II. TEORI PENUNJANG

1. COMPUTER VISION DAN IMAGE PROCESSING

Computer vision adalab. salab. satu cabang dari artificial intelligence (kecerdasan buatan) yang difokuskan pada pengembangan algoritma untuk nienganalisa isi dari suatu gambar. Beberapa pendekatan diberikan untuk mengenali gambar telah dikembangkan.

Computer Vision mempunyai tujuan utama untuk membuat keputusan yang berguna tentang objek fisik nyata dan pemandangan *(scenes)* berdasarkan *image* yang didapat dari sensor. *Computer Vision* ingin membangun sebuah mesin pandai yang dapat melihat. Tentunya hal ini bukanlah hal mustahil. Ada berbagai contoh dari aplikasi *computer vision* seperti mesin yang mengawasi atau memeriksa jvrtaan filaraen dari bola lampu atau ribuan mil serat pabrik setiap hari. ATM telah dibangun dilengkapi dengan retina *scan.* Mobil sudah dapat dikemudikan oleh komputer dengan menggunakan kamera sebagai input.

Pengertian sederhana dari *image processing* adalah manipulasi dan analisis suatu informasi gambar oleh komputer. Yang di maksud dengan informasi gambar di sini adalah gambar visual dalam dua dimensi. Segala operasi vintuk memperbaiki, analisa, atau pengubahan suatu gambar disebut *image processing.* Konsep dasar dari sistem dari *image processing* diambil dari keinampuan indera penglihatan manusia yang selanjutnya dihubungkan dengan kemampuan otak manusia. Dalam sejarahnya, *image processing* telah

diaplikasikan dalam berbagai bentiik, dengan tingkat kesuksesan yang cukup besar. Seperti berbagai cabang ilmu iainnya, *image processing* menyangkut pula berbagai gabungan cabang-cabang ilmu. Seperti diantaranya optik, elektronik, matematika, fotografi, dan teknologi komputer.

Beberapa faktor menyebabkan perkembangan sistern *image processing* menjadi lebih berkembang lebih pesat saat ini. Salah satu yang utama adalah penurunan biaya akan peralatan komputer yang dibutuhkan. Kedua peralatan unit *processing* dan *bulk storage* menjadi semakin murah dari tahun ke tahun. Faktor kedua adalah peningkatan tersedianya peralatan untuk proses digital dan tampilan gambar.

Berbagai bidang telali banyak menggimakan aplikasi dari *image processing* baik dibidang komersial, industri, dan medik. Bahkan bidang militer telah menggunakan perkembangan dunia digital *image processing* ini.

Pada umumnya, tujuan dari *image processing* adalah mentransformasikan atau menganalisis suatu gambar seliingga informasi baru tentang gambar dibuat lebih jelas. Ada banyak cara yang dapat diaplikasikan dalam suatu operasi *image processing.*

Hampir sebagian besar dalam bentuk optikal. Gambar optikal dikonversikan menjadi sinyal elektrik dengan menggunakan kamera video atau peralatan lain sejenisnya. Konversi ini merubah representasi gambar dari suatu cahaya optik menjadi sinyal elektrik kontinyu. Sinyal elektrik ini disebut sinyal analog. Lebih lanjut, gambar analog dapat didigitalkan dan berubah menjadi data digital. Operasi pada sistem *image processing* dapat diaplikasikan pada suatu gambar dengan bentuk optikal, analog, atau digital.

2. OPEN SOURCE COMPUTER VISION LIBRARY

Open Source Computer Vision Library mulai dikembangkan dua tahun yang lalu oleh *Visual Interactivity Group* didalam *Intel's Microprocessor Research Lab.* Proyek ini dibuat dengan tujuan untuk mendirikan sebuah komvinitas *open source vision* dan menyediakan sebuah situs dimana usaha terdistribusi dari komunitas dapat dikonsolidasikan dan *perfbmance-nya* dapat dioptimalkan. *Library* ini ditujukan untuk digunakan oleh peneliti dan pengembang software komersial. *Open Source Compnier Vision Library* dibuat berdasarkan fungsi-ftmgsi dasar yang terdapat pada *Intel Performance Library.* Keunggulan *lihrary* ini adalah semua fungsi-fiingsinya telah dioptimasi untuk prossesor Intel sehingga dapat berjalan jauh lebih cepat

Open Source Computer Vision Library dibuat berdasarkan ftingsi-fungsi dasar dan library dari Intel® *Image Processing Library.* Intel® *lmage Processing Library* menyediakan sekumpulan ftingsi-fUngsi C sangat teroptimasi yang mengimplementasikan fungsi-fungsi *image processing* pada prosessor berarsitektur Intel.

