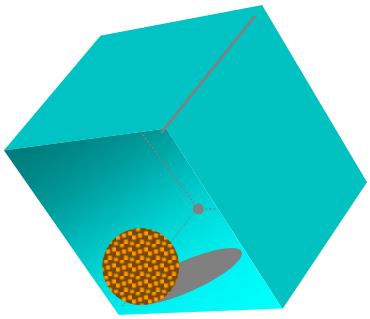
Pengolahan Citra
Pada
Mobil Robot

Tabratas Tharom

tharom@yahoo.com

BAB 4

PEMBENTUKAN MODEL OBJEK



4.1. PEMETAAN LINGKUNGAN

Pada pengolahan citra pada mobil robot, proses pemetaan lingkungan dinilai sangat penting agar lintasan dan objek serta penghalang yang dilalui mobil robot dapat didekati dengan sempurna oleh pencitraan tersebut. Pada kesempatan kali ini, proses pemetaan lingkungan dibagi dalam dua garis besar pembahasan, yakni proses segmentasi yang meliputi pendeteksian garis dan tepi, dan sedikit mengenai transformasi Hough yang menggunakan konsep *ray tracing*.

4.1.1. SEGMENTASI

Segmentasi adalah suatu proses untuk memisahkan sejumlah objek dalam suatu citra dari latar belakangnya. Proses segmentasi dapat dilakukan dengan menggunakan dua buah pendekatan sebagai berikut.

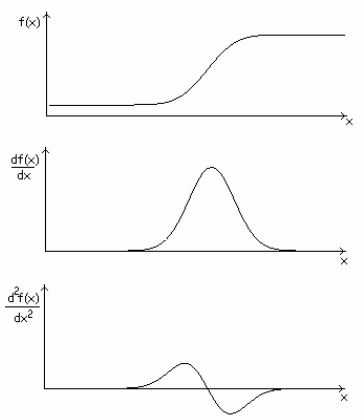
- Metode berdasarkan tepi (*edge–based*)
  Metode ini berbasiskan perbedaan atau perubahan mendadak nilai intensitas suatu piksel terhadap piksel tetangganya.
- Metode berdasarkan daerah (*region-based*)
  Metode ini berbasiskan kesamaan nilai suatu piksel terhadap piksel tetangganya.

Deteksi tepi

Tepi adalah sejumlah tempat pada citra dengan intensitas kontras yang kuat. Tepi biasanya muncul pada lokasi citra yang merepresentasikan batasan objek. Dengan demikian, pendeteksian tepi digunakan secara luas dalam proses segmentasi citra. Kita ingin membagi citra ke dalam sejumlah daerah yang menyatakan objek yang berbeda-beda.

Tepi mengandung sebagian besar komponen frekuensi tinggi. Dengan demikian, secara teori, pendeteksian tepi dapat dilakukan dengan menggunakan filter frekuensi tinggi dalam domain Fourier atau dengan cara mengkonvolusikan citra dengan kernel tertentu pada domain spasial. Secara praktis, pendeteksian tepi dilakukan dalam domain spasial karena mempunyai komputasi yang lebih sederhana, cepat, dan sering memberikan hasil yang lebih baik.

Karena tepi berhubungan dengan perubahan iluminasi yang besar, maka tepi dapat diperoleh dengan menghitung turunan citra. Hal ini direpresentasikan pada Gambar 4.1. Posisi tepi dapat diestimasi dengan nilai maksimum pada turunan pertama atau *zero-crossing* pada turunan kedua.



Gambar 4.1 Turunan pertama dan kedua dari tepi dimensi satu.

Metode pendeteksian tepi terbagi dua, yaitu metode berdasarkan turunan pertama (*gradient based*) dan metode berdasarkan turunan kedua (*Laplacian*).

Pendeteksian tepi berdasarkan gradien

Suatu citra f(x,y) akan mempunyai turunan pertama atau gradien sebagai berikut :

$$\nabla f(x,y) = G\left[f(x,y)\right] = \begin{bmatrix} Gx \\ Gy \end{bmatrix} = \begin{bmatrix} \dfrac{\partial f(x,y)}{\partial x} \\ \dfrac{\partial f(x,y)}{\partial y} \end{bmatrix} \qquad (4.1.)$$

dengan

$$\frac{\partial f(x,y)}{\partial x} \approx \frac{f(x+\Delta x, y) - f(x,y)}{\Delta x} \qquad (4.2)$$

$$\frac{\partial f(x,y)}{\partial y} \approx \frac{f(x, y+\Delta y) - f(x,y)}{\Delta y} \qquad (4.3)$$

Turunan pertama ini dapat diimplementasikan dengan menggunakan kernel 2×2, 3×3, atau kernel yang lebih besar.

Untuk kernel 2×2, gradien citra adalah sebagai berikut:

$$\begin{aligned} Gx &= f(x+1, y) - f(x, y) \\ Gy &= f(x, y+1) - f(x, y) \end{aligned} \quad (4.4)$$

Untuk kernel 3×3, gradien citra adalah sebagai berikut:

$$\begin{aligned} Gx &= \left| f(x-1, y) - f(x, y) \right| + \left| f(x+1, y) - f(x, y) \right| \\ Gy &= \left| f(x, y-1) - f(x, y) \right| + \left| f(x, y+1) - f(x, y) \right| \end{aligned} \quad (4.5)$$

Nilai gradien adalah:

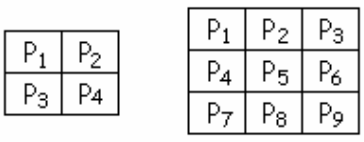
$$|G| = \sqrt{Gx^2 + Gy^2} \quad (4.6)$$

Biasanya nilai aproksimasi magnitude dihitung dengan menggunakan:

$$|G| = |Gx| + |Gy| \quad (4.7)$$

Pada fungsi yang menggunakan kernel 2×2 seperti pada gambar, nilai aproksimasi magnitudenya adalah sebagai berikut:

$$|G| = |P_1 - P_4| + |P_2 - P_3| \quad (4.8)$$



Gambar 4.2 Kernel untuk menghitung aproksimasi magnitude secara cepat.

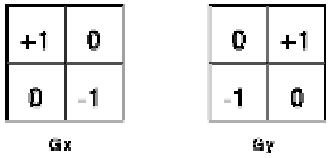
Dengan menggunakan kernel 3×3, aproksimasi magnitude adalah:

$$|G| = |(P_1 + 2\times P_2 + P_3) - (P_7 + 2\times P_8 + P_9)| + |(P_3 + 2\times P_6 + P_9) - (P_1 + 2\times P_4 + P_7)| \quad (4.9)$$

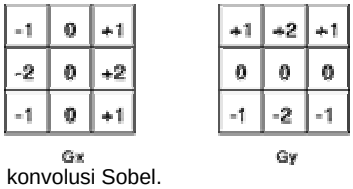
Sudut orientasi gradien terhadap gradien spasial (relatif terhadap orientasi *grid*) adalah sebagai berikut :

$$\theta = \arctan(Gy / Gx) - 3\pi / 4 \quad (4.10)$$

Implementasi metode gradien adalah seperti yang digunakan pada metode Roberts dan Sobel.



Gambar 4.3 Kernel konvolusi Roberts Cross.

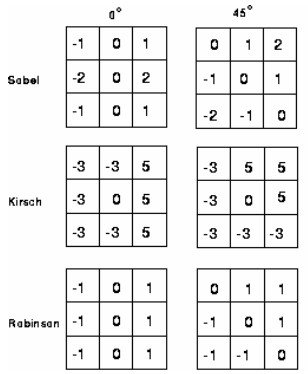


Gambar 4.4 Kernel konvolusi Sobel.

Pendekatan alternatif terhadap pendeteksian tepi gradien diferensial (Robert Cross dan Sobel operator) adalah pendeteksian tepi Compass Mask. Dalam pendeteksian tepi Compass Mask, citra dikonvolusikan dengan sejumlah kernel konvolusi (biasanya 8). Setiap kernel sensitif terhadap semua tepi dalam orientasi yang berbeda. Untuk setiap piksel, nilai gradien lokal tepi diestimasi berdasarkan respons maksimum dari semua kernel pada lokasi piksel ini.

$$|G| = \max(|G_i| : i = 1...n) \qquad (4.11)$$

Parameter G_i adalah respons kernel ke-i pada posisi piksel tersebut dan n adalah jumlah kernel konvolusi. Orientasi lokal tepi diestimasi dengan orientasi kernel yang menghasilkan nilai maksimum. Magnitude dan orientasi tepi setiap piksel kemudian ditentukan oleh *template* yang paling sesuai dengan area lokal piksel.



Gambar 4.5 Beberapa contoh kernel *compass edge* yang sering digunakan.

Ada beberapa kernel yang dapat digunakan dalam pendeteksian tepi Compass Mask. Kernel yang biasanya digunakan adalah seperti pada Gambar 4.5. Untuk setiap *template*, kedelapan kernel dapat diperoleh dengan menggeser koefisien kernel secara sirkular.

Pendeteksi tepi Laplacian

Metode pendeteksian ini menggunakan turunan kedua dari fungsi citra. Untuk suatu citra dengan fungsi *f*(*x*,*y*), turunan keduanya adalah sebagai berikut.

$$L[f(x,y)] = \nabla^2 f(x,y) = \frac{\partial^2 f(x,y)}{\partial x^2} + \frac{\partial^2 f(x,y)}{\partial y^2}$$

$$\nabla^2 f(x,y) = \frac{f(x+\Delta x,y) - 2f(x,y) + f(x-\Delta x,y)}{\Delta x^2} + \frac{f(x,y+\Delta y) - 2f(x,y) + f(x,y-\Delta y)}{\Delta y^2} \quad (4.12)$$

Untuk $\Delta x = \Delta y = 1$, turunan kedua fungsi adalah

$$\frac{\partial^2 f(x,y)}{\partial x^2} = [f(x+1,y) - f(x,y)] - [f(x,y) - f(x-1,y)]$$
$$\frac{\partial^2 f(x,y)}{\partial y^2} = [f(x,y+1) - f(x,y)] - [f(x,y) - f(x-1,y-1)] \quad (4.13)$$
$$L[f(x,y)] = f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1) - 4f(x,y)$$

Metode Laplacian ini dapat diimplementasikan dalam kernel konvolusi seperti pada Gambar 4.6.

0	-1	0
-1	4	-1
0	-1	0

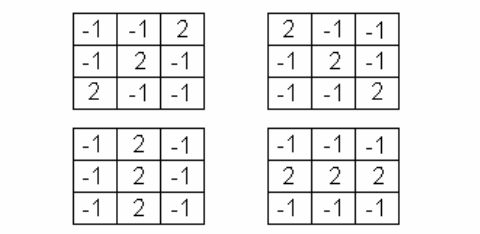
-1	-1	-1
-1	8	-1
-1	-1	-1

Gambar 4.6 Kernel konvolusi metode Laplacian

Pendeteksi garis menggunakan *template*

Tepi hasil proses pendeteksian tepi biasanya membentuk kumpulan piksel yang dapat direpresentasikan sebagai garis lurus maupun garis dengan fungsi geometri tertentu. Proses pendeteksian garis lurus pada suatu citra dapat dilakukan dengan menggunakan proses yang hampir sama dengan proses pendeteksian tepi. Proses ini sangat mirip dengan proses pendeteksian tepi *Compass Mask*. Metode ini bekerja dengan cara *template matching,* yaitu mencari kumpulan tepi yang sesuai dengan kernel yang digunakan sebagai *template*.

Metode pendeteksian garis ini dapat dilakukan dengan menggunakan kernel seperti pada Gambar 4.7. Kernel ini akan mendeteksi kumpulan tepi yang membentuk garis dengan sudut 0°, 45°, 90°, dan 135°.

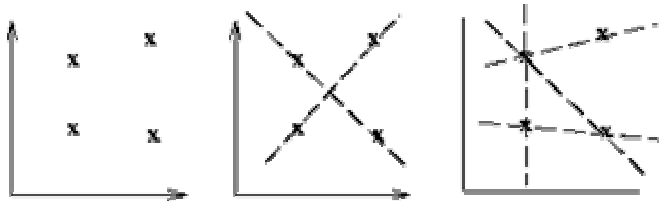


Gambar 4.7. Kernel pendeteksi garis

Transformasi Hough

Transformasi Hough adalah sebuah metode yang dapat digunakan untuk mengisolasi *feature* tertentu dalam sebuah citra. Metode transformasi Hough klasik biasanya digunakan untuk mendeteksi bentuk geometri yang dapat dispesifikasikan dalam bentuk parametrik seperti garis, lingkaran, elips, dan lain–lain.

Prinsip kerja metode transformasi Hough dalam mendeteksi garis adalah dengan mencari bentuk geometri yang paling sesuai dengan kumpulan titik pada citra. Untuk garis lurus, transformasi Hough akan mencari garis lurus yang melewati titik tepi pada citra. Gambar 4.8 menunjukkan beberapa kemungkinan solusi masalah ini. Kurangnya pengetahuan *a priori* mengenai jumlah segmen garis yang diinginkan dan ambiguitas mengenai apa yang menyatakan sebuah segmen garis menjadi masalah yang perlu dipertimbangkan.

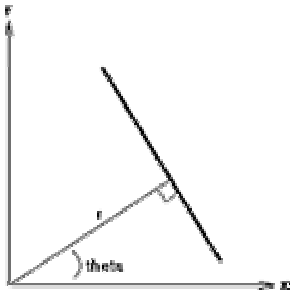


Gambar 4.8 Titik koordinat dan kemungkinan pencocokan garis.

