

**PENGHITUNGAN PANJANG DAN BERAT IKAN  
MENGUNAKAN HARRIS-LAPLACE DETECTORS**

**Skripsi**

**Disusun untuk memenuhi salah satu syarat  
memperoleh gelar Sarjana Komputer**



**Oleh:  
Prabowo Darmawi  
1313619001**

**PROGRAM STUDI ILMU KOMPUTER  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS NEGERI JAKARTA**

**2024**

## KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Allah SWT, karena dengan rahmat dan karunia-Nya, penulis dapat menyelesaikan proposal skripsi yang berjudul *Penghitungan Panjang dan Berat Ikan dengan Menggunakan Harris-Corner Detection*.

Keberhasilan dalam penyusunan proposal skripsi ini tidak lepas dari bantuan berbagai pihak yang mana dengan tulus dan ikhlas memberikan masukan guna sempurnanya proposal skripsi ini. Oleh karena itu dalam kesempatan ini, dengan kerendahan hati penulis mengucapkan banyak terima kasih kepada:

1. Yth. Para petinggi di lingkungan FMIPA Universitas Negeri Jakarta.
2. Yth. Ibu Dr. Ria Arafiah, M.Si selaku Koordinator Program Studi Ilmu Komputer.
3. Yth. Bapak Muhammad Eka Suryana, M.Kom selaku Dosen Pembimbing I yang telah membimbing, mengarahkan, serta memberikan saran dan koreksi terhadap proposal skripsi ini.
4. Yth. Bapak Med Irzal, M.Kom selaku Dosen Pembimbing II yang telah membimbing, mengarahkan, serta memberikan saran dan koreksi terhadap proposal skripsi ini.
5. Kedua orang tua dan kakak penulis yang telah mendukung dan memberikan semangat serta doa untuk penulis.
6. Teman-teman Program Studi Ilmu Komputer 2019 yang telah memberikan dukungan dan memiliki andil dalam penulisan proposal skripsi ini.

Penulis menyadari bahwa penyusunan proposal skripsi ini masih jauh dari sempurna karena keterbatasan ilmu dan pengalaman yang dimiliki. Oleh karenanya, kritik dan saran yang bersifat membangun akan penulis terima dengan senang hati. Akhir kata, penulis berharap tugas akhir ini bisa bermanfaat bagi semua pihak khususnya penulis sendiri. Semoga Allah SWT senantiasa membalas kebaikan semua pihak yang telah membantu penulis dalam menyelesaikan proposal skripsi ini.

Jakarta,

Prabowo Darmawi

# DAFTAR ISI

<b>DAFTAR ISI</b>	<b>iv</b>
<b>DAFTAR GAMBAR</b>	<b>v</b>
<b>DAFTAR TABEL</b>	<b>vi</b>
<b>I PENDAHULUAN</b>	<b>1</b>
1.1 Latar Belakang Masalah . . . . .	1
1.2 Rumusan Masalah . . . . .	3
1.3 Batasan Masalah . . . . .	3
1.4 Tujuan Penelitian . . . . .	3
1.5 Manfaat Penelitian . . . . .	3
<b>II KAJIAN PUSTAKA</b>	<b>4</b>
2.1 <b>Multi-scale representation</b> . . . . .	4
2.1.1 <b>Gaussian scale-space</b> . . . . .	4
2.1.2 <b>Scale-space derivatives</b> . . . . .	6
2.1.3 <b>Second moment matrix</b> . . . . .	8
2.2 <b>Automatic scale selection</b> . . . . .	11
2.2.1 <b>Scale-space maxima</b> . . . . .	12
2.2.2 <b>Gamma normaozation</b> . . . . .	14
2.2.3 <b>Differential expressions for scale selection</b> . . . . .	15
2.3 <b>Scale invariant detector</b> . . . . .	16
2.3.1 <b>Harris-Laplace detector</b> . . . . .	16
2.3.2 <b>Scale covariant points</b> . . . . .	18
2.4 <b>Affine invariant detector</b> . . . . .	19
2.4.1 <b>Harris-Affine detector</b> . . . . .	19
2.4.2 <b>Affine covariant points</b> . . . . .	24
<b>III METODE PENELITIAN</b>	<b>26</b>
3.1 <b>Deskripsi Sistem</b> . . . . .	26
3.2 <b>Perancangan Sistem</b> . . . . .	26
3.2.1 <b>Pemrosesan Citra</b> . . . . .	26
3.2.2 <b>Smoothing Gaussian</b> . . . . .	27
3.2.3 <b>Perhitungan Gradien</b> . . . . .	28
3.2.4 <b>Autokorelasi</b> . . . . .	29
3.2.5 <b>Harris Respon</b> . . . . .	29
3.2.6 <b>Thresholding dan <i>Non-Maximum suppression</i></b> . . . . .	30
3.2.7 <b>Memilih Piksel</b> . . . . .	30
3.2.8 <b>Menghitung Panjang dan Berat</b> . . . . .	31