2.1. Cara Menginstall OPEN SOURCE COMPUTER VISION LIBRARY

Instalasi OpenCV *{Open Source Computer Vision Library)* pada Windows 98/2000/XP:

1. Dahului dengan instalasi Microsoft Visual C++ ver 6.0® (MSVC 6.0)
2. Instal DirecX 8.1 SDK dan taruh/arahkan instalasi pada directory c:\mssdk (ganti directoty default c:\dxsdk menjadi c:\mssdk). Jawab

Yes ketika ditanya apakah ingin menambahkan path directory pada MSVC. DirecX SDK ini dibutukan imtuk akses informasi multimedia (video, sound, streaming dll).

3. Jalankan ipl25.exe iintuk instlasi *Intel Image Processing Library. Library* ini dibutuhkan pada banyak aplikasi OpenCV.
4. Instal OpenCV ver 2.0 dengan menjalankan opencv_core_b2_0.exe (library dasar), lalu lanjutkan dengan menjalankan opencv_apps_b2_0.exe imtuk instalasi aplikasi OpenCV.
5. Upgrade OpenCV dengan versi 2.1
   - Extrak opencv_core_b2.1 .zip dan taruli pada dlrectory c:\Program File\Intel\
   - Extrak opencv_apps_b2.1 .zip dan taruh pada directory c:\Program File\Intel\
   - Extrak opencv_patch_b2.1.1. zip dan taruh pada directory c:\Program File\Mel\
   - Extrak opencv_dll_addon_b2.1.1. zip dan taruh pada directory c:\Program File\lntel\
6. Perhatikan path directory pada MSVC (klik Tools>Options>Directories), dan tambahkan include dan library path:
   - Pada include path:
     - Dari MS DirectShow SDK:
       - C:\Mssdk\Samples\Multimedia\Directshow\Baseclasses
       - C :\Mssdk\Samples\Multimedia\Common\Include
       - CAMssdk\Include

- Dari OpenCV list:
  - C:\Program Files\Intel\OpenCV\caux\Include
  - C:\Program Files\Intel\OpenCV\cv\Include
  - C:\Program Files\lntel\OpenCV\Apps\Common
  - CAProgram Files\Intel\OpenCV\Otherlibs\Vlgrfints
  - C:\Program Files\Intel\openCV\otherlibs\highgui
- Dari IPL:
  - C:\Program Files\Intel\Plsuite\Include

Libraries yang perlu disertakan untuk linking:

- Dari MS DirectShow SDK:
  - C:\Mssdk\Samples\Multimedia\Directshow\baseclasses\release
  - C:\Mssdk\Samples\Multimedia\Directshow\baseclasses\debug
  - C:\Mssdk\Lib
- Dari OpenCV:
  - C:\ :\Program Files\Intel\OpenCV \Lib
  - C:\Program Files\Intel\OpenCV\Otherlibs\Highgui
- Dari Ipl:
  - C:\Program Files\Intel\plsuite\Lib\Msvc

