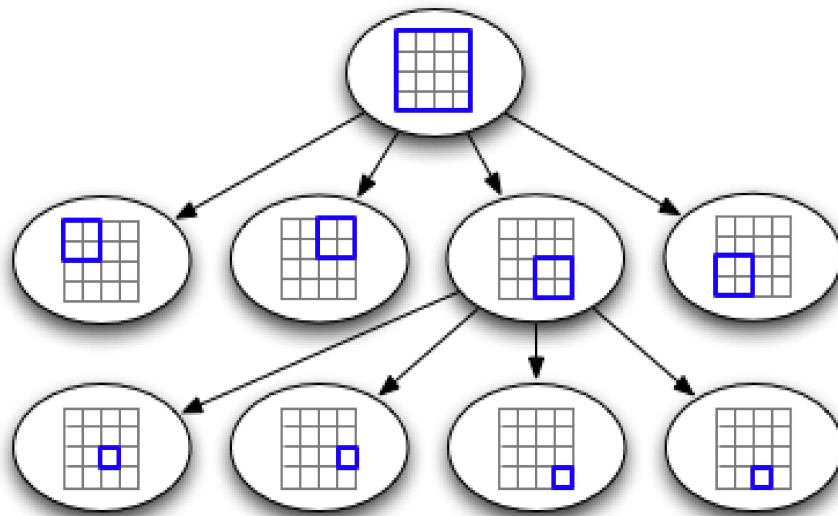


QUADTREE

Definisi

Quadtree adalah sebuah struktur data pohon yang akarnya memiliki persis empat buah simpul anak. *Quadtree* digunakan untuk merepresentasikan informasi spasial yang terdapat pada sebuah daerah tertentu. Biasanya struktur data ini digunakan untuk menyimpan informasi sebuah data dua dimensi, contohnya seperti pada informasi pewarnaan pada sebuah gambar.



Contoh *Quadtree*

Quadtree komplit yaitu pohon yang setiap simpulnya memiliki empat buah anak(kecuali simpul daun) dan semua simpul daunnya berada pada level yang sama. *Quadtree* ini disebut pohon piramid atau *Tree-Pyramid*.

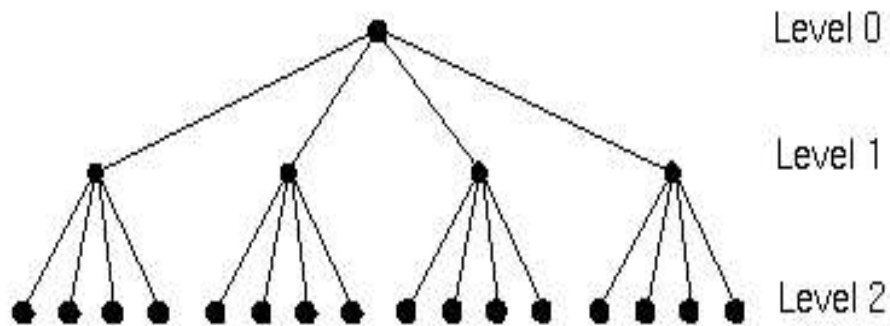


Figure 3.5 T-pyramid.

Jenis-Jenis *Quadtree*

- Region Quadtree** : *Quadtree* yang merepresentasikan partisi dari sebuah daerah dua dimensi menjadi empat subkuadran dengan simpul daun menyimpan informasi dari sebuah subkuadran tertentu. Simpul dalam pohon tersebut mempunyai empat buah simpul anak atau tidak sama sekali(simpul daun). Dalam merepresentasikan sebuah gambar, Region Quadtree dapat digunakan, yaitu dengan membagi daerah setiap gambar secara rekursif sampai pada level tertentu(biasanya sampai satu *pixel*) dan menyimpan informasi warna setiap *pixel*nya. Region Quadtree juga dapat digunakan untuk merepresentasikan nilai variabel pada sebuah kumpulan data, contohnya penyimpanan temperatur rata-rata pada sebuah daerah.