## DAFTAR GAMBAR

Gambar 2.1	. . . . .	5
Gambar 2.2	. . . . .	8
Gambar 2.3	<i>Scale trace of squared gradient and Laplacian</i> diterapkan pada model sudut dan tepi. Kolom pertama: tunjuk di dalam sudut. Kolom kedua: titik di luar sudut. Kolom ketiga: arahkan di dekat tepi. Di bingkai: Baris atas: Model sudut dan tepi. Baris tengah: Jejak skala untuk $\gamma = 1$ . Baris bawah: Jejak skala untuk $\gamma = 0,5$ . . . . .	13
Gambar 2.4	. . . . .	18
Gambar 2.5	. . . . .	19
Gambar 2.6	. . . . .	20
Gambar 2.7	. . . . .	24
Gambar 2.8	. . . . .	25

## **DAFTAR TABEL**

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang Masalah**

Indonesia merupakan sebuah negara yang memiliki berbagai macam kekayaan hayatinya, dan juga negara dengan jumlah pulau terbanyak di dunia. Salah satu dari kekayaan hayati yang dimiliki oleh Indonesia adalah ikan. Ikan merupakan bahan pangan yang paling banyak dikonsumsi oleh masyarakat Indonesia sejak zaman dahulu. Terbukti dengan banyaknya kuliner dengan bahan baku ikan di Indonesia. Selain menjadi bahan pangan masyarakat Indonesia juga memelihara ikan sebagai hiasan yang mereka taruh di dalam kolam ataupun akuarium.

Terdapat beberapa cara untuk mendapatkan ikan yang pertama adalah menangkapnya pada habitatnya. Cara ini sangat efektif, cepat dan dapat dilakukan setiap hari, namun dengan menangkap pada habitatnya setiap hari akan mengakibatkan terjadi overfishing yang menyebabkan ikan berkurang dari habitat asli-nya. Cara kedua adalah membudidayakan ikan. Budidaya ikan membutuhkan infrastruktur yang mendukung seperti lahan, kolam atau tambak yang memadai, bibit ikan yang akan dibudidayakan, dan serta pakan ikan dalam jumlah yang banyak. Walaupun mengeluarkan modal yang cukup besar dalam budidaya ikan tetapi budidaya ikan juga memberikan hasil yang sangat besar juga.

Ikan hasil dari budidaya ikan dapat dijual hidup dari habitat aslinya, hal ini sangat karena banyak dari rumah makan atau restoran yang membutuhkan ikan hidup untuk menjaga kualitas dari makanan yang mereka sajikan. Ikan hidup juga mempunyai rasa yang berbeda dari ikan yang sudah mati pada saat dijual. Selain membudidayakan ikan sebagai bahan pangan, ada juga ikan yang dibudidayakan sebagai hiasan. Secara tidak langsung membudidayakan ikan telah mengurangi terjadinya overfishing. Pemerintah Indonesia mencatat pada tahun 2022 produksi ikan mencapai 17,76 juta ton dengan 5,57 juta ton dari budidaya ikan dan 5,97 juta ton dari hasil tangkap laut dan perairan umum daratan.

Pembudidayaan ikan di Indonesia sangat besar, tetapi memiliki masalah yang sama besarnya yaitu diperlukannya usaha untuk menghitung dan mengawasi jumlah ikan yang ingin dibudidayakan. Salah satu contohnya adalah penghitungan bibit ikan, dalam hal tersebut penghitungan bibit lele oleh para pedagang masih menggunakan cara manual (Al-Amri, 2020). Ikan dipindahkan kedalam sebuah wadah dan mengukur besar ikan dengan alat yang telah dibuat, lalu dipindahkan lagi satu persatu atau dihitung sesuai dengan berat untuk mendapatkan jumlah ikan. Cara-cara seperti tadi sangat tidak efisien dan atau tidak akurat, ikan yang diukur lalu dihitung satu persatu memerlukan waktu yang sangat lama bila memerlukan jumlah yang sangat banyak. Sementara teknik penimbangan hanya memberikan hasil perkiraan jumlah yang akurasi masih perlu dipertanyakan.

