

# Kompresi Video Menggunakan Discrete Cosine Transform

Hananto Edy Wibowo<sup>1</sup>, Indra Sakti Wijayanto<sup>2</sup>, Nugroho Herucahyono<sup>3</sup>

Laboratorium Ilmu dan Rekayasa Komputasi  
Departemen Teknik Informatika, Institut Teknologi Bandung  
Jl. Ganesha 10, Bandung

E-mail : [if14002@students.if.itb.ac.id](mailto:if14002@students.if.itb.ac.id)<sup>1</sup>, [if14029@students.if.itb.ac.id](mailto:if14029@students.if.itb.ac.id)<sup>2</sup>,  
[if14038@students.if.itb.ac.id](mailto:if14038@students.if.itb.ac.id)<sup>3</sup>

## Abstrak

Dewasa ini, seiring perkembangan zaman, kebutuhan akan data video digital semakin meningkat, baik itu dalam lingkungan perusahaan, industri hiburan, layanan telekomunikasi maupun di rumah – rumah. Hal ini menjadikan teknologi video digital menjadi suatu kebutuhan yang harus dipenuhi. Permasalahan terbesar yang dihadapi adalah besarnya ukuran file video ini. Untuk mengatasi masalah ini, telah dicari berbagai macam cara agar dapat melakukan kompresi terhadap file video. Hasil yang diharapkan adalah ukuran file yang sekecil – kecilnya dengan kualitas yang cukup baik. Salah satu metoda yang banyak digunakan dalam melakukan kompresi file video adalah *Discrete Cosine Transform* (DCT). Pada makalah ini, Penulis mencoba membahas tentang kompresi file video menggunakan metoda *Discrete Cosine Transform* ini.

**Kata kunci:** *Discrete Cosine Transform, DCT, kompresi, video, decoding, lossy compression, lossless compression*

## 1. Pendahuluan

Kompresi video adalah bentuk kompresi data yang berhubungan dengan data video digital. Kompresi video diperlukan agar penulisan data video dalam file menjadi lebih efisien. Kompresi juga diperlukan dalam streaming video agar transmisi data menjadi lebih cepat dan tidak memakan terlalu banyak bandwidth.

Kompresi adalah pengubahan data kedalam bentuk yang memerlukan bit yang lebih sedikit, biasanya dilakukan agar data dapat disimpan atau dikirimkan dengan lebih efisien. Jika kebalikan dari proses ini, yaitu dekompresi, menghasilkan data yang sama persis dengan data aslinya, maka kompresi tersebut disebut *lossless compression*. Sebaliknya, dekompresi tersebut menghilangkan sebagian data, maka disebut *loosy compression*. *Loosy compression* biasanya diterapkan dalam kompresi data berupa gambar. Walaupun tidak dapat menghasilkan data yang sama persis dengan aslinya, namun dianggap lebih efisien.

Video pada dasarnya merupakan *array* tiga dimensi. Dua dimensi digunakan untuk menggambarkan ruang pergerakan gambar, dan satu dimensi menggambarkan waktu. Sebuah *frame* adalah kumpulan *pixel* pada suatu waktu. Pada dasarnya, *frame* sama dengan gambar.

Data video mengandung *redundancy* (pengulangan). Kesamaan tersebut dapat dikodekan dengan mencatat perbedaan dalam sebuah *frame* atau antara *frame*. Kompresi video pada umumnya mengurangi pengulangan tersebut dengan *loosy compression*.

Pada saat ini, hampir semua kompresi video menerapkan *Discrete Cosine Transform* (DCT). Metoda lain seperti *fractal compression*, *matching pursuits*, dan *discrete wavelet transform* (DWT) tidak banyak digunakan karena kurang efektif.

## 2. Discrete Cosine Transform

Perkembangan aplikasi gambar digital telah meningkatkan kebutuhan akan teknik kompresi gambar dan video yang standar dan efektif. Standar yang banyak digunakan adalah JPEG untuk gambar, MPEG untuk video dan H.261 untuk *video teleconference*. Ketiga standar tersebut menggunakan teknik dasar yang disebut *Discrete Cosine Transform* (DCT).