7. Proses instalasi selesai, dan selanjutnya perhatikan struktur directory OpenCV pada c:\Program File\Intel\OpenCV

```
l
 + _DSW  // Workspace file for Microsoft Visual Studio and a few
Perl utilities for statistics
 l
 + CV    // The library itself
 I  + Include   // External library interface
 |  + _Inciude  // Internal library interface
 I  + Make      // Project file
 I  + Src       // Source files
 1
 + CVAux // Additional (experimental) stuff.
 I
 + Docs  // documentation
 I  + HTML      // overview documentatation in HTML format
 1
 + Bin   // all the pre-built binaries
 I
 + Lib   // pre-built import and static libtaries
 I
 + Apps  // Demo Applica,tions

 I  + CamShiftDemo // Application - Wrapper for CamShift Tracker
Filter
 I  + VMDemo       // View Morphing demo
 I  + LkDemo       // Lucas-Kanade Pyramid-based Point Tracker
 I  + HMMDemo      // Hidden Markov Models Face Recognition Demo
 |  + StereoGR     // Stereo-based Gesture Recognition App for
PointGrey Stereo Cameras
 I  + Hawk         // Scripting Environment
```

```
    + Common       // Clmage and CCamera classes
 I
 + Pilters  // Direct Show filters
 |   + CalibFilter // Camera Calibration filter
 |   + CamShift    // CamShift tracker
 |   + Condens     // ConDensation based tracker
 |   + Kalman      // Kalman filter based tracker
 |   + ProxyTrans  // Proxy DirectShow filter
 I
 + OtherLibs
 I    + __Mkl      // Math Kernel Library - used for tests on
matrix functions
 I    + _IPL       // Image Processing Library - base iibrary for
the OpenCV
 ]    + HighGUI    // Simple GUI library with platform-independed
interface
 i    + viGrFmts   // Library for reading/writing raster images
 I    + GestRec    // Experimental gesture recognition module
 |    + PtGrey     // Interface Module for PointGrey Stereo Camera
 1
 + Tests// sources for algorithmic tests
```

Untuk aplikasi-aplikasi menggunakan video/webcam, membutiikan *library* DirecShow. Build baseclasses dari direcshow, dengan open project pada MSVC yang ada pada C:\mssdk\samples\Multimedia\DirectShow\BaseClasses\baseclasses.dsw lalu build secara batch baik untuk versi release maupun debug

3. METODEPENGENALANWAJAH

3.1 GaborWavelet

Salah satu bentuk teknik feature *extraction* adalah dengan menggunakan Gabor filter. Fungsi wavelet dan fimgsi gabor dikenal sangat variatif. Dalam menggabungkan pendapat yang berbeda, namun pada prinsipnya semuanya itu akan menghasilkan ftmgsi dengan tujuan yang sama. Tujuan utama Gabor Wavelet ini dalam pengolahan adalali untuk memunculkan ciri-ciri khusus dari gambar yang telah dikonvolusi terhadap kemel. Proses yang berlangsung dalam bidang frekuensi mempengaruhi kecepatan proses yang terjadi, baik dalam preproses gambar maupun saat proses konvolusi.

3.2 NilaiJet

Pada Gabor wavelet diinotivasi oleh konvolusi kemel dalam bentuk *plane wave* atau bidang gelombang yang dibatasi oleh sebuah fimgsi Gaussian envelope. Kumpulan dari koefisien-koefisien convolution untuk kemel dari orientasi-orientasi dan frekuensi-frekuensi yang berbeda pada satu *image pixel* yang di namakan *jet.*

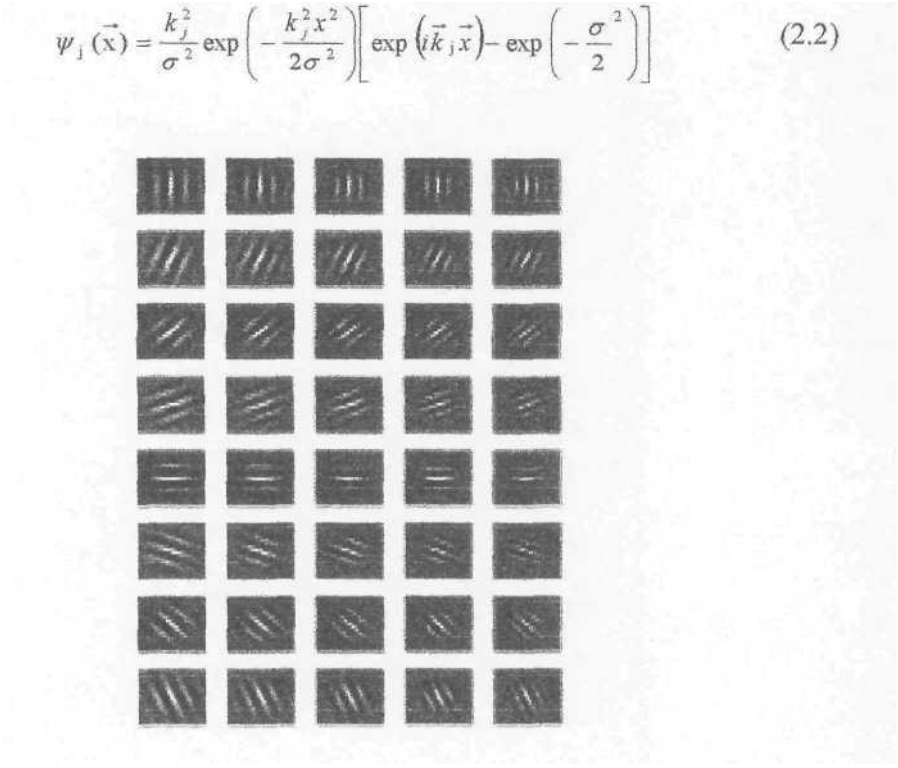
Jet merupakan kumpulan dari *grey values* dalam sebuah gambar