Masalah sangat penting dimana budidaya ikan sangat mementingkan

banyak-nya jumlah ikan dalam tepat budidaya-nya, namun populasi ikan yang berlebihan akan memperlambat pertumbuhan ikan (Diansari et al, 2013), tapi disisi lain populasi ikan yang sangat sedikit akan mengurangi efisiensi lahan yang dimiliki oleh peternak ikan. Dalam hal ini Al-Amri (2020) menciptakan sebuah sistem dapat menghitung jumlah ikan dengan menggunakan sensor proximity. Hasil dari uji coba yang didapat sangat baik dengan presentasi error rata-rata sebesar 4,07 dengan waktu selama 228 detik untuk 1000 bibit ikan, sedangkan jika dihitung secara manual akan membutuhkan waktu 20 menit untuk 1000 bibit ikan.

Deteksi objek (Object detection) adalah salah satu dari visi komputer. Salah satu objektif nya adalah mengetahui lokasi dari sebuah objek pada gambar atau video. Pada penelitian Alim H (2021) telah membuat sebuah tracking movement ikan dengan menggunakan metode GMM dan Kalman filter penggunaan metode tersebut dapat memungkinkan pendeteksian dan mengamati pergerakan ikan, lalu pada penelitian Nugraha B (2022) pengestraksi gambar menggunakan metode grabcut memudahkan mengekstrak gambar ikan dalam sebuah citra ikan. Namun kedua penelitian sebelumnya masih sangat general hanya untuk mendeteksi ikan saja dan masih belum memiliki fungsi lainnya.

Pencocokan gambar adalah aspek dasar dari banyak permasalahan di dalam komputer, termasuk pendeteksian benda atau pemandangan, memecahkan bangunan 3D dari banyak gambar, dan pelacakan Gerakan (D.G. Lowe, 2005). SIFT atau Scale Invariant feature transform (D.G. Lowe, 1999) pendekatan ini mengubah gambar menjadi kumpulan besar vektor fitur lokal, yang masing-masing tidak berubah terhadap terjemahan, penskalaan, dan rotasi gambar, dan sebagian tidak berubah terhadap perubahan iluminasi dan proyeksi affine atau 3D. SIFT sudah banyak digunakan untuk mencocokkan gambar seperti Lokalisasi dan pemetaan dengan robot (Se, S et al, 2001), penyatuan panorama (Brown et al, 2003) dan lain-lain. SIFT di-identifikasi secara efisien dengan menggunakan pendekatan pemfilteran bertahap.

Tahap pertama mengidentifikasi lokasi kunci dalam ruang skala dengan mencari lokasi yang maksimal atau minimal dari fungsi Different of Gaussian. Setiap titik digunakan untuk menghasilkan vektor fitur yang mendeskripsikan wilayah gambar lokal yang diambil sampelnya relatif terhadap bingkai koordinat ruang-skalanya. Fitur mencapai invarian parsial terhadap variasi lokal, seperti proyeksi affine atau 3D, dengan mengaburkan lokasi gradien gambar. Detection of local features invariant to affine transformations (K. Mikolajczyk, 2002) menjelaskan penggunaan dua kombinasi detektor yaitu Harris dan Laplacian-of-Gaussians. Kombinasi Harris detector memberikan hasil yang lebih baik, Laplacian memungkinkan pemilihan skala karakteristik untuk titik yang diekstraksi dengan harris detector, dengan demikian descriptor dihitung pada lingkungan titik yang sama dalam gambar dengan resolusi yang berbeda, dan oleh karenanya tidak varian ke perubahan skala besar.

Berdasarkan latar yang telah dijelaskan, penulis mengusulkan untuk mengukur panjang serta menghitung berat ikan dengan menggunakan metode *Harris-Corner Detection*. Metode ini menggunakan dua kombinasi detektor yaitu



Harris-detector dan Laplace-of-Gaussians. Hasil yang diharapkan adalah sebuah sistem yang mampu mendeteksi ikan pada gambar secara akurat dan menghitung beratnya.

## 1.2 Rumusan Masalah

Dari uraian permasalahan diatas, perumusan masalah dalam penelitian ini adalah **“Bagaimana cara untuk mengukur panjang serta menghitung berat ikan menggunakan metode *Harris-Corner Detection*?”**

## 1.3 Batasan Masalah

1. Penghitungan panjang dan berat dari pada ikan dengan menggunakan *harris-laplace detection* sebagai pendeteksi benda.
2. Penelitian dilakukan sampai mendapatkan hasil panjang dan berat dari ikan.
3. Gambar atau Data yang akan digunakan adalah Ikan lele, Ikan Mas, dan Ikan Nila.
4. Citra yang digunakan hanya citra ikan tanpa menggunakan latar belakang atau transparan.
5. Bahasa Pemrograman yang digunakan adalah python 3.