*Discrete Cosine Transform* adalah sebuah teknik untuk mengubah sebuah sinyal kedalam komponen frekuensi dasar. Teknik ini biasanya digunakan dalam kompresi gambar dan video.

## 3. Standar Kompresi Video

Antara tahun 80 – 90an, algoritma kompresi berbasis *Discrete Cosine Transform* (DCT) dan standar internasional dikembangkan untuk mengurangi penyimpanan dan keterbatasan bandwidth yang disebabkan oleh gambar digital dan aplikasi video. Sekarang ada tiga standar berbasis DCT yang banyak digunakan dan diterima secara luas.

- JPEG (Joint Photographic Expert Group)

- H.261 (Video codec for audiovisual service)
  - MPEG (Motion Picture Expert Group)
- Masing – masing standar baik untuk aplikasi yang khusus : JPEG untuk kompresi gambar, H.261 untuk konferensi video, dan MPEG untuk system multimedia berkualitas tinggi.

#### 4. Standar Kompresi Video

Sebagaimana telah disebutkan sebelumnya, standar kompresi video dengan JPEG, H.261 dan MPEG semuanya berbasis pada DCT. Skema kompresi standar secara sederhana adalah sebagai berikut: bagi gambar dalam 8 x 8 blok, tentukan informasi penting dari gambar yang akan dikompres, abaikan informasi yang tidak terlalu penting, dan sandikan informasi gambar yang penting dengan jumlah bit seminimal mungkin.

Fungsi yang biasa digunakan adalah:

- DCT
- Zig-zag scanning
- Quantization
- Entrophy Coding
- Motion Estimation

#### 5. DCT dan Zig-Zag Scanning

Discrete Cosine Transform berhubungan erat dengan Discrete Fourier Transform (FFT) dan, sehingga menjadikan data direpresentasikan dalam komponen frekuensinya. Demikian pula, dalam aplikasi pemrosesan gambar, DCT dua dimensi (2D) memetakan sebuah gambar atau sebuah segmen gambar kedalam komponen frekuensi 2D (dua dimensi nya).

Untuk aplikasi kompresi video, jika variasi dalam blok cenderung rendah, kebanyakan transformasi ini akan menghasilkan representasi blok yang lebih kompak . Blok dipadatkan dalam ‘bin’ dengan frekuensi yang lebih rendah yang sesuai.

#### 6. Kuantisasi

Kuantisasi adalah data sumber utama yang hilang dalam algoritma kompresi image yang berbasis DCT. Kuantisasi mengurangi jumlah informasi yang dibutuhkan untuk merepresentasikan frekuensi bin dengan mengkonversi amplitude dalam range tertentu mejadi satu dalam kumpulan level kuatisasi. Secara sederhana, semua standar dari semua algoritam kompresi image menggunakan kuantisasi linier dimana level dari ukuran kuantisasi adalah konstan.

Kuantisasi dalam domain frekuensi mempunyai banyak keuntungan secara langsung dalam mengkuantisasi nilai pixel. Kuantisasi dalam nilai pixel menghasilkan efek fisual yang dinamakan distorsi kontu dimana perubahan kecil amplitude dalam wilayah gradient menyebabkan perubahan peningkatan ukuran dalam rekonstruksi simpangan. Kecuali untuk DC bin,

kesalahan kuantisasi untuk tiap-tiap tempat penyimpanan frekuensi rata-rata mendekati 0 untuk blok 8x8.

#### 7. Entropy Coding

*Entropy coding* adalah sebuah skema *lossless* kompresi berbasis pada properti statistik dari gambar atau aliran informasi yang dikompres. Meskipun *entropy coding* diimplementasikan secara berbeda untuk tiap-tiap standar, dasar dari skema entropy coding adalah dengan menyandikan pola yang paling sering muncul dengan jumlah bit yang paling kecil. Dengan cara ini, data dapat dimampatkan dengan faktro tambahan dari 3 atau 4.