Sebuah segmen garis dapat dinyatakan secara analitik dalam beberapa bentuk. Persamaan yang paling sesuai dalam menyatakan himpunan garis dengan menggunakan notasi parametrik atau normal adalah.

$$x\cos\theta + y\sin\theta = r \qquad (4.14)$$

Parameter *r* adalah panjang normal dari origin ke garis ini dan θ adalah orientasi *r* terhadap sumbu *x*. Untuk setiap titik (*x*,*y*) pada garis ini, *r* dan θ adalah konstan.



Gambar 4.9 Deskripsi parametrik sebuah garis.

Transformasi Hough akan mentransformasikan ruang spasial (*x*,*y*) ke dalam ruang Hough (*r*, θ). Transformasi Hough akan mencari semua kemungkinan nilai *r* dan θ bagi setiap titik tepi (*x*, *y*) dalam bidang citra. Semua titik yang berada pada suatu lintasan garis akan memiliki nilai (*r*, θ) yang sama.

Transformasi ini diimplementasikan dengan cara mengkuantisasi ruang parameter Hough ke dalam sejumlah interval tertentu atau sel akumulator. Ketika algoritma berjalan, setiap (x_i,y_i) ditransformasikan ke dalam kurva (r, θ) yang terdiskretisasi dan sel yang berada sepanjang kurva ini ditambahkan isinya. Puncak hasil dalam deretan akumulator menyatakan bukti kuat garis lurus dalam citra.

Prosedur yang sama dapat digunakan untuk mendeteksi bentuk – bentuk lain dengan deskripsi – deskripsi analitikal. Untuk bentuk lingkaran, persamaan parametriknya adalah :

$$(x-a)^2 + (y-b)^2 = r^2 \qquad (4.15)$$

dengan *a* dan *b* adalah koordinat pusat dari lingkaran dan *r* adalah radiusnya. Dalam kasus ini kompleksitas komputasi algoritma ini akan meningkat, karena jumlah parameter koordinat dan akumulator berdimensi 3. Secara umum komputasi dan ukuran deret akumulator meningkat secara polinomial dengan jumlah parameternya.

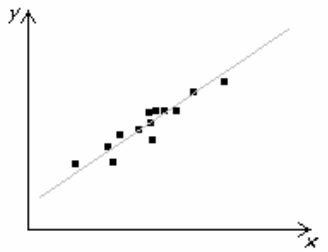
Metode pendeteksian menggunakan transformasi Hough mempunyai kelebihan dapat mendeteksi garis pada bentuk geometri kumpulan tepi yang terputus–putus. Transformasi Hough juga relatif tidak terlalu terpengaruh oleh *derau*. Parameter yang mempengaruhi performansi transformasi ini adalah kuantisasi parameter pada ruang Hough.

Masalah utama dalam transformasi Hough adalah dalam menentukan persamaan garis yang paling sesuai atau maksimum lokal pada ruang Hough. Suatu kumpulan tepi yang merepresentasikan suatu garis dapat menghasilkan beberapa garis dalam ruang Hough. Semakin tebal garis tepi tersebut, maka persamaan garis yang dihasilkan semakin banyak. Masalah ini dapat diatasi dengan menggunakan beberapa cara.

Salah satu cara adalah dengan menggunakan sebuah metode yang melibatkan proses *thresholding* dan kemudian menerapkan proses *thinning* untuk mengisolasi titik maksimum lokal dari daerah terang dalam ruang Hough atau deret akumulator citra tersebut.

Pendeteksian garis menggunakan regresi linier

Suatu kumpulan titik dapat diaproksimasikan sebagai garis lurus menggunakan metode regresi linier.



Gambar 4.10 Aproksimasi kumpulan titik dalam garis lurus

Persamaan garis lurus secara umum dapat dituliskan sebagai berikut.

$$y = ax + b \tag{4.16}$$

Persamaan ini akan mempunyai kelemahan jika nilai parameter $a \to \infty$. Dengan demikian apabila parameter $a \to \infty$ maka persamaan garis akan direpresentasikan oleh persamaan berikut.

$$x = c, \quad \text{dengan } a \to \infty \tag{4.17}$$

Parameter–parameter persamaan garis tersebut dihitung dengan persamaan–persamaan berikut.

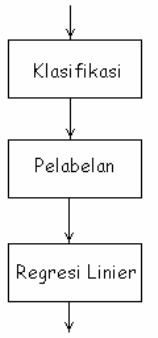
$$a = \frac{n\sum xy - \sum x \bullet \sum y}{n\sum x^2 - \left(\sum x\right)^2}$$

$$b = \frac{\sum y - a \bullet \sum x}{n} \qquad (4.18)$$

$$c = \frac{\sum x}{n}$$

Proses penghitungan parameter persamaan tersebut dilakukan dengan menggunakan semua data titik yang ada. Pada aplikasi pemrosesan citra, data titik adalah tepi hasil pendeteksian tepi.

Metode ini mempunyai kelemahan dimana untuk semua data tepi akan diaproksimasikan ke dalam satu persamaan garis. Proses ini akan salah mengaproksimasikan persamaan garis jika dalam citra terdapat beberapa kumpulan tepi yang merepresentasikan lebih dari satu garis. Dengan demikian, pada metode ini perlu dilakukan beberapa proses awal, yaitu mengelompokan tepi yang menggambarkan sebuah garis lurus.



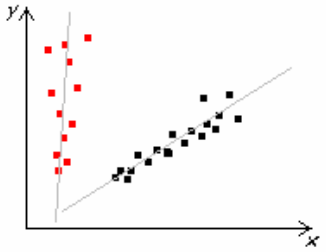
Gambar 4.11 Proses pendeteksian tepi menggunakan regresi linier

Proses pendeteksian garis menggunakan metode ini terdiri atas tiga bagian, yaitu :

- Klasifikasi atau segmentasi.
  Citra hasil proses pendeteksian tepi dikelompokkan berdasarkan arah atau orientasinya. Tepi yang mempunyai arah yang sama dikelompokkan menjadi satu kelompok.
- Pelabelan (*Labelling*)
  Semua kelompok tepi hasil proses segmentasi dikelompokkan kembali dalam beberapa kelompok yang merepresentasikan garis lurus. Tepi yang mempunyai arah sama dan mempunyai letak yang berdekatan dikelompokkan menjadi satu kelompok serta diberi label yang unik.
- Regresi linier
  Pada setiap kelompok hasil proses pelabelan diaproksimasikan suatu garis lurus dengan menggunakan metode regresi linier.

Pada proses ini, setiap kumpulan tepi yang merepresentasikan suatu garis lurus akan langsung diaproksimasi satu persamaan garis. Dengan demikian, proses pendeteksian ini akan mempunyai komputasi yang lebih cepat daripada metode transformasi Hough. Jumlah garis lurus yang akan diaproksimasi sangat bergantung pada proses pengelompokkan tepi.

Proses yang paling penting dalam metode ini adalah proses pengelompokan tepi ke dalam sejumlah kelompok yang merepresentasikan sebuah garis lurus.



Gambar 4.12 Regresi linier dua buah kelompok tepi

Proses segmentasi tepi berdasarkan arah tepi dapat dilakukan dengan menggunakan pendeteksi tepi *Compass Mask*. Dengan demikian, semua tepi hasil pendeteksian tepi langsung dikelompokkan berdasarkan arahnya. Setelah itu dilakukan proses segmentasi pada masing–masing kelompok arah. Hal ini dilakukan untuk mengantisipasi adanya beberapa garis yang mempunyai arah atau gradien yang sama. Pengelompokan ini dapat dilakukan dengan cara melihat jarak relatif antar piksel yang berdekatan. Piksel-piksel yang berdekatan dikelompokkan menjadi satu kelompok. Masing–masing kelompok kemudian diberi label yang unik untuk membedakan kelompok yang satu dengan yang lainnya. Setelah itu, baru dilakukan proses regresi linier untuk mendapatkan persamaan garis untuk setiap kelompok.

4.1.2. Pengenalan objek menggunakan aplikasi ray tracing

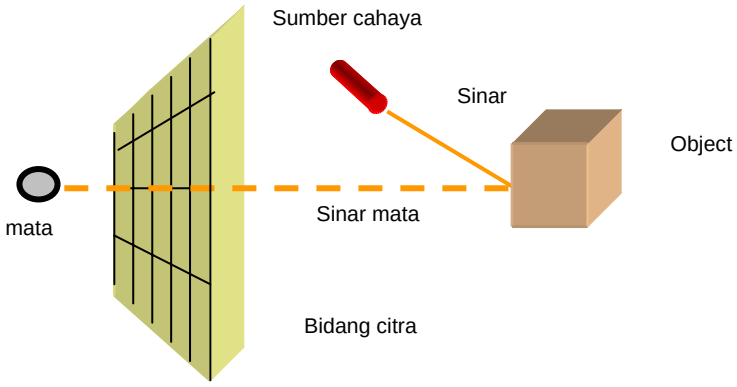
Dalam pasal ini akan dijelaskan konsep pengenalan objek dengan menggunakan aplikasi *ray tracing* dan beberapa dasar pemakaian fungsi untuk *ray tracing*. Beberapa rincian yang akan diuraikan sekilas adalah definisi *ray tracing*, penggunaan suatu model kamera, penelusuran cahaya dengan fungsi, pembangkitan cahaya, penghitungan cahaya datang, penentuan cahaya yang memotong objek, penerapan model pencahayaan, penentuan objek tembus pandang, dan pemakaian fungsi terhadap model pencahayaan.

Definisi *ray tracing*

Ray tracing merupakan teknik pencahayaan global yang mengenali dan merender suatu gambar pada basis piksel demi piksel. Ide *ray tracing* adalah menelusuri cahaya untuk tiap piksel, dari mata atau titik pandang lewat piksel dan menuju ke tempat kejadian atau *scene*. Metode untuk

menelusuri sebuah cahaya dari sumber cahaya dan merambat ke mata atau titik pandang, disebut *ray tracing* arah mundur. Ada sejumlah cahaya yang tidak hanya berasal dari sumber cahaya dan ini menimbulkan masalah. Oleh karena itu, dengan merunut cahaya yang arahnya berlawanan dengan penyebaran cahaya, saat suatu objek ditemukan, dihitunglah kontribusi tiap sumber cahaya yang tampak dari titik perpotongan. Garis cahaya yang merambat ke dalam lingkungan sesuai dengan model pemantulan dan perpotongan hingga diabaikan kontribusi yang mungkin berasal dari objek yang lebih jauh. Nilai akhir intensitas dihitung sebagai jumlah seluruh kontribusi perpotongan. Untuk menciptakan suatu citra layar dari titik pandang yang berbeda dibutuhkan pengulangan seluruh algoritma; tak ada hasil dari tempat kejadian atau *scene* yang dapat digunakan.

Di bawah ini, akan diberikan suatu gambar sderhana untuk mengilustrasikan suatu proses yang menggunakan ray tracing untuk menentukan permukaan yang diinginkan. Suatu titik mata (*eye*) dan bidang citra (*view plane*) yang terpikselkan, suatu sinar (*eye ray*) ditembakkan dari titik mata, lewat sebuah piksel, dan menuju scene. Sinar yang memotong objek adalah objek yang kelihatan pada piksel bidang citra (*view plane*) dan piksel itu ditandai dengan warna objek. Gambarannya, sebagai berikut :

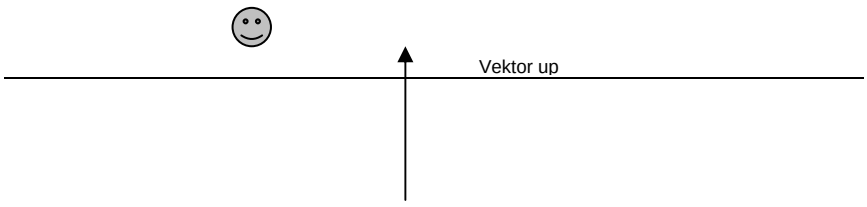


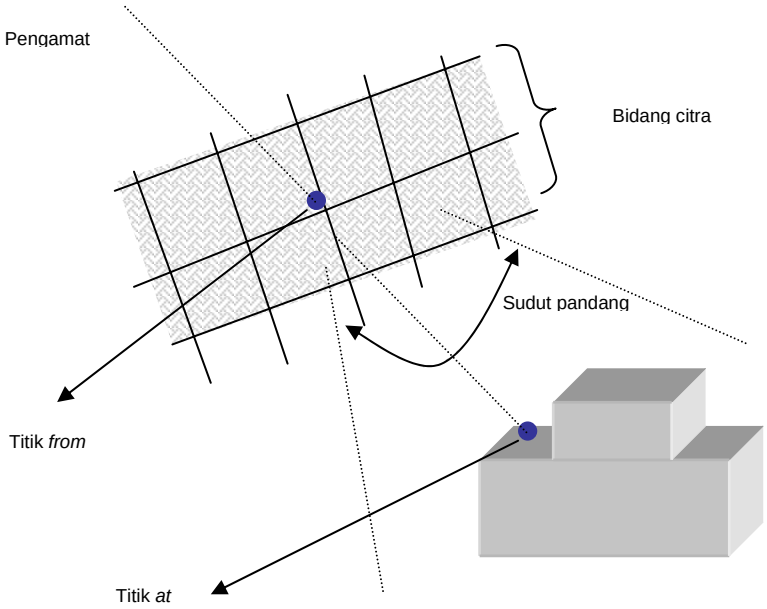
Gambar 4.13 Deskripsi sederhana *ray tracing*

Sinar yang ditembakkan pada tiap sumber cahaya dari titik tempat sinar mata dan objek berpotongan. Sinar bayangan membantu dalam menentukan intensitas dan warna dari titik perpotongan objek atau sinar mata. Oleh karena itu, warna dan intensitas diberikan ke piksel. Proses ini dilakukan terhadap piksel dalam bidang citra. Saat terpenuhi, bidang citra mendapatkan gambar dari tempat kejadian dengan resolusi yang ditentukan oleh pikselisasi bidang citra. Energi cahaya yang datang ke permukaan yang bukan kaca sempurna dipantulkan secara acak (pemantulan yang tidak sederhana). Untuk permukaan spekular, arah penyebaran adalah variabel acak sebuah distribusi yang bergantung pada sudut datang dan sifat permukaan. Pada tiap-tiap perpotongan, suatu sinar pantul ditembakkan dalam arah yang ditentukan dengan menyampling distribusi yang digunakan untuk memodelkan penyebaran secara fisik.