I (x) mengelilingi sebuah pixel yang diberikan x = (x,y). Hal ini didasarkan pada wavelet transforai, didefinisikan sebagai proses konvolW[51]

$$J_j(\vec{x}) = \int I(\vec{x})\,\psi_j(\vec{x}-\vec{x}')\,d^2\vec{x}' \qquad (2.1)$$

dengan sebuah kumpulan dari *Gabor kemefi^*

$$\psi_j(\vec{x}) = \frac{k_j^2}{\sigma^2}\exp\left(-\frac{k_j^2 x^2}{2\sigma^2}\right)\left[\exp\left(i\vec{k}_j\vec{x}\right) - \exp\left(-\frac{\sigma^2}{2}\right)\right] \quad (2.2)$$



Gambar 1.2

Gabor kemel[5]

dalam bentuk *plane wave* dengan wave vektor k_j, dibatasi oleh sebuah fungsi *Gaussian envelope.* Dan digunakan sebuah kumpulan terdiri dari 5 frekuensi yang berbeda, index v = 0,..,4 dan 8 *onentation,* index

$\mu = 0,\ldots,7$ [5]

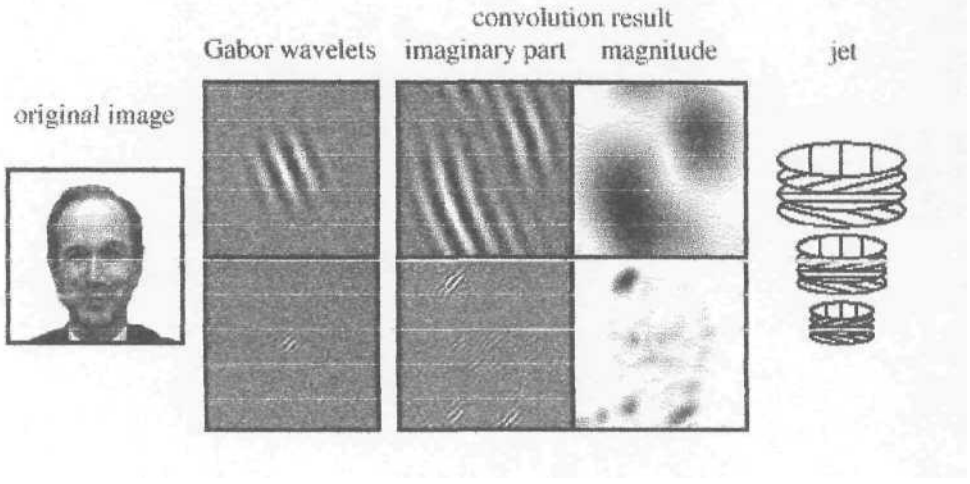
$$\vec{k}_j = \begin{pmatrix} k_{jx} \\ k_{jy} \end{pmatrix} = \begin{pmatrix} k_v \cos\varphi_\mu \\ k_v \sin\varphi_\mu \end{pmatrix} \quad (2.3)$$

dimana

$$k_v = 2^{-\frac{v+2}{2}}\pi$$

$$\varphi_\mu = \mu\frac{\pi}{8}$$

dengan index $j = \mu + 8v$. Sampling ini mengcover sebuah ikatan dalam frekuensi space. Kelebaran σ/k dari *Gaussian* dikendalikan oleh parameter $\sigma = 2\pi$. Hal ini diketahui sebagai perubahan wavelet karena kelompok kernel-kernel adalah sama, semua kernel di*generated* dari satu induk wavelet oleh dilatasi dan rotasi