Region Quadtree untuk penyimpanan gambar

- *Point Quadtree* : *Quadtree* untuk merepresentasikan data dua dimensi, contohnya kumpulan titik-titik yang ingin disimpan. *Quadtree* ini sangat efisien apabila digunakan untuk membandingkan data yang terurut, biasanya kompleksitas waktunya adalah $O(\log n)$.
- *Point-Region Quadtree* : *Quadtree* yang sama seperti *Region Quadtree*, hanya saja informasi yang disimpan pada setiap simpul adalah kumpulan titik-titik yang berada pada daerah tertentu(yang direpresentasikan oleh simpul tersebut).
- *Edge Quadtree* : *Quadtree* untuk merepresentasikan kumpulan garis. Garis yang mempunyai kelengkungan dibagi menjadi beberapa garis hingga menjadi garis lurus atau mencapai level tertentu. Cara ini dapat menciptakan pohon yang sangat tidak seimbang sehingga mempersulit pencarian simpul tertentu.
- *Compressed Quadtree* : *Quadtree* ini adalah sebuah pohon yang direduksi agar tidak menyimpan informasi yang tidak dibutuhkan. Contohnya seperti penggunaan *Region Quadtree* untuk merepresentasikan gambar. Dengan menggunakan *Compressed Quadtree*, sebuah daerah pada gambar yang memiliki warna yang sama dapat direpresentasikan oleh sebuah simpul, tanpa harus membagi simpul yang merepresentasikan daerah tersebut hingga level satuan *pixel*. Penyimpanan data menjadi lebih efisien dan pencarian warna pada *pixel* tertentu menjadi lebih mudah.

Implementasi

```

/*
    Berikut adalah implementasi sederhana dari struktur data Region Quadtree
    menggunakan bahasa Java
*/

// Kelas Point sebagai penyimpan informasi kordinat dua dimensi
class Point{
    // Absis
    int x;
    // Ordinat
    int y;
}

// Kelas QuadNode sebagai penyimpan lokasi titik pusat simpul dan informasinya
class QuadNode{
    // Informasi yang disimpan oleh simpul
    int info;
    // Titik pusat simpul
    Point center;
}

// Kelas QuadTree sebagai representasi pohon berakar dan bercabang empat

```

```

class QuadTree{
    // Simpul akar dari pohon
    QuadNode root;
    // Titik yang menandai batas kiri atas dari daerah simpul
    Point topLeftBound;
    // Titik yang menandai batas kanan bawah dari daerah simpul
    Point bottomRightBound;
    // Cabang pertama sebagai representasi dari daerah kiri atas
    QuadTree topLeft;
    // Cabang kedua sebagai representasi dari daerah kanan atas
    QuadTree topRight;
    // Cabang ketiga sebagai representasi dari daerah kiri bawah
    QuadTree bottomLeft;
    // Cabang keempat sebagai representasi dari daerah kanan bawah
    QuadTree bottomRight;
}

```

Selain *getter* dan *setter*, struktur data di atas memiliki fungsi primitif seperti :

- void insert(QuadNode Q) : Memasukkan sebuah simpul ke dalam *Quadtree*.
- int search(Point P) : Mencari sebuah titik pada simpul dengan daerah terkecil yang memiliki titik tersebut dalam *Quadtree*, dan mengembalikan informasi yang ada pada simpul yang ditemukan(jika ada).

Kompresi Gambar Dengan Quadtree

Tujuan dari algoritma kompresi ini adalah memecah suatu gambar menjadi upa-gambar dan disimpan kesimpul dari Quadtree yang tiap upa-gambar-nya memiliki suatu kesamaan yang telah ditetapkan menggunakan algoritma Divide and Conquer.

Berikut algoritma kompresi:

1. Diberikan sebuah gambar x_0 .
2. Bagi gambar x_0 menjadi 4 upa-gambar x_1 , x_2 , x_3 , dan x_4 .
3. Untuk setiap upa-gambar x_i , hitung nilai error E_i dari upa-gambar tersebut.
4. Jika nilai error E_i lebih besar dari batas maksimum Error Limit, bagi upa-gambar menjadi 4 upa-gambar yang lebih kecil.
5. Jika nilai error E_i kurang dari sama dengan maksimum error Limit, simpan rata-rata nilai RGB upa-gambar tersebut.
6. Ulangi dari langkah ke-3 hingga semua nilai error E_i upa-gambar sudah dalam batas maksimum error.

Semakin besar batas nilai error yang digunakan, semakin tidak sesuai gambar hasil kompresi dengan gambar aslinya. Jika nilai error lebih dari 0, maka disebut lossy compression.

Lossy compression adalah jenis kompresi yang dimana selama proses kompresi ada data yang sengaja dihilangkan.

Perhitungan nilai error dapat dilakukan dengan banyak cara.

Referensi / Sumber

- <https://en.wikipedia.org/wiki/Quadtree>
- <https://www.geeksforgeeks.org/quad-tree/>
- <http://codeforces.com/blog/entry/57498>
- <https://blog.jverkamp.com/2014/05/28/quadtree-image-compression/>
- <https://arxiv.org/ftp/arxiv/papers/1502/1502.00324.pdf>

Pembagian Tugas:

- Jose Hosea (13516027) : Penjelasan Quadtree dan Implementasi
- Tony (13516010) : Algoritma Quadtree untuk kompresi gambar