Menggunakan suatu model kamera

Ray tracing menggunakan pemodelan kamera yang ditujukan untuk menentukan bagaimana memproyeksikan dengan inisialisasi sinar berarah mundur terhadap *interface* komputer, seperti yang akan diperlihatkan pada gambar 4.14. Komponen terdiri atas titik *from*, titik pengamat, titik *at*, bidang citra, sudut pengamat, kamera objek, dan vektor *up*. Titik *from* merupakan lokasi kamera atau pengamat, titik *at* adalah tempat kamera yang mengarah di depan objek. Bidang citra, secara konsep terletak antara pengamat dan model dunia, yang berisi citra yang dibuat. Vektor *up* menentukan bagaimana bidang citra diarahkan mengenai kamera. Yang terakhir adalah dunia yang diambil dalam satu gambar yang tergantung pada sudut pengamat.





Gambar 4.14 Pemodelan *ray tracing* dengan sebuah kamera

Citra, dari penjelasan sebelumnya, dibuat dari titik-titik kecil atau piksel. Resolusi citra bergantung pada berapa banyak piksel dan warna yang digunakan, dan keduanya dapat diprogram.

Aspek penting dari model kamera adalah memberikan arsiran (lebih jauh mengenai render ataupun arsiran ada pada Bab 5) objek yang menggunakan proyeksi perspektif dan memperoleh rasa kedalaman pada citra dua dimensi. Proyeksi perspekstif membuat objek yang muncul terputus-putus (terdistorsi) saat mendekati pengamat, dan garis paralel berkumpul (konvergen) saat menjauh. Sejumlah distorsi berhubungan dengan sudut pandang. Umumnya semakin benar sudut pandang, semakin besar pula distorsi.

Ray tracing memproyeksikan sinar cahaya dari pengamat lewat tiap piksel di dalam citra yang menggunakan kekuatan model kamera. *Ray tracer* (pengusut sinar) menerapkan piksel terhadap warna objek yang sinarnya mengenai objek tersebut. Suatu *ray tracer* me-*render* citra apa pun di dalam model dunia. Untuk me-*render* atau memberi arsiran pada layar komputer yang realistik, harus diperhitungkan pemantulan, bayangan, pencahayaan, dan bermacam-macam keberadaan permukaan objek.

Kumpulan konstanta menggambarkan tiap objek. Konstanta menetapkan ukuran, lokasi, dan permukaan objek seperti halus dan bersinar, warna dan *hue*-nya, dan sebagainya. Konstanta ini dimasukkan ke dalam persamaan yang menentukan tipe khusus objek. Sebagai contoh, *ray tracer* memberikan persamaan untuk memperagakan suatu segi tiga, elips, silinder, dan sebagainya.

Model-model pencahayaan

Model iluminasi mengacu pada berbagai tipe sumber cahaya yang dapat disimulasi oleh grafik komputer. Grafik komputer berusaha memodelkan bermacam-macam tipe sumber cahaya yang dapat ditemukan pada dunia nyata. Sumber cahaya dapat dikelompokkan ke dalam beberapa tipe, yaitu:

a. sumber cahaya titik
b. sumber cahaya berarah (*directional*)
c. sumber cahaya setempat

Hal lain yang dianggap penting pada model pencahayaan adalah :

- Intensitas menurun dengan jarak
- Keseimbangan level iluminasi
- Bayangan

Sumber cahaya titik merupakan sumber penerangan yang paling sederhana untuk pemodelan, karena itu kita harus meletakkan posisinya dalam ruang bersamaan dengan intensitas dan warnanya. Sumber cahaya yang berarah dapat dikenali dari warna, intensitas, dan arahnya. Contohnya sinar cahaya yang tiba pada sebidang kecil tanah di atas permukaan bumi, akan kelihatan paralel dan memiliki intensitas yang sama. Sudut antara vektor dan permukaan normal digunakan untuk menentukan kuantitas energi cahaya. Sumber cahaya *spot* (setempat) mensimulasikan tingkah lakunya yang menciptakan pancaran cahaya yang dikendalikan dalam bentuk *cone* (kerucut). Kisaran minimum parameter dibutuhkan untuk memodelkan sumber cahaya ini, terdiri atas posisi, intensitas, warna, dan sudut setempat.

Model pemantulan

Tiga tipe pemantulan cahaya terdiri atas *ambient, diffuse*, dan *specular*. Cahaya *ambient* mensimulasikan level cahaya konstan yang dapat menyebabkan banyak pantulan.

Cahaya *diffuse* diciptakan oleh permukaan yang memiliki kekasaran sehingga menyebabkan cahaya yang datang dipantulkan sama ke semua arah.

Cahaya *speculer* mengacu pada cahaya yang dipantulkan oleh permukaan licin dan menciptakan suatu sinar terang dari sumber cahaya yang teriluminasi dan menerangi lingkungan sekitarnya. Warna objek ditentukan oleh radiasi penerangan (*iluminasi*) bagian-bagian yang diserap serta dipantulkan.

Menentukan sinar yang memotong objek

Untuk menentukan jatuhnya sinar *ray tracing* pada objek yang terdekat, diperkirakan akan diperlukan waktu yang relatif tidak sedikit. Pengujian pada *ray tracing* dilakukan dengan mendapatkan objek yang tepat berpotongan dengan sinar. *Ray tracer* pada beberapa fungsi berusaha memecahkan masalah ini dengan memanfaatkan perhitungan matematis dan pemilihan objek yang tepat. Contoh yang paling sederhana adalah bidang. Setiap titik objek dapat ditentukan dari persamaannya. *Ray tracer* mensubtitusi persamaan sinar ke dalam persamaan objek dan mencari pemecahan permasalahannya. Jika ada, maka koordinatnya merupakan tempat sinar memotong objek. Jika tidak, sinar tidak mengenai objek dan sisa objek lainnya harus diuji lagi.

4.2. PENJEJAKAN OBJEK

Penjejakan objek adalah satu dari sekian banyak pembahasan yang ada pada sistem *vision* yang aktif. Kompleksitas struktur masalah gerak dapat dikurangi dengan penjejakan (*tracking*) – yang menghubungkan sebuah titik pada objek yang bergerak dan pengukuran aliran optik dalam suatu *frame*; titik tersebut dapat mewakili sejumlah titik lain pada objek yang bergerak itu. Hal ini berkaitan erat dengan keuntungan dari paralaks ataupun pergerakan relatif dalam efek yang dipindahkan dalam pengukuran kedalaman kecepatan rotasi yang diselidiki. Lebih jauh lagi kontinuitas sementara dari fitur yang dijejak akan lebih akurat bila digunakan juga sebuah binokuler stereo. Tapi, tanpa perhitungan komputasi akan timbul juga permasalahan yang kompleks.

Aplikasi penjejakan ini antara lain terdapat pada alat survei automatik, monitor trafik, koordinasi pada robot (dalam hal ini mobil robot), dan pengendali jalan.

Sebelum melangkah lebih jauh ke penjejakan objek, ada baiknya diberikan terlebih dahulu beberapa definisi yang ada pada sistem *vision* aktif yang kelak akan berguna pada pembahasan berikutnya. Hal-hal lain yang terdapat pada sistem vision aktif adalah sebagaimana yang diuraikan berikut ini.

- **Struktur gerak yang dikendalikan**

Selama dasawarsa terakhir ini, teori tentang komputasi egomotion dan struktur permukaan aliran optik telah dikembangkan, antara lain oleh:

- *Koenderink dan Van doorn* (tahun 1975 dan 1978)
- *Ullman* (tahun 1979)
- *Longuet – Higgins dan Pradzny* (tahun 1980)
- *Huang dan Tsai* (tahun 1981)
- *Waxman dan Ullman* (tahun 1985)
- *Maybank* (tahun 1985)
- *Faugeras* (tahun 1987) ; dan
- *Murray dan Buxton* (tahun 1990)

Kelihatannya sangat alamiah untuk mengharapkan bahwa seandainya analisis citra tunggal itu berat, maka analisis citra yang berurutan seharusnya lebih berat. Tetapi, konsekuensi kekayaan informasi geometrik dalam aliran citra membuat kondisi aktual yang diperoleh robot melalui sistem visonnya lebih riil dari sistem vision biasa. Jika sebuah robot dapat melakukannya, maka untuk sementara dia akan bergerak dengan ‘matanya’. Hal tersebut sering terjadi pada robot untuk membangun gerakan untuk menginduksi pergerakan citra.

- **Perhatian terfokus (f*ocussed attention*)**

Mekanisme atensi telah lama diketahui dalam pandangan manusia (Treisman dan Gelade, 1980) dan sangat beralasan sekali untuk mengharapkan bahwa hal-hal tersebut akan terpakai pada sistem mesin *vision*. Mekanisme kemandirian level rendah untuk penjejakan fitur level rendah membebaskan proses pada level yang lebih tinggi untuk menganalisis fitur yang relevan dengan perintah yang khusus, pembebasan *switching* di antara data citra yang dibutuhkan. *Switching* tersebut kelihatan lebih efektif pada *vision* yang binokuler yang penjejakannya akan memberi ukuran yang pas dengan ruangan tiga dimensi. Pengendali fiksasi inilah yang akan digunakan sebagai bentuk fungisional 'mata' robot.

- **Prediksi**

Alasan yang sangat memaksa pada sebuah sekuens citra yang temporal untuk diproses adalah keharusan untuk memproses lebih dari satu citra yang dalam sekuens tersebut tidak independen. Fitur yang ditemukan dalam satu citra berhubungan satu sama lain dengan eratnya. Kelemahan dari hubungan ini adalah bentuk sementara yang kontinu yang telah disebutkan di atas tadi. Bentuk yang kuat dapat dibuat jika tersedia model prior dari gerak tiga dimensi. Filter kalman adalah mekanisme klasik untuk mengerjakan hal tersebut, dan mampu mengkombinasikan model sementara yang deterministik dengan ketidakpastian yang stokastik.

- **Strategi sensing**

Kesempatan utama bagi pengamat yang aktif adalah merencanakan sensor aksi dalam waktu nyata (*real-time*), yang bertambah secara kumulatif dengan kebutuhan akan informasi yang luar biasa. Tujuan tahapan-tahapan *sensing* adalah untuk memaksimalkan informasi yang diperlukan selama tahapan berlangsung dengan tujuan akhir pada pasca proses citra. Pengukuran berdasarkan temuan Bayes dapat digunakan

untuk rencana *sensing* ini, yang mengikutsertakan sensor yang sederhana (Cameron and Durrant White, 1990).

Pembahasan mengenai penjejakan objek ini dibagi-bagi atas:

1. **Metode Kalman snakes**
2. **Kontur dinamik**
3. ***Deformable templates***
4. **Penjejakan model dan objek tiga dimensi pejal maupun non pejal**
5. **Metode asosiasi data dengan sistem penjejakan**
6. **Geometri gerakan yang visual**
7. **Perencanaan gerakan menggunakan divergensi citra dan deformasi**
8. **Navigasi lokal adaptif**
9. **Sistem *vision* tiga dimensi paralel**

4.2.1. METODE KALMAN SNAKES

Snakes (termasuk model kontrol yang aktif) adalah bentuk deformabel yang telah banyak digunakan dalam berbagai aplikasi analisis citra, termasuk penjejakan citra berdasarkan objek pejal maupun yang tidak pejal. Kontur yang deformabel dan generalisasi bentuk multidimensi biasanya mengacu pada bentuk objek dan pergerakan yang direpresentasi oleh persamaan diferensial integral dari elastodinamika Lagrange. Persamaan *snakes* dari gerak menyediakan mekanisme penjejakan yang digerakkan oleh turunan gaya dari citra yang bergantung pada waktu.

Snakes diperkenalkan pada tahun 1987 oleh Terzopoulos.Pada dasarnya ide snakes mengambil konsep yang hampir sama dengan pendeteksian sudut dan kurva, yang sebelumnya dibuat oleh Montanari (tahun 1972) dan Martelli (tahun 1972). Model *snakes* akan men-

generalisasi banyak notasi melalui penggunaan model elastodinamika dan gaya yang akan diaplikasikan.

Pada pembahasan mengenai Kalman *snakes* ini, banyak pendekatan dibuat berdasarkan sistem titik dinamik dan hubungannya dengan teori estimasi. Ada beberapa jenis varian yang terkenal, termasuk *snakes* yang dikenalkan oleh Fourier, elemen hingga, dan representasi diskret. *Snakes* dan variannya telah banyak dipakai untuk citra yang statik yang memerlukan sudut, kurva, dan pendeteksian *boundary*, juga termasuk proses segmentasi (yang ada pada pembahasan sebelumnya) dan proses skeletonian. Sejak awalnya, *snakes* diaplikasikan untuk mendeteksi struktur biologi yang kompleks, dengan banyak penggunaan pada interpretasi citra biomedik.