## 1.4 Tujuan Penelitian

Tujuan dari Penelitian adalah Membangun sebuah sistem yang mampu menghitung panjang dan berat ikan dengan menggunakan *Harris-corner detection*.

## 1.5 Manfaat Penelitian

1. Bagi penulis Memperoleh gelar sarjana dalam bidang Ilmu Komputer, serta menambah pengalaman dalam pembangunan sebuah sistem operasi komputer dengan aplikasi dunia nyata, dan menambah pengetahuan tentang pendeteksian sudut atau korner dari *Harris-corner Detection*.
2. Bagi Universitas Negeri Jakarta

## BAB II

### KAJIAN PUSTAKA

#### 2.1 Multi-scale representation

Pada bagian ini akan menjelaskan representasi citra multi-resolusi berdasarkan filter Gaussian. Turunan Gaussian sering digunakan untuk mengekstrak fitur-fitur karakteristik. Bagian ini akan menjelaskan bagaimana menghitung stabilitas suatu gambar dan cara mengnormalisasi gambar tersebut untuk mendapatkan respons turunan yang tidak tergantung pada resolusi. Fitur lokal didefinisikan berdasarkan lokasi, skala, dan bentuk, yang akan mengalami transformasi saat dilihat dari sudut yang berbeda. Untuk memperkirakan perubahan transformasi pada suatu titik penting, dapat dilakukan eksplorasi sifat dari *Second moment matrix*, dengan komponen dari matriks Hessian digunakan untuk mendeteksi karakteristik skala struktur lokal dan mendeskripsikan bentuk struktur.

##### 2.1.1 Gaussian scale-space

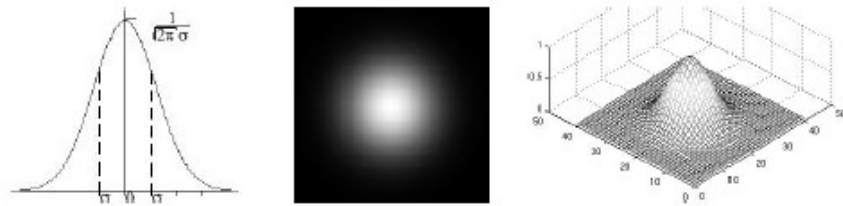
Dalam domain diskrit dari gambar digital, parameter skala juga diambil dalam bentuk diskrit. Sebagai hasilnya, representasi ruang-skala adalah kumpulan gambar yang direpresentasikan pada tingkat resolusi diskrit yang berbeda. Banyak penelitian dalam konteks skala-ruang menunjukkan bahwa salah satu persyaratan penting adalah bahwa representasi skala-ruang harus memenuhi persamaan yang dapat dicapai dengan melakukan konvolusi menggunakan kernel Gaussian. Selanjutnya, penelitian ini mengindikasikan bahwa kernel Gaussian adalah satu-satunya kernel unik yang dapat menghasilkan representasi skala-ruang. Keunikan dari kernel Gaussian ini telah dikonfirmasi melalui berbagai formulasi dalam berbagai penelitian lain. Penemuan ini menyimpulkan bahwa konvolusi dengan kernel Gaussian merupakan solusi optimal untuk permasalahan konstruksi representasi skala-ganda. Fungsi Gaussian dua dimensi didefinisikan sebagai berikut:

$$g(\sigma) = \frac{1}{2\pi\sigma_x\sigma_y} \exp^{-\frac{(x^2 + \mu_x) + (y^2 + \mu_y)}{2\sigma_x\sigma_y}}$$

Jika  $\mu_x = \mu_y = 0$  dan  $\sigma_x = \sigma_y$  maka dapat disederhanakan menjadi :

$$g(\sigma) = \frac{1}{2\pi\sigma^2} \exp^{-\frac{x^2 + y^2}{2\sigma^2}} \quad (2.1)$$

Keunikan dari gaussian kernel dapat diketahui dari beberapa sifat berikut : *linearity, separability, causality, and semi group property*. Keterpisahan atau *separability* memungkinkan *multi-dimensional gaussian kernel* untuk diperoleh sebagai produk satu dimensi kernel :