*Entropy coding* untuk aplikasi pemampatan video mempunyai dua proses : Zero-Length Coding(RLC) dan kode Huffman.

Data RLC adalah representasi simbolik dari tempat penampung yang terkuantisasi yang memanfaatkan sepasang angka. Angka pertama merepresentasikan jumlah dari 0 yang berturutan sedangkan yang kedua melambangkan jumlah dari nilai antara panjang zero-run. Sebagai contoh kode RLC(5,8) melambangkan urutan angka dari (0,0,0,0,0,8).

Kode Huffman menempatkan variable panjang kode menjadi data RLC, menghasilkan variable panjang data aliran bit. Hal ini memerlukan table Huffman yang dapat di komputasi kembali yang berbasis pada properti statistic dari image (sebagaimana dalam JPEG) atau dapat kembali ditentukan kembali jika table default sedang digunakan. Dalam kasus yang lain, table yang sama digunakan untuk mendecode aliran bit data.

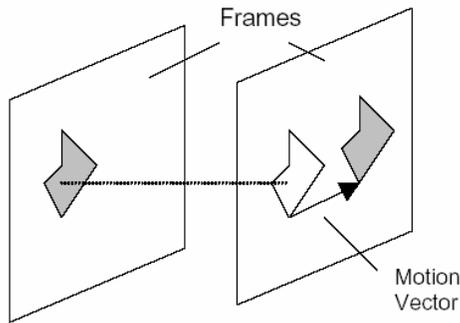
Seperti dijelaskan diatas, pola RLC yang sering muncul disandakan dengan jumlah bit yang paling kecil. Dalam hal ini aliran digital, yang merupakan representasi digital dari image, tidak memiliki batasan atau panjang yang tetap. Informasi ini sekarang dapat disimpan atau disiapkan untuk pengiriman.

#### 8. Motion Estimation

Secara umum, motion video yang berurutan cenderung mempunyai hubungan erat, sehingga, pergantian gambar ditampilkan dalam waktu yang sangat cepat dalma periode waktu yang singkat. Hal ini mengakibatkan perbedaan secara aritmatika antara gambar sangat kecil. Untuk alasan ini, rasio kompresi untuk motion video yang berurutan meningkat dengan menyandakan perbedaan aritmatika diantara dua atau lebih frame yang saling berhubungan.

Perkiraan motion adalah proses dimana elemen dalam gambar mempunyai korelasi terbaik dengan elemen di gambar yang lain (didepan atau dibelakang) dengan memperkirakan jumlah dari motion. Jumlah dari motion dibungkus dalam vektor motion. Motion vector

selanjutnya mengacu pada *motion vector* yang ada sebelumnya.



Algoritma perkiraan motion yang efisien meningkatkan korelasi frame, dimana dapat meminimalkan pixel perbedaan aritmatik. Menghasilkan bukan hanya rasio kompresi yang tinggi tapi juga kualitas video yang didecode. Perkiraan pergerakan sesaat merupakan operasi komputansi intensif yang sulit untuk diimplementasikan secara real-time.

### 9. Algoritma kompresi JPEG

Algoritma JPEG dibuat untuk menghasilkan kompresi gambar film secara efisien. Dalam penggunaan kompresi gambar film, JPEG juga sudah disesuaikan untuk penggunaan dengan gerakan video yang berurutan. Penyesuaian ini menggunakan algoritma JPEG untuk mengkompres setiap frame sendiri-sendiri pada urutan gerakan video.



Setiap bagian warna dari gambar film diperlakukan sebagai gambar yang terpisah oleh JPEG. Meskipun JPEG memungkinkan beberapa pembagian warna, gambar biasanya dibedakan dalam warna merah, hijau, biru. Atau Luminance (Y), dengan perbedaan warna biru dan merah ( $U=B-Y$ ,  $V=R-Y$ ).