Gambar 1.3

Hasil Konvolusi Dengan Gabor Kemel[w]

Pada gambar 1 perwakilan grafik dari sebuah wajah digunakan pada perubahan Gabor wavelet, sebuah *convolution* dengan sebuah rangkain dari kernel-kernel wavelet. Ini mempunyai bentuk dan plane wave yang dibatasi oleh sebuah flmgsi *Gaussian enveiope.* Dengan raenghitung 40 koefisien (5 frekuensi x 8 *orientatwn).* Fase koefisien *approximately* dengan frekuensi wavelet. (lihat bagian imaginari), magnitude varies secara perlahan-lahan. Rangkaian dari 40 koefisien diperoleh 1 titik gambar ditunjuk sebagai sebuah *jet* (untuk kejelasan, hanya 3 frekuensi dan 4 orientasi dijelaskan dalam gambar). *Jet* bersama-sama dengan beberapa mformasi tentang lokasi yang berhubungan membentuk sebuah

grafik gambar, yang dipakai untuk menjelaskan sebuah benda, seperti wajah.

Sebuah *Jet* adalah kumpulan 40 koefisien yang kompleks yang didapat dari satu titik pada suatu gambar[5]

$$J_j = a_j \exp(i\varphi_j) \quad (2.4)$$

dengan magnitude $a_j(\vec{x})$dan phasa $\phi_j(\vec{x})$, akan memberikan ragam perubahan pada pendekatan karakteristik frekuensi dari kernel Gabor yang digunakan.

Gabor wavelet dipilih karena kekuatannya sebagai sebuah format data dan karena hubungan biologicalnya. Gabor Wavelet akan memberikan kekuatan melawan *brightness* yang berbeda-beda pada gambar. Kekuatan melawan kontras yang berbeda-beda dapat diperoleh dengan cara menormalkanye^/ef. Lokasi yang terbatas dalam space dan frekuensi menghasilkan sejumlah kekuatan tertentii melawan translasi, distorsi, rotasi, dan scalling. Hanya fase yang berubah secara drastis dengan translasi. Kekurangan dari titik-titik yang besar adalah kesensitivitas terhadap variasi-variasi background. Hal ini ditunjukkan, akan tetapi jika garis bentuk objek diketahui, pengaruh pada *backgmund* dapat ditekan.

3.3 Membandingkan Jet

Dari semua variasi yang ada, perbandingan *jets* tidak dapat dilakukan secara langsung karena perubahan kecil pada jarak dalam bidang *spatial* akan berpengaruh pada koefisien individual secara drastis.

Karena itu dapat digunakan baik hanya nilai magnitude ataupun menganggap besar terjadinya perababan phasa terhadap perabahan jarak yang memungkinkan. Karena fase rotasi tersebut, *jets* dari titik pada gambar yang berjarak hanya beberapa pixel satu sama lainnya dapat memiliki koefisien yang berbeda walaupun mewakili bentuk-bentuk lokal yang hampir sama. Dengan adanya variasi tersebut maka harus digunakan suatu fungsi yang bertujuan untiik mencari koefisien yang benar-benar tepat, yaitii dengan menggunakan persamaan^ :

$$S_a(J,J') = \frac{\sum_j a_j \cdot a'_j}{\sqrt{\sum_j a_j^2 \cdot \sum_j a_j'^2}} \qquad (2.5)$$

di mana *J*merupakan/ete pada satii posisi dan *jets* J'=J'(*x)* diambil pada posisi variabel *x,* sehingga $S_a(J,J'(x))$ merupakan fungsi yang memperhalus posisi dengan mencari nilai optimal untuk posisi jet tersebut. Fungsi ini mengabaikan perubahan phasa yang terjadi pada setiap indeks filter.