Ide penjejakan objek yang bergantung pada waktu menggunakan *snakes* yang diusulkan pada tahun 1987 yang pada waktu itu diaplikasikan untuk menjejak bentuk mulut pembicara melalui gaya dinamik yang dibangun dari citra. Aplikasi dan pemrograman lanjut dari *snakes* yang berbentuk *multiple snakes* adalah penjejakan artikulasi *facial*. *Snakes* tertutup juga dipakai untuk mengenali permukaan sel yang sering berubah bentuk menjalani pseudopodia. Pada tahun 1990 sampai 1991, *snakes* waktu nyata digunakan untuk menjejak kontur yang terdapat pada objek tiga dimensi yang berasal dari penangkapan citra melalui sebuah kamera pada lengan robot yang bergerak. Semua aplikasi di atas telah menunjukkan kepada dunia ilmu pengetahuan pemakaian *snakes* yang sangat berguna untuk menjejak objek pejal maupun yang tidak pejal.

Pendekatan penjejakan lainnya pada konteks *vision* aktif sering didasarkan pada teori filter Kalman. Pendekatan filter Kalman pada awalnya digunakan untuk mengestimasi pertambahan yang bertahap dari

gerak objek pejal dan juga untuk menjejak fitur pejal seperti garis dan banyak titik.

Snakes merupakan bentuk planar yang bergerak di bawah pengaruh gaya citra. Kita dapat mendefinisikan bahwa *snakes* juga merupakan kontur yang dapat berubah bentuk dengan membangun energi deformasi/bentukan ($\varepsilon_s(v)$) yang sesuai dan tepat, dengan v merepresentasi kontur sebagai hasil pemetaan daerah asal dari pemetaan parametric $s \in [0,1]$ ke dalam permukaan citra $\Re^{2\,3}$. Komponen pemetaan v(s) = (x(s),y(s)) adalah koordinat fungsi kontur. Sebuah *snakes* juga berfungsi untuk meminimalkan energi.

$\varepsilon(v) = \varepsilon_s(v) + \mathrm{P}(v)$, sedangkan untuk sebuah *snakes* yang sederhana (dalam bentuk linier), energi deformasi internalnya adalah

$$\varepsilon_s = \int_0^1 \omega_1(s)\left|v_s\right|^2 + w_2(s)\left|v_{ss}\right|^2 \, ds$$

Kalman *snakes* adalah metode pertama penjejakan yang dipakai pada mobil robot ini. Kita memakai Kalman *snakes* dengan mengembangkan persamaan *snakes* dari gerak sebagai sebuah model sistem dengan filter Kalman yang kontinu. Untuk sebuah pengertian yang singkat, persamaan yang fundamental dari gerak *snakes* dapat dirangkumkan dalam bentuk berikut:

$\frac{du}{dt} = Fu + g$, yang berkaitan dengan filter Kalman.

Pada kondisi *snakes* yang *massless* (kasus M = 0), dapat kita peroleh sebuah persamaan lain, yaitu

$\frac{d}{dt}u = -C^{-1}Ku + C^{-1}f$, yang merupakan bentuk standar. Untuk menyempurnakan bentukan persamaan orde satu di atas, diperlukan kecepatan nodal $\frac{du}{dt} = \dot{u}$ yang menjelaskan posisi u dengan memperoleh sebuah set pasangan persamaan orde satu sebagai berikut :

$$\frac{d}{dt}\begin{bmatrix}\dot{u}\\ u\end{bmatrix} = \begin{bmatrix}-M^{-1}C & -M^{-1}K\\ I & 0\end{bmatrix}\begin{bmatrix}\dot{u}\\ u\end{bmatrix} + \begin{bmatrix}M^{-1}\\ 0\end{bmatrix} \quad , \quad \text{yang}$$

merupakan bentuk yang lebih komplet daripada persamaan sistem dinamik sebelumnya.

Pada tahapan selanjutnya, kita dapat menurunkan aproksimasi dari persamaan Kalman *snakes* dengan mudah dengan mengasumsikan bahwa besar invers dari matriks kovarian S dapat dipartisi ke dalam elemen internal yang bergantung pada waktu dengan komponen K dan komponen diagonal S'

$$S(t) = K_s + S'(t).$$

Kita dapat mengatur K_s = K, sebagai matriks *snakes* yang *stiff*. Kemudian, kita memakai persamaan Riccati yang erat hubungannya dengan filter Kalman langsung ke nilai S' tadi dan mengabaikan segala jenis kondisi diagonal yang akan timbul. Persamaan berikutnya yang muncul dari akibat tersebut adalah

$$\dot{\hat{u}} = F\hat{u} + (K_s + S')^{-1}H^{\mathrm{T}}R^{-1}(d - H\hat{u}),$$

Persamaan yang merupakan penambahan dari persamaan sebelumnya itu mengubah estimasi keadaan sesuai dengan tingkat kedinamikan sistem yang digambarkan dengan **F** dan juga sesuai dengan differensi penyaringan di antara berbagai sampel *state*, **Hu** dan juga nilai

data **d** (yang dapat digantikan dengan gaya eksternal lainnya). Faktor penguatan Kalman terdiri atas sejumlah komponen R^{-1} yang secara proporsional diinverskan terhadap derau dalam bentuk pengukuran yang baru. Sebagai catatan terhadap **Metode penjejakan dengan Kalman *Snakes*** ini adalah bahwa kelas model yang dapat berubah bentuk dikenal dengan sebutan *snakes*, yang telah terbukti sangat berguna dalam beberapa perintah penjejakan objek.

Ketika metode ini diaplikasikan pada citra yang statik, sifat fisik model akan bermakna banyak untuk mendukung interaksi yang terdapat antarbenda. Bentuk alami yang dinamik dari *snakes* berdampak lebih besar lagi ketika diaplikasikan pada pencitraan yang bergantung pada waktu. Jika pendekatan probabilitas yang digunakan, khususnya teknik yang banyak dikenal seperti estimasi Bayes dan filter Kalman, akan diperoleh pengukuran dengan derau yang terukur, juga informasi integrasi dari beberapa sumber.

4.2.2. KONTUR DINAMIK

Sebelumnya, perancangan kontur dinamik lebih dikenal pada saat menjalankan video dengan kecepatan rata-rata. Kontur dinamik adalah salah satu metode terkenal yang berkembang dengan sangat cepat yang pada hakikatnya mempunyai prinsip yang hampir sama dengan penjejakan dengan Kalman Snakes yang telah dijelaskan di atas. Perbedaannya hanya terletak pada tujuan akhir yang ingin dicapai. Pada kontur dinamik, penekanan tujuan berakhir pada pencapaian kecepatan yang sangat tinggi untuk penjejakan *real-time* yang dilakukan bersama-sama dengan bentuk yang terpilih. Kontur dinamik bersifat elastis, yang juga sama dengan *snakes*, tapi secara parameter menggunakan *B-splines*. Parameter yang kecil merepresentasi faktor efisien dari persamaan *state-space* (ruang keadaan) dari kontur. Ketika hal tersebut dikombinasikan dengan komputasi

paralel dalam bentuk yang lebih kecil, jaringan transputer serta kinerja penjejakan *video-rate* dapat dicapai untuk kontur dengan simultan.

Pada kontur yang berpasangan, *snakes* yang tidak mengalami tekanan akan kembali ke pergerakan yang semula. Kemudian, langsung membentuk konfigurasi garis yang sesuai. Dalam hal ini, kontur dinamik, jika tidak mengalami tekanan, akan mendapatkan derajat kebebasan yang penuh dengan *state* yang unik. Seringkali dalam penjejakan diinginkan bekerja pada tendensi yang kuat kepada konfigurasi biasa. Hal tersebut hanya dapat dilakukan ketika target bentuk bertambah dengan aproksimasi yang diinginkan. Pencapaian kondisi hanya dapat terjadi bila kontur *B-spline* dipasangkan dengan *template B-spline* yang pejal. Efek yang didapat dari pengkombinasian ini adalah sifat elastis *snakes* dengan banyak parameter *template*. *Template* tersebut dapat dikatakan '*belajar'* berdasarkan kontur dinamik yang terkunci pada fitur target yang representatif. Selanjutnya, kontur ini dinamakan '*frozen'* dan menjadi pasangan *template* yang secara elastis menjadi kontur dinamik yang baru.

Dalam penjejakan bentuk dan kontur yang dinamis, perubahan kinerja dapat direalisasikan ketika semua model yang terintegrasi, terdistribusi secara massal dan mediumnya *viscous*. Persamaan berikutnya yang menjadi pembahasan pada bab ini sama dengan yang kita pakai pada penjejakan dengan Kalman *snakes*, yaitu formulasi Lagrange untuk kontur dinamik. Pasangan kontur dinamik dengan *template*-nya dan fitur citra ditentukan oleh elastisitas dan paramater viskositas.

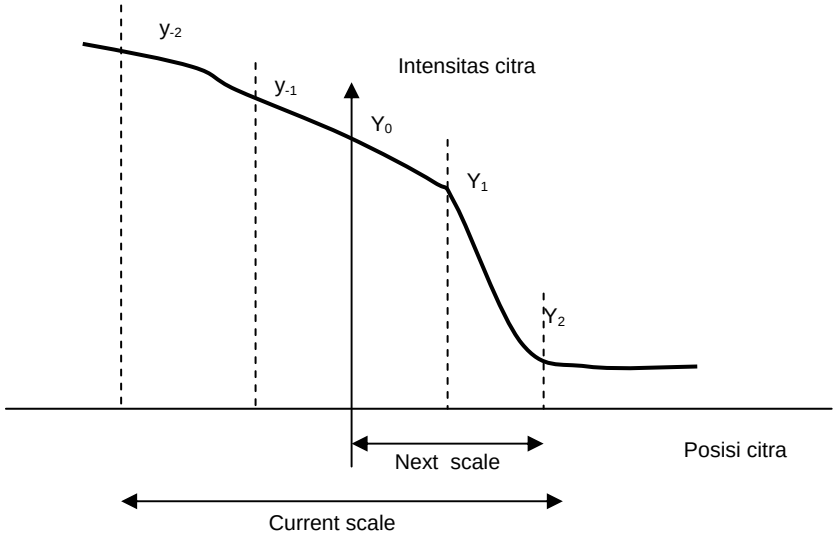
Dalam sistem yang lebih dinamik titik-titik yang terdapat sepanjang kontur *B-spline* diprogram untuk memiliki afinitas sebagai tambahan buat citra seperti *brightness* ataupun gradien intensitas tingginya. Afinitas ini, berpasangan dengan kontinuitas intrinsik dan elastisitas *B-spline*, sekaligus menghasilkan penjejakan yang sifatnya lebih fleksibel. Prinsip ini juga

digunakan pada kerja penjejakan dengan *snakes*. Ide utama penjejakan kontur dinamik adalah menekankan pentingnya konvolusi sebagai arti kecerdasan dalam teknik *blur* untuk mengendalikan perputaran spasial benda. Hal ini mengimplikasikan anggapan tenaga komputasi seandainya penjejakan dalam waktu nyata dicapai untuk selang waktu tertentu. Bagaimana pun kita telah membuktikan bahwa teknik blur tidak bukanlah faktor yang krusial dalam mencapai kinerja pada waktu nyata yang diinginkan.

Selanjutnya, pencarian lintasan dimungkinkan dalam bentuk garis-garis yang paralel pada arah dan tujuan yang tetap pada citra. Alternatif lain adalah melakukan penjejakan sepanjang normal ke *x(s)*. Alternatif ini memiliki keuntungan, karena pada kenyataannya tidak ada arah dan tujuan yang tetap pada penjejakan yang dipilih. Sebaliknya, alternatif ini menyebabkan kerugian dalam kestabilan penjejakan karena munculnya kontur usang berulangkali yang menempati titik-titik penting pada citra, juga sudut-sudut yang lain. Implementasi secara praktis dari *search normal* telah berhasil dilaksanakan dalam menstabilkan sepasang mekanisme kontur. Alternatif berikutnya adalah mencari titik yang tetap dengan arah radial. Proses tersebut membolehkan tumbuhnya kontur tertutup secara radial keluar dari titik yang diketahui sebagai bagian dalam dari bagian tambahan citra yang tertutup. Akibatnya, secara teori masih saja timbul masalah, tapi dalam kenyataan praktisnya tidak timbul masalah yang terlalu rumit sehingga dianggap masih dalam batas kerja yang memuaskan.

