|  |
| --- |
| **HANDOUT PRAKTIKUM ISYARAT DAN SISTEM 2**  **TOPIK 3 – PENCUPLIKAN DAN REKONSTRUKSI CITRA** |

1. **Peralatan/Komponen**
   1. Hardware: 1 set PC
   2. Software: MATLAB dan alat bantu komputasi lainnya.
2. **Tujuan**
   1. Mahasiswa mampu melakukan pencuplikan pada citra digital.
   2. Mahasiswa mampu merekonstruksi citra dari sampel citra.
3. **Dasar Teori**

Pada topik ini, kita akan belajar tentang penerapan dari teori pencuplikan (pada topik sebelumnya) pada *image processing*. Sebagai contoh, kita akan mencuplik citra cameraman.tif. Berikut ini adalah gambar citra yang dimaksud.

|  |  |
| --- | --- |
|  | input = imread('cameraman.tif');  imshow(input); |
|  |  |

Citra tersebut adalah citra 8 bit dengan ukuran 256 × 256. Untuk memeriksa informasi tersebut, anda lihat pada bagian *workspace*. Perhatikan bahwa untuk variabel input, value-nya adalah 256 × 256 uint8 (maksudnya tipe data-nya berupa *unsigned* *integer* 8 bit dengan ukuran 256 × 256).

Katakanlah kita akan mencuplik citra tersebut sehingga ukuran-nya menjadi setengah dari ukuran asli (atau dengan kata lain citra 256 × 256 akan kita cuplik menjadi citra 128 × 128).

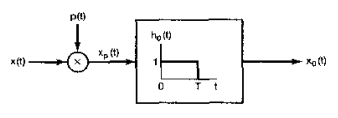
|  |  |
| --- | --- |
|  | n = 2;  output = transpose(downsample(transpose(downsample(input, n)), n));  imshow(output); |
|  |  |

Perhatikan bahwa ukuran citra menjadi setengah dari ukuran asli. Cek kembali *workspace* anda dan amati bahwa ukurannya! Sekarang, kita akan merekonstruksi citra dengan menggunakan citra sampel tadi.

|  |  |
| --- | --- |
|  | output = transpose(upsample(transpose(upsample(output, n)), n));  imshow(output); |
|  |  |

Ternyata citra rekonstruksi tidak menyerupai dengan citra yang asli. Perhatikan ada garis-garis hitam yang membuat citra menjadi kotak-kotak. Hal ini tentu karena ada ‘*blank* *pixel*’ yang tidak ada informasinya. Karena saya hanya menggunakan *upsample* standar, maka *blank* *pixel* tadi akan diisi nilai 0.

Adapun cara lain untuk merekonstruksi citra adalah dengan menggunakan *zero* *orde* *hold*. Itu berarti kita akan ‘menahan’ nilai *pixel* sampel tersebut hingga waktu pencuplikan berikutnya.



Gambar 1. Diagram alir rekonstruksi *zero orde hold*.

Mari kita coba teknik tersebut pada citra sampel sebelumnya.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | zoh = ones(2,2);  rekon = conv2(double(zoh), double(output));  imshow(uint8(rekon)); |  |  |  |
|  |  |  |  |  |

Terlihat bahwa hasilnya lebih baik dari pada citra rekonstruksi tanpa *zero orde hold*.