Pembagian dalam bagian warna YUV memungkinkan algoritma ini untuk memanfaatkan kelemahan sensitifitas mata manusia.

JPEG membagi setiap komponen warna gambar dalam  $8 \times 8$  blok pixel. DCT  $8 \times 8$  diterapkan pada setiap blok. Untuk kuantisasi, JPEG menggunakan matrix. JPEG memungkinkan perbedaan matrix kuantisasi ditentukan untuk setiap bagian warna.

Penggunaan matrix kuantisasi memungkinkan setiap frekuensi untuk dikuantisasi pada setiap tahapan yang berbeda. Secara umum, frekuensi yang lebih rendah pada komponen dikuantisasi menjadi tahapan yang kecil dan frekuensi tingginya menjadi tahapan yang besar. Hal ini menguntungkan karena mata manusia tidak terlalu sensitif pada frekuensi yang tinggi, tapi lebih sensitif pada frekuensi yang lebih rendah.

Modifikasi pada matrix kuantisasi adalah cara yang utama untuk mengontrol kualitas dan rasio kompresi pada JPEG. Meskipun ukuran tahapan kuantisasi untuk setiap frekuensi dapat dimodifikasi, sebagian besar teknik adalah untuk memperhitungkan elemen matrix bersama-sama.

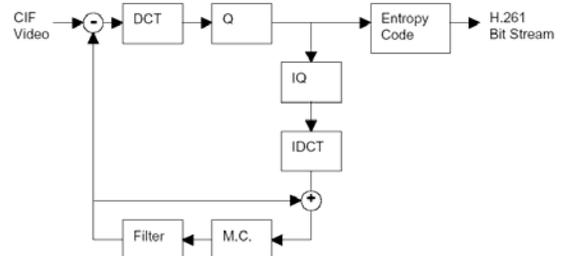
Tahapan terakhir dari kompresi adalah zig-zag scanning dan pengkodean entropi. Meskipun JPEG memungkinkan 2 jenis entropy coder, penerapan JPEG lebih mirip menggunakan pilihan pengkodean Huffman. Standar JPEG memungkinkan untuk menggunakan table kode Huffman yang didefinisikan sendiri. Untuk melakukan dekompres pada gambar JPEG, setiap proses ditunjukkan dalam urutan yang terbalik.

### 10. Algoritma Kompresi H.261

Algoritma H.261 dimaksudkan untuk diterapkan pada aplikasi Video conferencing dan video telephony. Pada aplikasi ini, keterbatasan gerakan video menjadi bagian terpenting.

Agar lebih efektif, H.261 menentukan banyak parameter system. Hanya pembagian warna YUV dengan perbandingan 4:2:0 dapat digunakan secara standar. Sebagai tambahan, H.261 hanya menggunakan 2 ukuran frame, CIF ( $352 \times 288$ ) dan QCIF ( $176 \times 144$ ).

Seperti standar JPEG, setiap bagian warna gambar dibagi dalam  $8 \times 8$  blok pixel. Disamping mengkodekan setiap blok secara terpisah, H.261 mengelompokkan 4 blok Y, 1 blok U, dan 1 blok V bersama-sama dalam satu unit yang disebut macroblock. Macroblock adalah unit dasar untuk kompresi.



Untuk mengkompres setiap macroblock, standar H.261 memungkinkan kompresor untuk memilih dari beberapa pilihan kompresi. Standar H.261 hanya menentukan proses decoding dari setiap pilihan kompresi. Cara yang digunakan tidak dibakukan. Hal ini memungkinkan untuk menghasilkan produk yang berbeda dengan menggunakan cara yang berbeda efektifitasnya. Salah satu cara yang digunakan untuk mengkompres H.261 adalah sebagai berikut.