Gambar 2.1

$$g(x, y) = g(x)g(y)$$

Sifat ini memiliki kegunaan yang sangat signifikan dalam praktiknya karena penghalusan sinyal dua dimensi dapat digantikan oleh dua penghalusan satu dimensi, satu untuk setiap dimensinya. Filter Gaussian satu dimensi dapat diimplementasikan sebagai filter rekursif, yang secara signifikan mempercepat proses perhitungan terutama pada kasus kernel Gaussian yang lebih besar (yaitu  $> \sqrt{2}$ ). Kondisi kausalitas menyatakan bahwa tidak ada struktur buatan tambahan yang perlu dibuat saat menghitung citra dalam skala kasar, yang berarti citra pada skala yang lebih kasar merupakan representasi yang disederhanakan dari citra pada skala yang lebih halus. Sifat semi-grup komutatif menyatakan bahwa pemulusan sebanyak  $n$  kali pada sebuah gambar menghasilkan hasil yang sama dengan pemulusan satu kali menggunakan kernel dengan ukuran yang sama dengan jumlah seluruh  $n$  kernel yang digunakan. Selain itu, operasi  $n$  dapat dilakukan dalam urutan apa pun:

$$g(\sigma_1) * \dots * g(\sigma_n) * I(x) = g(\sigma_1 + \dots + \sigma_n) * I(x)$$

Biasanya, ruang-skala yang seragam digunakan, tetapi representasi ruang-skala umum dihitung dengan filter lain.

**Uniform scale-space.** Perbeddan tingkat pada representasi skala-ruang secara umum, terbuat dari konvolusi dengan *gaussian kernel* :

$$L(x, \sigma) = g(\sigma) * I(x) \quad (2.2)$$

Dengan  $I$  adalah gambar dan  $x = (x, y)$  adalah lokasi poin. *Kernel* adalah simetris sirkular dan diparameterisasi dari satu skala faktor  $\sigma$ .

Citra dalam skala kasar diperoleh dengan menghaluskan citra dalam skala halus. Proses ini diulangi secara berurutan pada tingkat skala yang semakin kasar untuk menghasilkan representasi multiskala. Untuk mempercepat operasi, seringkali dilakukan pengambilan sampel pada citra skala kasar dengan faktor skala yang sesuai setelah setiap operasi pemulusan. Perlu berhati-hati dalam memilih skala dan faktor pengambilan sampel karena hal ini dapat menyebabkan masalah aliasing. Selain itu, diperlukannya penambahan hubungan untuk mencocokkan lokasi titik yang sesuai pada tingkat skala yang berbeda. Ini membuat analisis teoretis menjadi lebih kompleks.

Alternatifnya, representasi skala-ruang juga dapat dibangun dengan meratakan gambar beresolusi tinggi secara berturut-turut dengan menggunakan kernel berbagai skala. Memang, membangun ruang-skala ini memerlukan lebih banyak waktu, terutama ketika tidak ada pengambilan sampel. Walaupun informasi menjadi redundant, namun tidak perlu menghitung lokasi titik yang sesuai pada setiap tingkat skala. Jika semua titik tetap dipertahankan pada setiap tingkat ruang-skala, maka hubungan antara analisis teoretis dan komputasi praktis tetap terjaga.

Detektor fitur, pada dasarnya, bergantung pada struktur sederhana seperti tensor matriks momen kedua dalam deteksi titik penting. Skala kepentingan dalam dan luar, yaitu di mana struktur dapat dianalisis, ditentukan oleh kondisi akuisisi, yaitu resolusi dan lapangan pandang. Ukuran minimal sebuah struktur gambar yang dapat dianggap sebagai karakteristik dibatasi oleh resolusi dan noise. Inner scale adalah faktor yang berhubungan dengan ukuran struktur tersebut, dengan kata lain, ukuran minimal lingkungan titik yang mengandung informasi penting tentang struktur. Outer scale, yang merupakan ukuran terbesar dari struktur, dibatasi oleh kendala yang menentukan sifat lokal dari fitur tersebut dan juga dibatasi oleh ukuran gambar itu sendiri. Faktor skala harus didistribusikan secara eksponensial antara batas dalam dan luar, yaitu  $\sigma_n = \sigma_0 s^n$ , agar perubahan informasi tetap seragam antara tingkat resolusi yang berurutan.