Dalam penjejakan dengan kontur dinamik ini, kita beranggapan bahwa metode *search* dimaksudkan untuk mendeteksi semua bentuk gerak dari kontur dan bergerak sepanjang garis-garis yang paralel. Strategi tambahan *search* merupakan bentuk yang rekursif, yang beroperasi dalam suatu *coarse-to-fine fashion.* Proses tersebut bekerja dengan mengaproksimasi data citra yang berbeda dengan jumlah yang terhingga

untuk kemudian dibuat penskalaan yang mendetail. Pada masing-masing skala, aproksimasi *finite difference* ke dalam gradien citra dibuat sedemikian rupa pada *search point* dan *point* pada bagian yang berbeda. *Search point* yang baru kemudian menjadi gradien yang mengalami kalkulasi terbesar. Diagram ditunjukkan pada gambar 4.15



Gambar 4.15 Perbandingan antara posisi citra dan intensitasnya

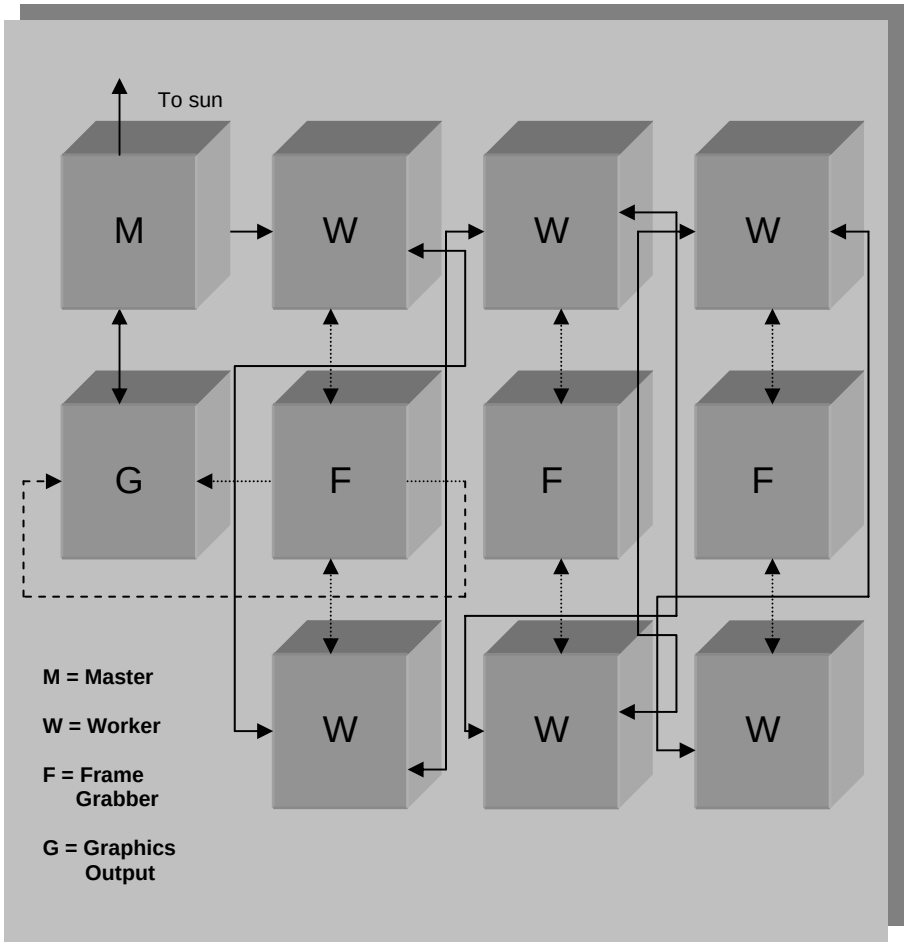
Untuk implementasi paralel dari kontur dinamik, persamaan gerak yang ada diintegrasikan dengan menggunakan skema implisit Euler untuk kecepatan pada jaringan transputer. Konfigurasi jaringan dapat dilihat pada Gambar 4.16.

Kalkulasi Bulk juga diikutsertakan dalam evaluasi dengan kondisi Q_f. Di sini, masing-masing *span* memberikan kontribusi ke dalam estimasi *B-spline* dari fitur posisi. Langkah-langkah yang terdapat di dalam estimasi ini untuk *span* ke – i, adalah:

- Lakukan sampel pada posisi yang diprediksi dalam B-spline pada sejumlah *point* sepanjang *span*, x_i. Prediksi dibuat berdasarkan

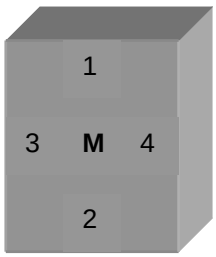
kecepatan kontur, yang telah dikalkulasikan sepanjang integrasi Euler dan pada interval antara *frame* dengan *frame*.

- Buatlah masing-masing *point* bergerak sepanjang vektor normal *spline* sampai gradien citra mencapai maksimum lokal, p_i.
- Kalikan bentuk-bentuk matriks untuk menghasilkan *span* yang diinginkan.



Gambar 4.16 Konfigurasi jaringan transputer. *Master* mengendalikan *scheduling*, sedangkan *workers* meng*up-date span*

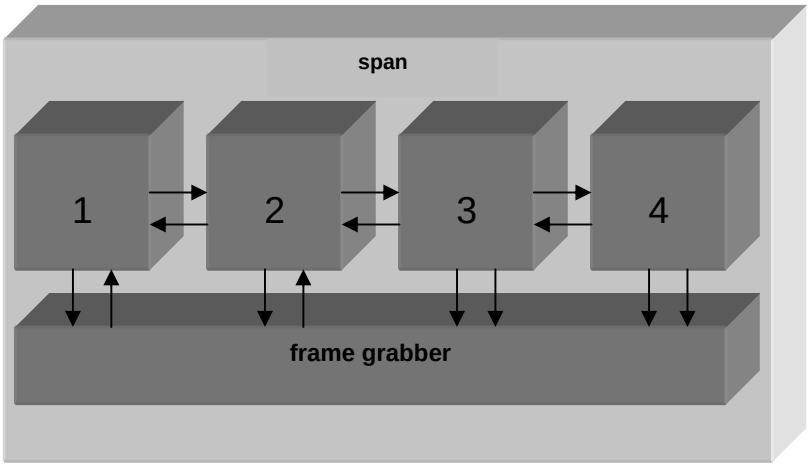
Keterangan : satu buah blok terdiri dari komposisi yang sama, yakni



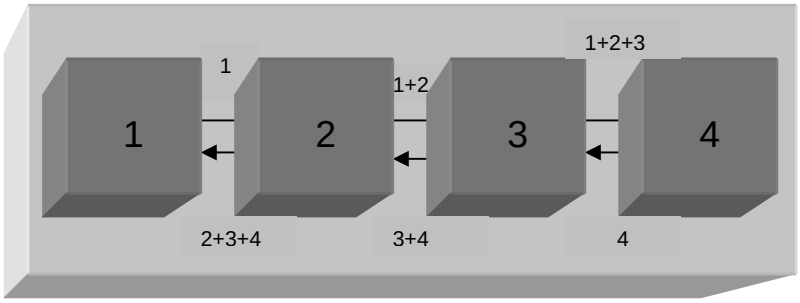
Analisis persamaan dinamik tersebut menyarankan paralelisasi berbasis *span*. Untuk model transputer konkurensi, proses tersebut menerjemahkan ke dalam banyak proses yang terpisah. Gambar 4.17 menunjukkan proses *comprising* dari empat buah kontur *span*, dan akan mengindikasikan bagaimana hasil akhir membutuhkan perhitungan Q_f yang diimplementasi dengan dua *running sums*.

Kontur dilokasikan pada *worker transputer* pada jangka waktu tertentu. *Span* individu berkomunikasi dengan kontribusi Q_f ke dalam kontur lain pada masing-masing langkah Euler. Dengan enam buah *worker transputer*, langkah-langkah Euler dapat ditunjukkan dalam *frame rate* (25 Hz). Lebar pita (*bandwidth)* dari hubungan transputer kurang cukup untuk menopang komunikasi data citra yang *real time*. Untuk mengatasi masalah ini, digunakan tiga buah *frame grabber* yang diletakkan terpisah, dan dikendalikan dari masukan video yang umum. Jaringan transputer diletakkan pada sebuah *personal computer* yang menggunakan *workstation* UNIX, yang diakses melalui rutin pada sebuah *library* yang ditulis dengan menggunakan bahasa pemrograman tingkat menengah, yakni bahasa C (penjelasan singkat mengenai bahasa C ada pada lampiran buku ini).

Skema berikut ini menunjukan empat buah *span* kontur dinamik dan hubungannya dengan *frame grabber*.



Gambar 4.17 Proses comprising empat buah span dan *frame grabber*



Gambar 4.18 Hasill akhir dari proses comprising (Q_f)

4.2.3. *DEFORMABLE TEMPLATES*

Sebuah mobil robot yang menggunakan sistem *active vision* memerlukan pengidentifikasian fitur yang terdapat pada dunia (*world)*. Dalam sistem yang terdapat sekarang ini, fitur-fitur yang ada mungkin mempunyai bentuk yang sederhana (walaupun tidak begitu penting untuk mudah dideteksi), seperti lintasan jalan. Pada sistem yang akan terdapat pada masa depan, berbagai fitur tersebut akan lebih kompleks dan rumit. Sebuah atau beberapa teori dibutuhkan untuk mendeteksi fitur-fitur tersebut, yang secara ideal dapat memberikan tingkat keyakinan dalam pengukuran dan kemudian melakukan penjejakan (*tracking)*.

Penjejakan dengan konsep *deformable templates* ini menawarkan pendekatan yang menjajikan untuk memecahkan beberapa permasalahan dalam penjejakan, khususnya yang berhubungan dengan fitur yang hadir dalam benuk yang lebih kompleks dan rumit.

Pada konsep *deformable templates* akan dijelaskan formulasi alternatif dalam melakukan proses unit *matching* bentuk diskret. Hal tersebut mengilustrasikan kepada kita beberapa aplikasi untuk mendeteksi jejak partikel dalam percobaan fisik dengan energi besar. Proses reformulasi ini menyediakan hubungan yang tepat antara *deformable template* transformasi Hough (lihat bagian sebelumnya). Pada hakikatnya bagian ini tidak akan menerangkan lebih jauh mengenai *deformable template,* melainkan hanya akan memberikan semacam *preview*, sehingga diharapkan pembaca sekalian dapat mencari sendiri referensi mengenai *deformable template* bila ingin memperdalam proses dan bentuk kegiatan yang terdapat di dalamnya.

Penjejakan dengan *deformable template* dipengaruhi oleh *snakes* dan oleh model yang elastik. Kerja pada *deformable templates* dengan proses *matching units* berhubungan dengan model jaring yang elastik dan kemudian men-generalisasikannya ke dalam banyak permasalahan *matching*.

Template matching adalah salah satu dari sekian banyak pendekatan klasik dalam pendeteksian fitur. Dalam kebanyakan bentuknya yang klasik, proses ini menyertakan *convolving* sebuah citra dengan *mask corresponding* agar fitur yang diinginkan dapat sesegera mungkin dideteksi. Pendekatan ini dinilai sangat efektif dalam domain tertentu, tapi memiliki beberapa kekurangan, yaitu akan gagal jika objek dalam citra lalai dideformasi atau seandainya pencahayaan pada objek sangat berbeda dari yang digunakan untuk membangunkan *template.* Jadi salah satunya

memerlukan *template* tersebut untuk secara relatif mengurangi distorsi secara geometri dan variasi pencahayaan yang kadang-kadang menjadi faktor penggangu.

Pendekatan *deformable template* diharapkan dapat memecahkan beberapa permasalahan, yang terdiri atas tiga elemen dasar berikut ini:

- Sebuah model geometri yang terparameter untuk tambahan yang meliputi kemungkinan utama dari parameter pada *template*.
- Model pencitraan untuk menspesifikasi bagaimana sebuah *deformable template* memberikan pertambahan intensitas pada citra dengan teliti. Hal ini dapat diekspresikan sebagai sebuah pengukuran citra.
- Sebuah algoritma yang menggunakan pengukuran citra dan bentuk geometri dari *fitness* untuk proses *matching* pada *template* dalam sebuah citra.

Dinilai sangatlah penting untuk memberi bentukan pada definisi di atas pada kondisi probabilitas (konsep dasar mengenai probabilitas akan diberikan pada Bab selanjutnya).

4.2.4. PENJEJAKAN MODEL DAN OBJEK TIGA DIMENSI PEJAL MAUPUN NONPEJAL

Model tiga dimensi berbasis *vision* dimaksudkan untuk menemukan okurensi objek tiga dimensi yang diketahui menyertai sebuah citra dan memperoleh pengukuran dari lokasi objek pada citra. Keberadaan maupun lokasi objek kemudian dapat digunakan untuk beberapa perintah seperti manipulasi robot, proses monitoring, maupun *vehicular control.* Karena hanya beberapa aspek saja yang biasa dilengkapi (seperti geometri sudut pada citra), maka aspek tersebut dikatakan membentuk model objek, yakni okurensi model yang sebenarnya hanya dapat dilihat oleh *user*. Model geometri adalah salah satu bentuk model yang mudah dalam pengerjaan,

karena invarian yang kuat dari geometri di bawah proyeksi perspektif dapat menyediakan *reliability* dan kemudahan komputasi. Dengan demikian hasil model geometri akan lebih bersifat kuantitif. Kelebihan model geometri akan menjadi dasar bagi model pejal, parameter lain yang terukur.

Bentuk geometri yang fleksibel, lunak, dan umumnya dideskripsikan sebagai objek sangat sulit digunakan. Tapi, pada bahasan selanjutnya mengenai penjejakan dengan model objek nonpejal akan Anda temukan pula kelebihan model tersebut, yakni mudah diperlengkapi seperti dengan tekstur dan warna, dan menguatkan eksistensi objek tersebut, meskipun bukan sebuah pengukuran kuantitatif pada lokasi tiga dimensi. Penjejakan berdasarkan model citra adalah *model-based vision* yang diaplikasikan pada sekuens video citra, dan muncul dengan banyak permasalahan rumit yang lainnya karena *high-date rate* dalam sekuens citra adalah 10 Mbyte/s pada *video rate*. Meskipun begitu, kontinuitas di antara citra dapat disederhanakan masalahnya, karena gerak objek dapat diantisipasi dengan beberapa presisi yang kemudian menjadi lebih menguntungkan untuk proses pada *rate* yang maksimum, yakni sekitar 50 Hz untuk kamera video standard.

