Interpolation)：在兩個方向分別進行一次線性插值，輸出像素的值為映射點四周的 2×2 像素強度加權平均，如下圖我們簡化問題，四周位置分別為 $(0, 0)$ 、 $(1, 0)$ 、 $(0, 1)$ 、 $(1, 1)$ ，強度分別為 $f(0, 0)$ 、 $f(1, 0)$ 、 $f(0, 1)$ 、 $f(1, 1)$ ， (x, y) 映射點強度為 $f(x, y)$ ，映射點到四邊的距離分別為 d_1 、 d_2 、 d_3 、 d_4 、我們依序進行以下三步驟：

1. 對上端的兩個頂點進行線性插值得到 x_1 的強度 $f(x_1)$ 。
$$f(x_1) = f(0, 0) + d_1 * (f(1, 0) - f(0, 0))$$
2. 對下端的兩個頂點進行線性插值得到 x_2 的強度 $f(x_2)$ 。
$$f(x_2) = f(0, 1) + d_1 * (f(1, 1) - f(0, 1))$$
3. 由 $f(x_1)$ 和 $f(x_2)$ 來求得 (x, y) 的強度 $f(x, y)$ 。
$$f(x, y) = f(x_1) + d_3 * (f(x_2) - f(x_1))$$

我們擴展這個概念，就可得到圖像所有的雙線性插值強度，這概念假設強度在兩個像素之間是線性變化的，顯然是合理的假設，因此在一般的情況之下，雙線性插值都能得到不錯的結果。



高階插值： 在一些幾何運算中，雙線性插值的平滑作用會導致細節的退化，這些可以透過高階插值彌補，可能會採用 4×4 或 8×8 的鄰域進行加權平均。

[回到首頁](#)

[回到OpenCV教學](#)

📅 2015-11-25 👤 阿宅 📁 OpenCV, 結構與基本概念 💡 Bilinear Interpolation, Interpolation, Nearest Neighbor Interpolation, 最近插值法, 線性插值, 雙線性插值法

0 Comments

猴子遇到0與1! 程式學習筆記

 Login ▾ Recommend Share

Sort by Best ▾



Start the discussion...

Be the first to comment.

ALSO ON 猴子遇到0與1! 程式學習筆記

Qt主窗口(Top Level Window)

1 comment • 6 months ago

mike — 喔喔

文件對話框(QFileDialog)

1 comment • 6 months ago

楊政穎 — dialog.cpp 裡面的 QString s
=
QFileDialog::getOpenFileName(this, tr

 Subscribe Add Disqus to your site Add Disqus Add Privacy

自豪的採用 WordPress