Pertama, perkiraan gerakan ditunjukkan pada setiap macroblock. Karena objek dalam frame mungkin bergerak dalam arah yang berbeda, setiap macro block dapat mempunyai vektor gerakan yang berbeda. Vektor gerakan digunakan sebagai vektor pemindahan untuk mengambil sebuah macroblock dari frame yang

sebelumnya untuk digunakan sebagai perkiraan. Perkiraan gerakan dalam H.261 adalah relatif pada frame yang sebelumnya, dan pada pencetakan full-pixel mencapai maksimal pada +/- 15 arah vertikal dan horizontal

Selanjutnya, keputusan harus dibuat untuk mengkodekan perbedaan aritmatik antara prediksi pada macroblock dan macroblock yang sebenarnya atau mengkodekan macroblock dari rancangan yang dibuat. Karena perbedaan aritmatik biasanya kecil, pengkodean perbedaan aritmatik menghasilkan kompresi yang lebih baik.

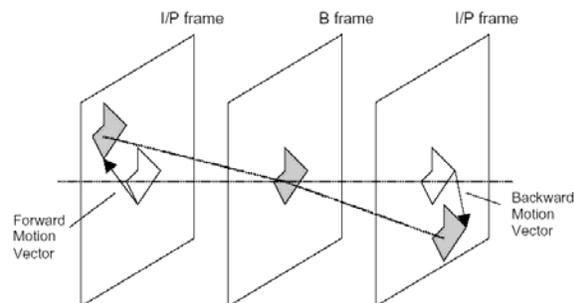
Tahapan terakhir pada proses kompresi adalah zig-zag scanning, run-length encoding dan entropy coding. Tidak seperti JPEG, H.261 menentukan table kode Huffman untuk mengkodekan entropi.

Untuk melakukan decompress sebuah proses pembalikan frame H.261 dilakukan dalam urutan yang terbalik. Perkiraan gerakan tidak diperlukan karena vektor gerakan sudah termasuk dalam bit hasil kompresi. Dekompresor H.261 secara mudah menerapkan pencetakan vektor gerakan untuk menerima kembali perkiraan jika diperlukan.

## 10. Algoritma kompresi MPEG

Algoritma kompresi MPEG dikembangkan untuk memenuhi kebutuhan akan gambar dengan kualitas yang lebih baik dan untuk meningkatkan fleksibilitas system yang diperlukan oleh system multimedia. Karena dikembangkan lebih akhir, MPEG dapat meningkatkan usaha dibalik pengembangan algoritma JPEG dan H.261.

Sebagaimana H.261, MPEG standar hanya menggunakan pemisahan komponen warna YUV dengan *sampling ratio* 4:2:0. Tidak seperti H.261, ukuran *frame* tidak dibatasi, walaupun ukuran *frame* 352 x 240 biasa digunakan. MPEG mengadopsi *macroblock* dari H.261 (4 blok Y, 1 blok U, dan 1 blok V) sebagai unit dasar kompresi. Untuk mengkompres tiap *macroblock*, MPEG standar mengizinkan kompresor untuk memilih dari beberapa pilihan kompresi.



Ada lebih banyak pilihan yang tersedia dalam MPEG standar dari pada H.261. Sebagaimana H.261, MPEG standar hanya menspesifikasikan *decoding* pada setiap pilihan kompresi. Metode untuk memilih pilihan tersebut tidak distandardisasi, sehingga memungkinkan vendor untuk membedakan produk mereka dengan menyediakan metoda dengan biaya dan kualitas yang berbeda. Berikut ini adalah metoda yang biasa digunakan dalam kompresi MPEG.

Pertama, estimasi pergerakan dilakukan pada tiap *macroblock*. MPEG mampu melakukan prediksi terhadap *frame* sebelumnya, sesudahnya atau kombinasi dari keduanya.

Karena objek dalam *frame* tidak bergerak secara tetap dari *frame* ke *frame*, setiap *macroblock* dapat memiliki *motion vector* hingga dua buah (satu untuk *frame* sebelumnya dan satu lagi untuk *frame* sesudahnya). Untuk melakukan prediksi terhadap *frame* selanjutnya, harus dilakukan *buffer* terhadap *frame* ekstra.