Fitur pada model geometri yang digunakan pada proses penjejakan haruslah sederhana untuk diekstrak (misalnya murah dalam proses komputasi) jika proses yang akan dikerjakan terletak di dekat *rate* video. Proses komputasi yang mahal dan rumit pada fitur model, seperti region yang tertutup yang merepresentasi permukaan, tidak akan dapat dihasilkan dari proses tadi. Hal ini mengindikasikan penggunaan fitur lokal yang sederhana seperti *points* (ataupun *corners*) serta sudut. Fitur titik kelihatannya lebih sulit untuk diekstrak dari citra bila dibandingkan dengan sudut.

Di bawah ini akan diberikan algoritma untuk peningkatan kualitas penjejakan objek dengan model geometri, yakni ketika objek yang dijejak

bergerak. *Control points* kadang-kadang mempunyai arah yang tidak jelas, misalnya perputaran berkeliling ke arah belakang objek. Masing-masing *control point* harus dimatikan agar seperti tidak kelihatan atau masing-masing *point* akan mengambil sudut yang salah, dan menggabungkannya dengan hasil penjejakan sebelumnya. Untuk mengatasi masalah yang sering datang ini, dibuatlah tabel *view-potential* yang terkuantisasi, yang masing-masing masukannya akan memberikan perbedaan pada *viewpoint* kamera dengan kamera sebenarnya yang digunakan untuk melakukan penjejakan pada objek. Masing-masing tabel masukan itu disimpan untuk kepentingan *control point* yang terkadang tidak jelas atau malah tidak terlihat itu. Tabel *view potential* dibuat berdasarkan bentuk persegi pada permukaan yang terletak jauh pada sekitar kotak dengan bentuk kubus yang dikelilingi objek.

Penggunaan sudut dengan tanggapan yang lemah untuk penjejakan sangat tidak dianjurkan karena dinilai sangat berbahaya karena secara periodik dapat menjadikan halangan yang tidak diharapkan. Untuk alasan ini, sebuah *control point* secara otomatis dimatikan, ketika tanggapan sudut terlihat sangat kecil dan tidak dapat digunakan ketika berubah menjadi bentuk yang lebih besar. Atribut yang sangat berguna pada kondisi ini adalah polaritasnya, yakni ketika diletakkan pada gelap/terang maupun terang/gelap. Polaritas dari sebuah sudut biasanya tidak banyak dipakai, agar dapat berperan untuk *device* yang bermanfaat untuk mengeluarkan sudut yang tidak terpakai, yakni sudut terdekat pada citra yang seringkali memliliki polaritas berlainan.

Untuk menghadapi kecepatan data masukan sampai dengan 10 Mbyte tiap detik, harus tercapai pengalamatan dan proses interpretasi gerak penuh video dalam beberapa perintah komputasi untuk sembarang sistem pemrosesan. Jika hal ini tidak tercapai, maka fokus pemrosesan akan terletak pada jangkauan yang jauh dari tujuan umum pada *personal*

computer, bahkan diasumsikan untuk model matematika objek di bawah observasi.

Penggunaan video konvensional untuk proses presisi posisi tiga dimensi memiliki beberapa aplikasi praktis dan menawarkan keuntungan signifikan dari segi biaya dan kompleksitas skema alternatif di dalam masing-masing analisis dan perintah pengendali *real-time.*

Untuk penjejakan model tiga dimensi nonpejal, model *deformable* melakukan pendekatan dengan estimasi gerak objek tiga dimensi yang pertama dikenalkan oleh Terzopoulos, pada tahun 1988. Model silinder yang *deformable* dibuat untuk mengambil bentuk dari *elastic tube* pada *deformable spine*.

Bentuk elastiknya men-generalisasi komponen *spline* dan membuatnya menjadi sepenuhnya *deformable*. Bila dianalogikan dengan *snakes*, sifat model diatur oleh persamaan elastodinamis dan model yang ada dipasangkan dengan data yang visual melalui medan gaya yang terkomputasi dari satu atau lebih masukan citra; bagaimana pun, komputasi medan gaya tidaklah sesederhana seperti yang terdapat pada *snakes.* Model akan merasakan gaya dengan *sampling* gradien dari fungsi potensial tiga dimensi yang didefinisikan melalui keseluruhan ruang yang meliputi model.

Potensial yang didapat tidak bernilai nol hanya dengan volume yang telah terdefinisi oleh penggabungan proyeksi yang terkerucut dari *viewpoint* yang telah diasumsikan melalui korespondensi citra dan keluar dari *domain space*. Potensial tersebut diturunkan dari pemrosesan masing-masing citra dan bentuk ekspansi sempurna dari proyeksi yang terkerucut tadi. Untuk membuat sebuah potensial yang mendukung konsistensi antara model tiga dimensi dan profil citra dari objek tiga dimensi, masing-masing citra akan diolah dengan pengaplikasian *blurring* dan gradien kernel. Model

silinder yang *deformable* melakukan pencuplikan atas gradien potensial yang terdekat dengan permukaan di luar kontur, misalnya ketika permukaan vektor normal diaproksimasi *perpendicular* terhadap garis pandangan.

Pada hakikatnya terdapat sejenis pertukaran antara deformasi lokal dan global. Deformasi global membutuhkan beberapa *data point* relatif untuk mengabstraksi banyak bentuk objek. Sebaliknya, deformasi lokal dapat menyediakan aproksimasi yang lebih akurat untuk menghitung bentuk yang paling tepat dari objek yang rumit dan kompleks, tapi tingkat *recovery*-nya membutuhkan lebih banyak data. Simbiosis antara deformasi lokal dan global dengan *deformable* yang *superquadrics* menawarkan bentuk terbaik dari dunia (*worlds*) .

4.2.5. METODE ASOSIASI DATA DENGAN SISTEM PENJEJAKAN

Pada dasarnya penjejakan merupakan estimasi parameter yang *time-variant* dari observasi yang tidak pasti pada beberapa parameter yang diketahui. Permasalahan pada penjejakan dapat dibagi atas dua bagian penting, yakni:

1. estimasi optimal dari keadaan yang ditargetkan
2. asosiasi data yang menghubungkan proses *sensing* untuk mengestimasi mekanisme.

Selanjutnya akan kita bahas masing-masing permasalahan dengan memberikan gambaran yang terdapat pada kedua butir di atas.

1. Estimasi optimal dari keadaan yang ditargetkan

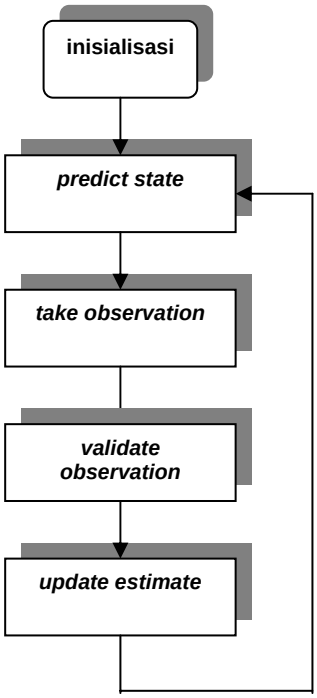
Permasalahan pertama ini terdiri atas proses mendeduksi kemungkinan terbaik untuk mengestimasi keadaan dari pengukuran yang bersifat sekuensial. Jika diberikan sekuen sementara dari pengukuran berderau untuk satu keadaan, semua faktor terukur telah diketahui untuk kemudian dimunculkan pada target, sebuah estimasi

dapat diturunkan dalam beberapa cara. Regresi dan rekursi linier, pendekatan filter nonlinier, telah dikembangkan untuk mendekati permasalahan dengan cara yang lebih sistematis dan matematis. Meskipun demikian, yang dirasakan lebih banyak memberikan kontribusi adalah pengestimasi rekursif linier, yang banyak didasari estimasi filter Kalman. Aplikasinya dalam penjejakan visual telah diberikan pada subbab sebelumnya.

Di antara banyak alasan mengenai penjejakan sistem dengan cara ini, adalah:

- Struktur rekursif tertutup dari algoritma yang membolehkan estimasi dari keadaan (*state*) yang dapat ditetapkan pertambahannya dengan tiap pengukuran set yang baru.
- *Stage* prediksi yang membentuk hubungan antara estimasi yang paling akhir dan estimasi yang berikutnya untuk menciptakan penjejakan *notional.*
- Variansi estimasi dari evaluasi yang muncul pada tiap-tiap bagian lup.
- Pengestimasi akan memberikan hasil yang optimal dalam *sense* Bayesian (variansi minimum) jika semua keadaan dan observasi derau adalah Gaussian; jika tidak, gunakan pengestimasi linier

Karena sembarang estimasi dapat menghasilkan hasil dan keadaan yang berbeda dan akurasi sangatlah fundamental, yang mungkin muncul dari target yang diinginkan, maka proses pemindahan yang salah akan membuat observasi yang diketahui sebagai validasi dan proses *deciding* yang mana pengukuran yang tervalidasi dari pengestimasi seharusnya digunakan sebagai asosiasi data. Struktur algoritma filter Kalman dapat dilihat pada Gambar 4.19



Gambar 4.19 Struktur algoritma filter Kalman

2. Permasalahan asosiasi data

Asosiasi data berkaitan erat dengan pemilihan elemen dari satu set pengukuran yang digunakan untuk meng-*update* sebuah estimasi keadaan. Dalam kasus target tunggal, yaitu sebuah objek *solitary* dijejak dengan filter tunggal, permasalahan asosiasi data mengurangi pengukuran yang tervalidasi yang seharusnya digunakan untuk meng-*update* filter. Karena hanya satu target yang akan dijejak, kita dapat melakukan pemanggilan validasi kawasan untuk melakukan *clutter* terhadap target tersebut.

Meskipun begitu, sistem penjejakan berkaitan langsung dengan beberapa *data point* yang diobservasi, yang seharusnya:

- melakukan asosiasi pada tiap kemungkinan penjejakan terbaik dari *data point*

- menghindari pengambilan *data point* yang salah untuk penjejakan yang salah, terutama ketika objek melewati masing-masing objek terdekat.
- menginisiasi penjejakan baru untuk *data point* untuk memprediksi objek baru yang muncul pada lingkungan (*environment*).
- membuang semua *point* yang terambil sebagai *clutter*.

Ada dua pendekatan untuk asosiasi data, yakni :

- mengasumsikan sebuah target membangun hanya sebuah *data point* untuk tiap langkah observasi (kebanyakan digunakan).
- mengasumsikan sebuah target membangun beberapa *data point* untuk tiap langkah observasi (metode yang berbasiskan model).

Penjejakan ini bertujuan untuk berkosentrasi pada pendekatan yang pertama. Bersama dengan pendekatan yang pertama, terdapat dua kelas algoritma asosiasi data, yakni : optimal (dengan *Bayesian sense*) dan suboptimal. Pendekatan optimal dikenal sebagai *Multiple Hypothesis Data Association*. Pada pendekatan suboptimal, terdapat dua cara yang populer, yaitu *Nearest Neighbour Data Association* dan *All Neighbour Data Association*. Kebanyakan sistem penjejakan memakai algoritma asosiasi data yang diturunkan dari ketiga metode asosiasi data tersebut.

Asosiasi data optimal dapat dicapai melalui implementasi kebijakan hipotesis perkalian, tapi kinerja yang *real time* merupakan persyaratan utama, sehingga pendekatan ini tidak begitu mudah dikendalikan. Dalam kasus komputasi yang lebih sederhana, solusi yang suboptimal lebih dianjurkan dan kita kemudian akan memilih dari semua metode yang ada.

Pilihan itu nanti akan bergantung pada faktor krusial tipe data dalam sistem *sensing*.

4.2.6. GEOMETRI GERAKAN VISUAL

Sistem *vision* tiga dimensi menggunakan gerakan *visual* dari fitur citra untuk membangun dari bawah ke atas (*bottom-up)* deskripsi geometri tiga dimensi *scene* yang terlihat. Geometri tiga dimensi sebuah *scene* umumnya dikenal dengan representasi intermediate untuk mencapai *vision* agar dapat berperan sebagai *starting point* untuk kinerja variasi perintah yang *high level* (level tinggi) tanpa pencitraan dengan *computational cost.*

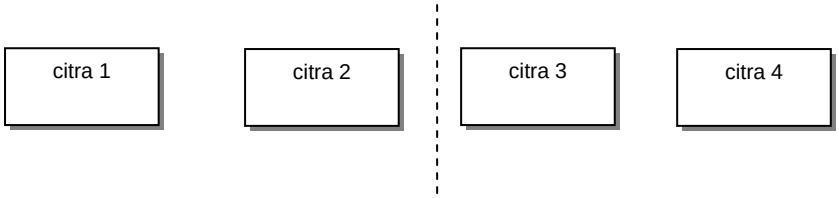
Sebuah representasi tiga dimensi seharusnya dapat menahan informasi struktural yang bernilai pada saat pembuangan sejumlah besar piksel yang *redudant* pada sebuah citra dan kemudian dapat digunakan sebagai konsentrator informasi. Hal ini sangat penting untuk sistem *vision* yang *real time* karena dinilai terlalu banyak piksel pada sebuah sekuen *video-rate* citra untuk diikutsertakan pada kinerja yang *high level* maupun pemrosesan (pengolahan) yang kompleks dan rumit.