### 2.1.2 Scale-space derivatives

Kegunaan turunan dalam analisis sinyal lokal dapat diilustrasikan melalui ekspansi Taylor. Ekspansi Taylor memegang peran penting dalam desain filter dan, pada urutan yang dipilih, mendekati struktur gambar secara lokal. Secara umum, ekspansi ini dihitung hingga urutan kedua.

$$I(x_0 + \Delta x) \approx I(x_0) + \Delta x^T \nabla I(x_0) + \Delta x^T H(x_0) \Delta x$$

**Gaussian derivatives** Sebuah fitur dapat diekstrak pada berbagai tingkat resolusi dengan menerapkan fungsi yang sesuai pada skala yang berbeda. Fungsi deteksi banyak didasarkan pada turunan Gaussian dalam ruang-skala, karena turunan Gaussian linear cocok untuk memodelkan pemrosesan visual dalam manusia. Perbedaan besar ini, yang dihitung dengan menggabungkan sinyal asli dengan turunan dari Gaussian, dapat dianggap sebagai generalisasi operator perbedaan. Ketika parameter skala mendekati nol, turunan dalam ruang-skala mendekati turunan dari fungsi asli. Tujuan dari analisis dalam ruang-skala adalah untuk menyelidiki representasi citra pada berbagai tingkat skala guna mengekstrak informasi yang signifikan.

**Uniform derivatives** Secara umum, dalam praktik pemrosesan citra, filter yang berasal dari kernel Gaussian seragam sering digunakan. Turunan pada berbagai tingkat skala dapat dihitung dengan menghaluskan citra menggunakan filter Gaussian dan kemudian menghitung perbedaannya. Semua properti yang berlaku

untuk kernel Gaussian juga berlaku untuk turunannya. Oleh karena itu, jika menerapkan operasi ini dalam urutan terbalik, akan mendapatkan hasil yang sama. Alternatifnya, didapat dengan menggabungkan citra dengan turunan dari kernel Gaussian pada berbagai skala. Semua metode ini bersifat setara. Dalam hal ini, untuk setiap fungsi citra  $I(x)$ , turunan pertama dapat didefinisikan sebagai berikut:

$$L_x(x; \Sigma) = \frac{\partial}{\partial_x} * g(\Sigma) * I(x)$$

Persamaan umum untuk turunan Gauss adalah sebagai berikut:

$$g_{i_1 \dots i_m}(x, \Sigma) = \frac{\partial}{\partial_{i_1} \dots \partial_{i_m}} \frac{1}{2\pi \sqrt{\det}} \exp^{-\frac{x^T \Sigma^{-1} x}{2}}$$

Di mana  $m$  adalah urutan turunan dan  $i$  adalah koordinat Cartesius pada citra. Dalam kasus di mana  $\Sigma$  adalah matriks identitas dikalikan dengan skalar, maka akan berurusan dengan turunan Gaussian seragam tradisional.

**Normalized derivatives** Amplitudo turunan spasial, secara umum, berkurang dengan skala karena responsnya menjadi lebih halus pada skala yang lebih besar. Namun, dalam kasus struktur yang ada pada rentang skala besar, seperti *corner* atau *step-edge*, diharuskan menjaga konstan turunan terhadap skala. Untuk menjaga properti *scale invariance*, fungsi turunan harus dinormalisasi dalam hubungannya dengan skala turunan. Turunan spasial  $D$  yang dinormalisasi skala orde  $m$  dapat didefinisikan sebagai berikut:

$$D_{i_1 \dots i_m}(x, \sigma) = \sigma^m L_{i_1 \dots i_m}(x, \sigma) = \sigma^m g_{i_1 \dots i_m}(\sigma) * I(x)$$

Persamaan berikut menunjukkan pentingnya penggunaan faktor normalisasi  $\sigma^m$ . Pertimbangkan dua citra,  $I$  dan  $I'$ , yang diambil pada skala yang berbeda. Hubungan antara dua citra ini dapat didefinisikan sebagai:  $I(x = I'(x'))$ , di mana  $x' = sx$ . Perlu diperhatikan bahwa pergeseran potensial suatu titik diabaikan dalam konteks ini karena akan tereliminasi oleh operasi diferensiasi. Turunan dari citra ini dapat didefinisikan sebagai berikut:

$$I_{i_1 \dots i_m}(x') = s^m I_{i_1 \dots i_m}(sx')$$

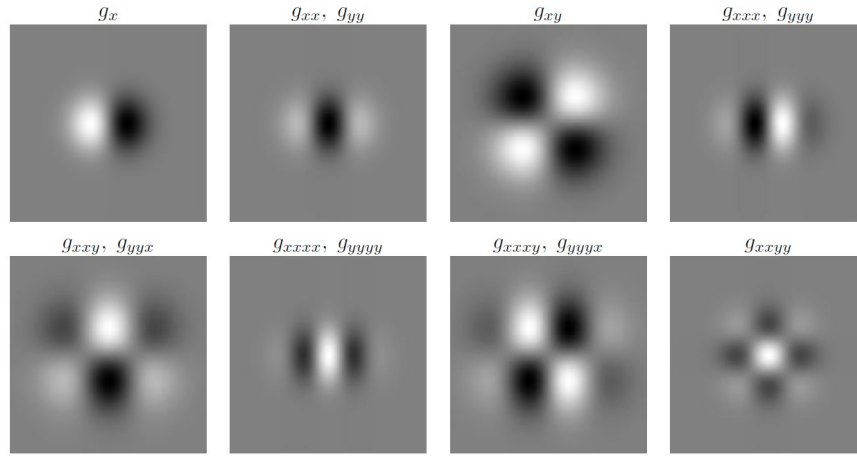
Jika menganggap bahwa kernel turunan dari skala  $\sigma$  dinormalisasi oleh faktor skala yang sama, maka memperoleh:

$$\sigma^m g_{i_1 \dots i_m}(\sigma) * I(x) = s^m \sigma^m g_{i_1 \dots i_m}(s\sigma) * I(x')$$

Jadi, untuk turunan yang dinormalisasi, responsnya memiliki nilai yang sama:

$$D_{i_1 \dots i_m}(x, \sigma) = D'_{i_1 \dots i_m}(x, s\sigma)$$

Dengan cara ini, dapat dilihat bahwa dengan mengalikan turunan dengan ukuran kernel yang sesuai, maka akan mendapatkan nilai turunan yang sama untuk



**Gambar 2.2**

struktur lokal yang direpresentasikan pada skala yang sesuai.

**Affine derivatives** Turunan affine sangat bermanfaat ketika berurusan dengan invarian affine. Untuk mencapai invarian affine dari turunan struktur lokal yang tidak terbatas, kernel Gaussian harus disesuaikan dengan bentuk dan skala struktur yang ada. Dengan cara ini, bentuk anisotropik dari fitur tidak akan terpengaruh oleh penghalusan isotropik yang dihasilkan oleh filter seragam. Mengadaptasi filter tanpa pengetahuan sebelumnya tentang bentuk struktur tersebut memerlukan eksplorasi banyak kemungkinan kombinasi parameter kernel. Namun, tiga derajat kebebasan dalam kernel affine Gaussian membuatnya sulit untuk mengeksplorasi semua kemungkinan kombinasi tersebut. Oleh karena itu, dalam praktiknya, sering kali dibatasi kemungkinan bentuk kernel yang akan digunakan. Selain itu, komputasi turunan arah affine untuk semua titik citra dapat dipercepat dengan implementasi yang bersifat rekursif.

### 2.1.3 Second moment matrix

*Second Moment matrix* dijelaskan pada bagian ini sering digunakan untuk deteksi fitur atau deskripsi struktur citra lokal. Matriks ini disebut juga *Auto-correlation Matrix* dan didefinisikan dengan:

$$\mu(x, \sigma_I, \sigma_D) = \begin{bmatrix} \mu_{11} & \mu_{12} \\ \mu_{21} & \mu_{22} \end{bmatrix} = \sigma_D^2 g(\sigma_I) * \begin{bmatrix} L_x^2(x, \sigma_D) & L_x L_y(x, \sigma_D) \\ L_x L_y(x, \sigma_D) & L_y^2(x, \sigma_D) \end{bmatrix} \quad (2.3)$$

Ini menjelaskan distribusi gradien di lingkungan lokal dari suatu titik. Derivatif gradien ditentukan oleh skala lokal  $\sigma_D$  (skala derivasi). Derivasi dirata-ratakan di sekitar titik dengan menghaluskan menggunakan jendela Gaussian berukuran  $\sigma_I$  (skala integrasi). Nilai eigen dari matriks ini merepresentasikan dua kelengkungan utama dari suatu titik. Properti ini memungkinkan ekstraksi titik, yang

kedua kelengkungannya signifikan, yaitu perubahan sinyal signifikan dalam arah ortogonal. Titik-titik ini lebih stabil dalam kondisi pencahayaan acak dan lebih representatif daripada titik gambar lainnya. Dalam ruang-skala affine, *Second moment matrix*  $\mu$  pada titik tertentu  $x$  didefinisikan oleh:

$$\mu(x, \Sigma_I, \Sigma_D) = \det(\Sigma_D) g(\Sigma_I) * ((\nabla L)(x, \Sigma_D)(\nabla L)(x, \Sigma_D)^T)$$

dimana  $\Sigma_I$  dan  $\Sigma_D$  adalah matriks kovarians yang menentukan integrasi dan derivasi kernel Gaussian. Dalam praktik tidak mungkin menghitung matriks untuk semua kemungkinan kombinasi parameter kernel. Untuk membatasi jumlah derajat kebebasan dengan menerapkan kondisi  $\Sigma_I = s\Sigma_D$ , dimana  $s$  adalah skalar. Oleh karena itu, derivasi dan integrasi kernel akan berbeda hanya dalam ukuran dan bukan dalam bentuk. Ini berarti faktor skala antara dua arah ortogonal sama untuk menghaluskan dan mengintegrasikan turunan dari *second moment matrix*.