Estimasi pergerakan juga dapat dilakukan hingga *range* yang lebih besar (hingga  $\pm 1023$ ) dan dengan resolusi *half-pixel*. *Loop-filter* pada H.261 tidak dimasukkan dalam MPEG karena *half-pixel motion vector* menyediakan fungsi yang sama.

MPEG dapat melakukan prediksi yang dibentuk dari perbedaan aritmatika antara *macroblock* sekarang dengan *macroblock* dari *frame* sebelumnya, *frame* selanjutnya, rata-rata antara *frame* sebelumnya dengan *frame* selanjutnya atau mengkodekan *macroblock* sekarang dari awal. Sebuah DCT 8x8 diaplikasikan kedalam masing-masing blok pada *macroblock* sekarang. MPEG menggunakan matriks (seperti JPEG) dan faktor skala untuk kuantisasi. Karena *DC bin* adalah yang paling penting, maka dikuantisasikan dalam 8 skala bit tetap.

Karena visual efek dari kuantisasi *frequency bin* berbeda antara blok perkiraan dengan blok sekarang, MPEG dapat menggunakan dua matriks (masing – masing satu untuk tiap tipe). Biasanya, matriks tersebut diset sekali untuk urutan gambar dan skala kuantisasinya disesuaikan untuk mengontrol rasio kompresi.

Tahap paling akhir dari kompresi adalah *zig-zag scanning*, *run-length encoding* dan *entropy coding*. Seperti H.261, MPEG menspesifikasikan tabel kode Huffman untuk *entropy coding*.

Untuk mendekompres *frame* MPEG, setiap operasi dilakukan secara terbalik, kecuali untuk estimasi pergerakan. Karena vektor pergerakan dimasukkan dalam *bit-stream* yang dikompresi, dekompresor MPEG hanya cukup menerapkan vektor pergerakan untuk memprediksi *frame* sebelumnya maupun *frame* selanjutnya jika diperlukan

## 11. Kesimpulan

Kompresi video diperlukan untuk memperkecil ruang penyimpanan maupun mempermudah transmisi video untuk berbagai keperluan. Inti dari kompresi video ini adalah memperkecil ukuran mengurangi kualitas gambar, karena mata manusia tidak cukup peka untuk membedakan gambar terkompresi dengan gambar aslinya. Sehingga kompresi dalam video hampir selalu menggunakan *loosy compression*, yaitu kompresi yang dapat menghilangkan sebagian data yang dikompresi.

Ada banyak sekali metoda dan algoritma kompresi video ini. Setiap system kompresi melibatkan perhitungan antara besarnya kompresi, kecepatan yang dihasilkan dalam kompresi dan dekompresi serta kualitas video. Tidak ada suatu format yang dapat dikatakan paling baik. Pilihan kompresi sebaiknya disesuaikan dengan kebutuhan. Metoda kompresi video ini terus berkembang dari waktu ke waktu, disesuaikan dengan perkembangan teknologi kompresi dan kebutuhan.

Diharapkan pada masa yang akan datang, ditemukan metoda atau algoritma kompresi yang lebih efektif dan efisien, sehingga pertukaran dan penyimpanan data video dan gambar dapat dilakukan dengan mudah tanpa memakan banyak tempat, bandwidth dan biaya.

## 12. Referensi

- [1] [http://en.wikipedia.org/Discrete\\_Cosine\\_Transform](http://en.wikipedia.org/Discrete_Cosine_Transform) diakses tanggal 17 Mei 2006
- [2] [http://en.wikipedia.org/Video\\_Compression](http://en.wikipedia.org/Video_Compression) diakses tanggal 17 Mei 2006
- [3] Jeef Bier., *Introduction to Video Compression*. Bercekeley Design Technology Inc.
- [4] <http://www.animemusicvideos.org/guides/avtech/video4.htm> diakses tanggal 17 Mei 2006
- [5] Array Microsystem Inc., *Video Compression An Introduction*.
- [6] Gary A. Thom, Alan R. Deutermann, *A Comparison Of Video Compression Algorithms*. Delta Information Systems, Inc. 2001