Kelengkapan struktur geometri tiga dimensi biasanya diperlukan untuk sekuen citra yang tepat; terdapat banyak kelebihan pada citra karena kekontinuan gerak dan invariansi geometri tiga dimensi dengan perubahan *viewpoint*. Pembentukan sebuah sekuen citra pada hakikatnya terlebih dahulu diasumsikan, untuk memperlihatkan kontinuitas yang cukup, dan juga karena tingkat *smoothness* dan *slowness* gerak, baik dari kamera maupun dari objek yang berkaitan. Invariansi geometri tiga dimensi dengan perubahan *viewpoint* tadi diturunkan dari tingkat kepejalan (*rigidity*), yang secara keseluruhan diciptakan dari objek individual yang terintegrasi di dalamnya. Struktur tingkat kepejalan ini memperbolehkan geometri *scene* berubah menjadi relatif terekstraksi dari gerak visual dalam sebuah sekuen citra. Hal tersebut kemudian dibandingkan satu sama lain untuk membuat

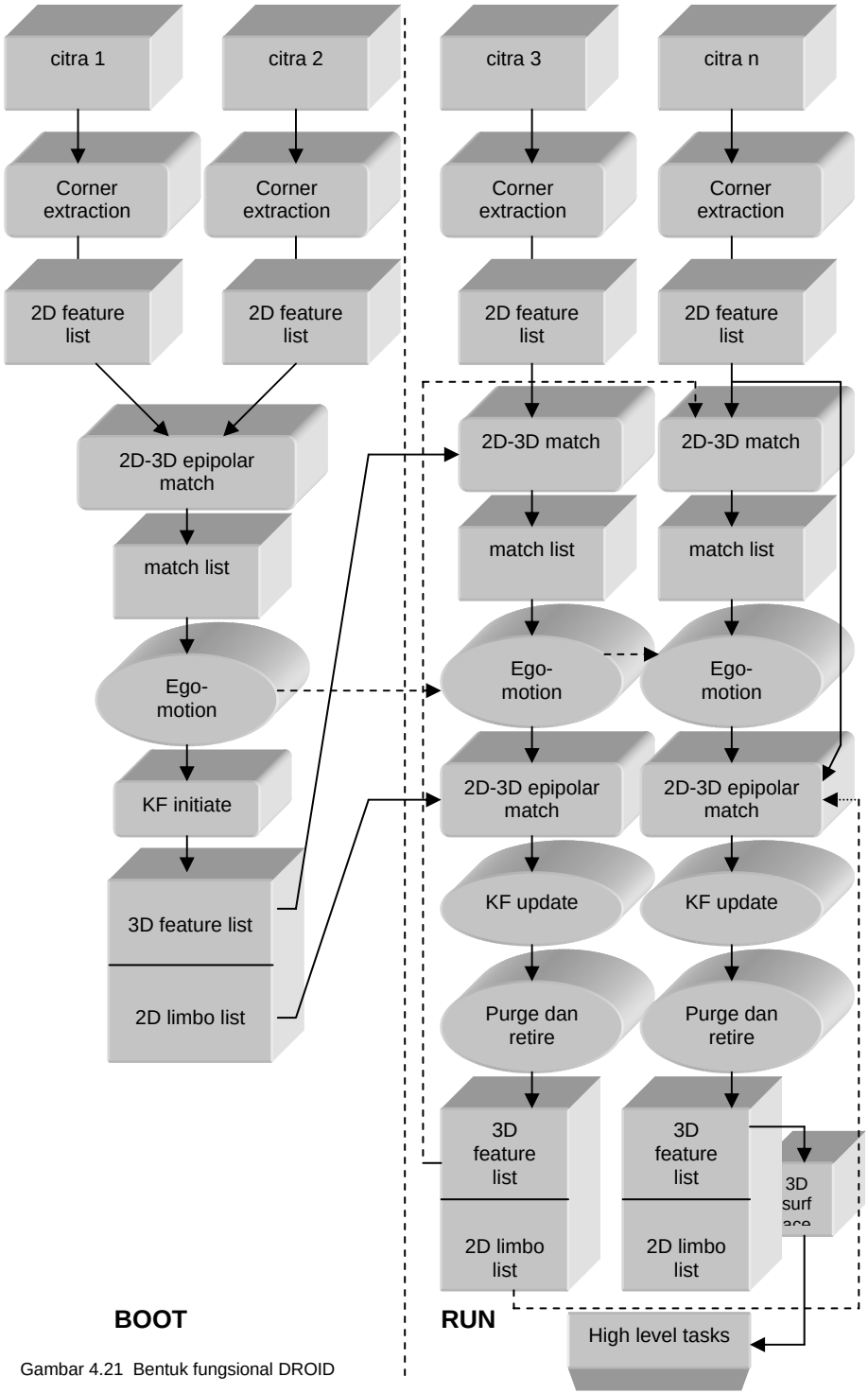
bentuk dari teknik X (dalam hal ini, X mewakili *shading, shadows, reflectance, texture,* dan juga *perspective*) yang kemudian diaplikasikan untuk citra tunggal, yang pada tahapan akhir hanya akan hadir sebuah saja scene yang tidak bebas.

Sebuah sistem *vision* yang hanya menggunakan geometri harus bergantung pada tingkat keakuratan geomtri tersebut. Tingkat keakuratan geometri akan dapat dicapai melalui penggunaan garis dasar yang panjang untuk *triangulation*. Beberapa mekanisme harus ada untuk melewati informasi citra yang menyeberangi sejumlah citra yang besar (dengan potensial yang tidak terbatas). Fitur yang berbasiskan metode ini memungkinkan hal ini dibuat menjadi lebih sederhana.

Sebuah sistem yang berkinerja visual geometri, yakni DROID, mampu menampilkan interpretasi tiga dimensi dari sekuen citra digital dengan memakai teknik ekstraksi yang otomatis, penjejakan, dan lokalisasi tiga dimensi dari fitur citra yang dimaksudkan. *Scene* yang terlihat diasumsikan menjadi *scene* yang pejal *piece-wise*. Sistem DROID didesain untuk dioperasikan dalam struktur lingkungan yang tidak bebas, tanpa pengetahuan mengenai isi lingkungan itu terlebih dahulu. Dari lokasi penjejakan fitur citra tersebut, DROID akan menentukan gerak kamera dan lokasi fitur tiga dimensi itu. Permukaan tiga dimensi ini diharapkan dapat membentuk *point awal* untuk proses pengambilalihan perintah *high level* yang sangat bervariasi, seperti pengenalan dan navigasi. Bentuk fungsional DROID ini dapat digambarkan seperti pada Gambar 4.20 dan Gambar 4.21.



Gambar 4.20 Sekuen citra



Gambar 4.21 Bentuk fungsional DROID

4.2.7. PERENCANAAN GERAKAN MENGGUNAKAN DIVERGENSI CITRA DAN DEFORMASI

Gerak relatif yang terletak di antara pelaku observasi dan *scene* akan cenderung menginduksi deformasi dalam bentuk dan detail citra. Jika semua perubahan tersebut masih terlihat tidak begitu jelas, mungkin dapat dideskripsikan lokal dengan invarian persamaan diferensial orde satu dari kecepatan medan citra, *curl, divergence, shear,* dan komponen. Semua invarian itu memiliki setidaknya bentuk geometri yang sederhana yang tidak bergantung pada pilihan khusus sistem koordinat yang ditentukan. Lebih–lebih lagi jika terhubung dengan struktur tiga dimensi dari *scene* dan pergerakan yang secara khusus merupakan orientasi permukaan.

Di samping itu, divergensi dan komponen deformasi kecepatan medan citra tidak mempunyai efek yang berarti oleh perputaran penglihat yang tidak menentu arahnya berdasarkan pusat data yang telah ditentukan. Selanjutnya, ditentukan pula sebuah faktor keefisienan, yaitu cara yang lebih mudah dipakai untuk memperbaiki orientasi permukaan dan waktu untuk kontak.

Orientasi permukaan dan waktu untuk kontak dapat dilakukan dengan beberapa cara sebagaimana dijelaskan berikut ini.

1. **Dengan pengetahuan translasi yang ada**

   Sebuah estimasi mengenai arah translasi biasanya disediakan ketika para penglihat dibuat dengan *vision* yang *binocular* (ketika kamera ataupun posisi mata tidak leluasa). Proses ini juga dapat dilakukan dengan mengestimasi melalui pengukuran dengan gerak yang paralaks jika translasi *viewer* diketahui, dan dianggap cukup tidak membingungkan untuk kemudian kembali pada orientasi permukaan dan jarak objek dan unit sifatnya tetap. Karena skala

kecepatan selanjutnya mungkin membingungkan dan diekspresikan sebagai waktu untuk kontak, sebuah solusi dapat dicapai dengan beberapa langkah berikut:

⇒ *axis* ekspansi (μ) dari komponen deformasi dan proyeksinya di dalam citra pada arah translasi (**< A**) membolehkan kembalinya kemiringan permukaan dari persamaan gerak yang telah ditentukan.

⇒ waktu untuk kontak memperbaiki translasi *viewer* dalam unit sementara. Hal ini memperbolehkan spesifikasi magnitude dari paralel translasi pada permukaan datar citra. Magnitude deformasi kemudian dapat digunakan untuk mengembalikan kemiringan permukaan.

2. Dengan fiksasi

Jika kamera ataupun mata melakukan rotasi untuk tetap mempertahankan keberadaan objek pada pertengahan citra tersebut, *magnitude* rotasi perlu membawa objek kembali ke pusat dari citra yang masih dapat ditentukan. Juga melawan efek utama dari sembarang kesalahan dalam mengestimasi rotasi untuk menentukan skala kedalaman dan orientasinya.

Seperti yang telah kita ketahui sebelumnya, divergensi citra dapat digunakan untuk menghindarkan bentrokan dengan penghalang. Nelson dan Aloimonos (pada tahun 1988) telah mendemonstrasikan sistem robotika yang menggunakan komputasi divergensi dengan teknik *spatio-temporal* yang dipakai pada citra dengan tingkat tekstur tinggi pada permukaan *visible*. Divergensi citra dapat dikatakan merupakan implementasi *real time* yang berbasiskan kontur citra dan berperan pada informasi turunan yang visual.

Manipulator robot, selanjutnya, membuat gerakan yang tidak disengaja terhadap target. Proses penjejakan terhadap kawasan kontur dan

komputasi perubahan rataannya memperbolehkan untuk diestimasi nilai divergensinya. Gerakan sepanjang sinar visual dinilai memiliki cukup informasi untuk mengestimasi waktu untuk kontak ataupun tubrukan. Estimasi waktu untuk tubrukan ini berkurang oleh ketidakpastian dalam pengukuran dan sembarang deformasi citra yang dapat digunakan untuk menuntun manipulator sehingga dapat berhenti sebelum tubrukan terjadi. Pada hakikatnya, manipulator berjalan 'secara buta' setelah aksi/perbuatannya yang dibuatnya dan dengan kecepatan yang *uniform* sebagai waktu *remaining* hingga terjadinya kontak.

Jika gerak translasi memiliki komponen yang paralel, maka divergensi citra akan disusun oleh dua komponen utama. Yang pertama adalah komponen yang menentukan waktu untuk kontak/tubrukan. Kondisi/komponen yang lain menunjuk pada *foreshortening* citra ketika permukaan memiliki kemiringan nol. Kesemua dampak itu dapat dikomputasi secara terpisah dengan pengukuran yang dilakukan pada saat deformasi. Deformasi juga memperbolehkan kita untuk memperbaiki orientasi gerak pada permukaan.

4.2.8. NAVIGASI LOKAL ADAPTIF

Hubungan yang efektif antara persepsi dan aksi dapat dicapai dengan pemetaan aliran sementara pada data sensor hingga *stream* aksi tersebut untuk mencapai tujuan yang diinginkan. Untuk sebuah mobil robot, pembatasan aktivitas yang *real time* dan sumber komputasi yang terbatas melarang perluasan aksi optimal yang dapat dipilih melalui metode *search* menggunakan model dunia yang akurat.

Untuk perintah yang membutuhkan tingkat kesuksesan yang bergantung pada sekuen dan sifat yang tepat seperti sensor derau, ketidakpastian mengenai lingkungan penjejakan, dan ledakan kombinatorial pada kemungkinan masa depan, dikombinasikan untuk memastikan bahwa

beberapa asumsi haruslah dibuat dan di-'jalan pintaskan' dalam proses pembuatan keputusan. Telah diketahui bahwa sifat cepat (*fast*) dan kokoh (*robust*) dapat dicapai melalui perintah yang didedikasikan pada modul yang menggunakan sensor data langsung, keadaan internal yang kecil, dan mekanisme pembuatan keputusan yang relatif sederhana, untuk memilih di antara aksi-aksi alternatif yang ada.

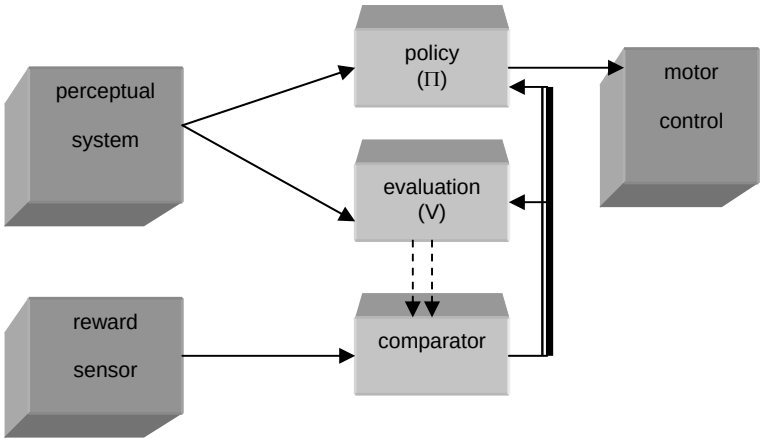
Meskipun demikian, untuk merencanakan dan memodelkan berarti aksi yang telah dipilih hanya dalam konteks *narrow time window*. Sifat yang telah ada tidak dapat diorganisasi dengan efisien dengan hanya mengandalkan tujuan berjangka panjang yang hanya akan dicapai dalam maksud yang oportunistik.

Satu jawaban untuk dilema ini terletak pada pengembangan modul adaptif yang dapat dimonitor kinerjanya berdasarkan tujuan yang ingin dicapai oleh sistem dan meningkatkan pilihan pada sifat pengendali yang sesuai. Dengan menggunakan pengalaman yang ada pada robot dengan tepat, sistem dapat mengelakkan halangan dengan cara melakukan proses pemilihan sifat yang telah sukses dilaksanakan pada situasi yang sama pada masa lampau. Bagaimana pun, jika sebuah sistem dapat 'belajar' secara langsung dari pencapaian yang ada, maka seharusnya dapat juga melakukannya dengan pen-*delay*-an dan pelatihan terhadap data yang tidak begitu informatif terhadap kebutuhan penjejakan pada sistem robot ini.

Cara yang dirasakan paling optimal untuk melakukan fungsi dan subrutin di atas adalah dengan menentukan pengendali yang optimal pula. Selain itu juga dengan penerapan kondisi pemrograman dinamik yang heuristik – HDP (*Heuristic Dynamic Programming*) yang digunakan oleh Werbos (pada tahun 1990) untuk mendeskripsikan algoritma '*belajar*' pada beberapa tipe robot.

Pemrograman dinamik yang klasik adalah dengan jalan mencari metode untuk menentukan kendali optimal yang melampaui ruang keadaan (*state space*) yang di dalamnya terdapat sejumlah terbatas kemungkinan aksi yang dilakukan.

Berikut ini diberikan gambaran singkat mengenai beberapa prinsip yang terkait pada sistem sambil mendeskripsikan algoritma HDP yang spesifik yang akan dipakai.



keterangan :

1. = *prediction*
2. = *prediction error*

Gambar 4.22 Algoritma HDP (*Heuristic Dynamic Proramming*)

Sifat navigasi lokal yang berhasil dapat meningkat dengan sekuen-sekuen yang dipelajari dari reaksi pada pengurutan data perseptual. Lintasan yang dibangun terkadang memiliki penampakan aktivitas yang terencana karena aksi indivisual hanyalah tepat dipakai sebagai bagian dari pergerakan *pattern*. Bagaimana pun, perencanaan yang muncul sebagai bagian implisit dari proses ‘belajar’ HDP akan mengizinkan terjadinya proses ‘keluar-masuknya’ pengaruh pada masa depan yang diambil berdasarkan konteks yang sama pula. Proses

belajar ini efektif karena mampu melingkupi interaksi robot dengan dunianya untuk melakukan pemetaan yang efektif dari sensor data hingga ke aksi pergerakan motornya.

4.2.9. SISTEM VISION PARALEL TIGA DIMENSI

Pada bagian ini akan lebih banyak ditemui skema ataupun diagram bagian-bagian penting dari penginderaan pada mobil robot, baik pada proses *matching* sampai ke arsitektur implementasi paralel yang digunakan.

Untuk bekerja dengan sistem *vision stereo* dengan banyak transputer, akan banyak dijumpai bentuk dari sejumlah besar keparalelan *device*. Sistem yang diimplementasikan dispesialisasi secara arsitektur. Perangkat keras akan menyediakan *datapath* video yang terdistribusi pada *array processor*. Untuk mengenali dan memperbaiki posisi model objek, kita cenderung memakai data yang terparalel pada masing-masing level objek dan subobjek.

Semua kepentingan itu selanjutnya akan diintegrasikan ke dalam tujuan arsitektur yang umum dan akan disajikan dalam beberapa hasil kinerja dan deskripsi dari penggunaan mesin *vision* dalam dua domain aplikasi.

Untuk mencapai *vision* dinamik yang *real time* dibutuhkan beberapa investigasi teknik pemrosesan paralel. Pemrosesan visual pada umumnya dapat dikelompokkan pada tiga level, yakni:

a. Iconic to iconic

Proses ini membentuk bagian *low level* dari interpretasi *scene* dan umumnya terdiri atas operasi relatif yang sederhana sepanjang data citra hingga lokasi piksel dengan karakteristik seperti kekuatan sudut ataupun informasi topologi.

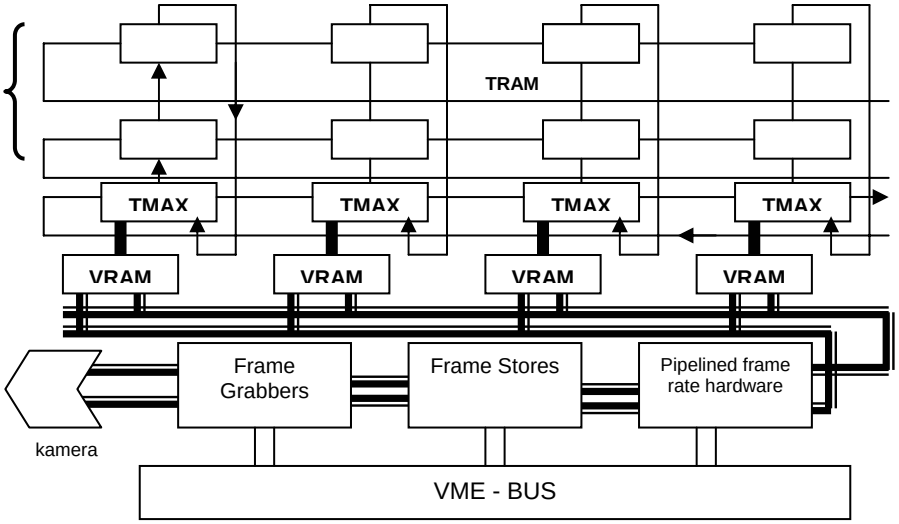
b. Iconic to symbolic

Ini merupakan pemrosesan pada level medium yang secara tipikal menerima data dari bagian *low level* untuk diekstrak secara deskripsi simbolik seperti segmentasi dan ekstraksi fitur.

c. Symbolic to interpretation

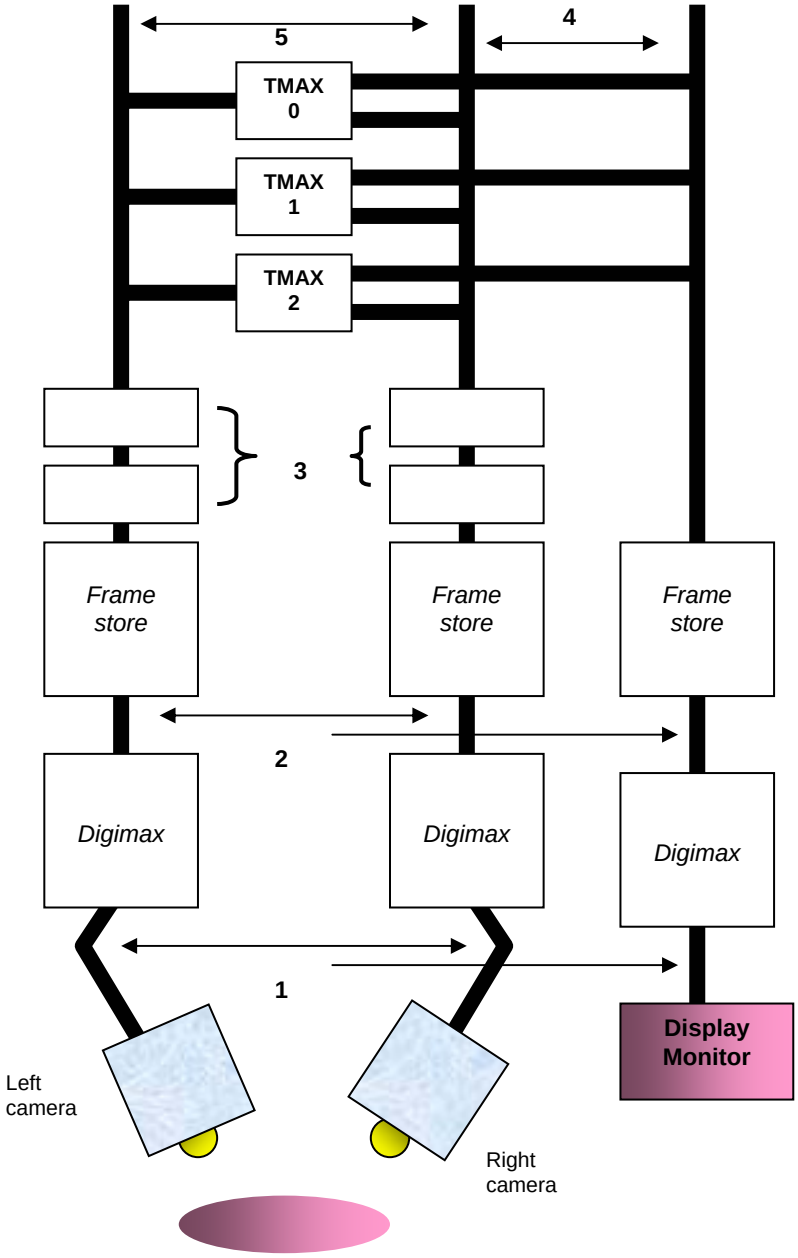
Proses yang ada mengekstrak arti data yang diproduksi oleh bagian sebelumnya seperti pengenalan objek. Kerumitan algoritmanya pada *vision* yang *high level* umumnya membutuhkan implementasi berbasiskan tujuan umum komputer tersebut.

Desain arsitektural dari **MARVIN** (**M**ultiprocessor **AR**chitecture for **VI**sio**N**) dimaksudkan untuk memenuhi beberapa persyaratan dari mesin *vision* yang berkinerja tinggi untuk aplikasi robotika yang : membutuhkan tenaga untuk komputasi yang berskala besar, memiliki lebar pita yang besar untuk *datapath* pada aliran data video, mendukung variasi level pada pengolahan *vision*, dan tanggapan yang *real time*.



Gambar 4.23 Desain arsitektural MARVIN

Gambar 4.23 adalah arsitektur MARVIN, sedangkan berikut ini disajikan diagram *video dataflow* MARVIN.



Gambar 4.24 Diagram *video dataflow* MARVIN

Keterangan gambar :

1. ***analogue video***
2. ***digital interlaced citras***
3. ***pipelined frame rate-hardware***
4. ***outgoing non interlaced display stream***
5. ***incoming non interlaced distributed video streams***

Sedikit penjelasan mengenai arsitektur **MARVIN**.

- **T MAX** membolehkan bekerjanya beberapa prosesor **T800** secara bersamaan, sampai dengan 4 buah *stream* citra. **TMAX** juga menawarkan 1 Mbyte Video RAM (**VRAM**), dengan batasan penyimpanan sampai 512 citras x 4 dan juga sampai 4 Mbyte **DRAM**.
- Bus video dapat digunakan bersama-sama dengan distribusi data sampai 32 bit.
- ***Card*** **TMAX** yang digunakan berjumlah delapan buah, dengan modul transputer sampai dengan 2 Mbyte.

LATIHAN DAN SOAL
BAB IV

1. Jelaskan berbagai metode pada proses segmentasi.
2. Jelaskan pendeteksian tepi berdasarkan gradien.
3. Sebutkan perbedaan yang mungkin antara pendeteksian *Gradient based* dan *Laplacian based*.
4. Sebutkan karakteristik transformasi Hough.
5. Jelaskan pula pendeteksian garis dengan regresi linier.
6. Jelaskan penjejakan objek dengan bantuan *ray tracing* dan sebuah model kamera dengan diagram dan keterangan yang bersesuaian.
7. Sebutkan model pencahayaan dan model pada pemantulan. Juga jelaskan bagaimana menentukan sinar yang memotong objek.
8. Berikan gambaran singkat mengenai proses penjejakan objek.
9. Sebutkan dan jelaskan hal-hal yang terdapat pada sistem vision aktif.
10. Jelaskan *Kalman Snakes Object Tracking* dengan konsep matematis yang anda ketahui mengenai.
11. Gambarkan konfigurasi jaringan transputer. Berikan keterangan yang memadai.
12. Sebutkan tiga elemen dasar pada pendekatan *deformable template*.
13. Sebutkan dan jelaskan dua bagian penting dari penjejakan.
14. Gambarkan bentuk fungsional dari struktur sistem DROID.
15. Jelaskan cara orientasi permukaan dan waktu untuk kontak pada penjejakan objek.
16. Apa yang Anda ketahui mengenai navigasi lokal adaptif ?
17. Jelaskan sistem vision paralel tiga dimensi.
18. Gambarkan arsitektur MARVIN dan beri penjelasan.
19. Gambarkan pula diagram *video dataflow* dari MARVIN.

20. Dengan pengetahuan yang telah diberikan sebelumnya, rancanglah (tidak perlu sampai pada tahap implementasi) sebuah sistem *vision* untuk melakukan penjejakan pada benda yang tidak mempunyai bentukan khusus, seperti air, dan lain-lain (Anda dapat memilih bentuk objeknya). Karakteristik unggul apakah yang terdapat pada sistem yang anda rancang ?
21. Berikan algoritma yang bersesuaian dengan diagram *video dataflow* pada MARVIN.
22. Jelaskan arti fisis sistem *vision* tiga dimensi tersebut.
23. Rancanglah sebuah sistem *vision* pada mobil robot yang paling sederhana (termasuk pengenalan pola, *buffering frame*-nya dengan perangkat RIO, serta penjejakan objek dengan metode yang paling Anda sukai).

    Catatan :

    Anda hanya diinstruksikan merancang algoritmanya saja. Sistem Anda dapat membedakan objek yang persegi, bentuk lengkung, serta bentuk dinding.