**Affine transformation of a point** Matriks momen kedua memiliki sifat yang membuatnya sangat berguna untuk memperkirakan bentuk anisotropik dari struktur citra lokal. Pertimbangkan sebuah titik  $x_L$  yang diubah oleh transformasi linear  $x_R = Ax_L$ . Matriks  $\mu_L$  yang dihitung pada titik  $x_L$  kemudian ditransformasikan dengan cara berikut:

$$\mu(x_L, \Sigma_{I,L}, \Sigma_{D,L}) = A^T \mu(Ax_L, A\Sigma_{I,L}A^T, A\Sigma_{D,L}A^T)A = A^T \mu(x_R, \Sigma_{I,R}, \Sigma_{D,R})A \quad (2.4)$$

Jika menunjukkan matriks yang sesuai dengan:

$$\mu(x_L, \Sigma_{I,L}, \Sigma_{D,L}) = M_L \quad \mu(x_R, \Sigma_{I,R}, \Sigma_{D,R}) = M_R$$

matriks-matriks tersebut kemudian dihubungkan dengan:

$$M_L = A^T M_R A \quad M_R = A^{-T} M_L A^{-1} \quad (2.5)$$

Derivasi dan kernel integrasi dalam hal ini diubah oleh:

$$\Sigma_R = A\Sigma_L A^T$$

Misalkan matriks  $M_L$  dihitung sedemikian rupa sehingga (Kondisi 1):

$$\Sigma_{I,L} = \sigma_I M_L^{-1} \quad \Sigma_{D,L} = \sigma_D M_L^{-1} \quad (2.6)$$

dimana skalar  $\sigma_I$  dan  $\sigma_D$  masing-masing adalah skala integrasi dan derivasi. Selanjutnya dapat menurunkan sebagai berikut (Kondisi):

$$\Sigma_{I,R} = A\Sigma_{I,L}A^T = \sigma_I (AM_L^{-1}A^T) = \sigma_I (A^{-T}M_L A^{-1})^{-1} = \sigma_I M_R^{-1} \Sigma_{D,R} = A\Sigma_{D,L}A^T = \sigma_D (AM_L^{-1}A^T) \quad (2.7)$$

Dapat dilihat bahwa memaksakan kondisi 1 memerlukan hubungan kondisi 2 dengan asumsi bahwa titik-titik tersebut terkait dengan transformasi affine. Sekarang dapat membalikkan masalahnya dan menganggap memiliki dua titik yang dihubungkan oleh transformasi affine yang tidak diketahui. Jika diestimasi matriks  $\Sigma_R$  dan  $\Sigma_L$  sehingga matriks tersebut memverifikasi kondisi 1 dan 2, maka relasi 3.4 benar. Properti yang disajikan memungkinkan parameter transformasi diekspresikan secara langsung oleh komponen matriks. Transformasi affine kemudian bisa didefinisikan oleh :

$$A = M_R^{-\frac{1}{2}} R M_L^{-\frac{1}{2}}$$

dimana  $R$  mewakili rotasi sewenang-wenang. Pada bab berikutnya akan disajikan algoritma iteratif untuk memperkirakan matriks  $\Sigma_R$  dan  $\Sigma_L$  dan bagaimana memulihkan rotasi  $R$  dengan cara yang kuat. Dengan demikian, dapat memperkirakan transformasi afin antara dua titik yang bersesuaian tanpa pengetahuan sebelumnya tentang transformasi ini. Selanjutnya, matriks  $M_L$  dan  $M_R$ , dihitung pada kondisi 3.5 dan 3.6, menentukan daerah korespondensi yang didefinisikan oleh  $X^T M_x = 1$ . Jika ketetanggaan titik  $x_R$  dan  $x_L$  dinormalisasi dengan transformasi  $x'_R = M_R^{\frac{1}{2}} x_R$  dan  $x'_L = M_L^{\frac{1}{2}} x_L$ , masing-masing, daerah yang dinormalisasi dihubungkan dengan rotasi sederhana  $x'_L = R x'_R$

$$x_R = A x_L = M_R^{-\frac{1}{2}} R M_L^{\frac{1}{2}} x_L, \quad M_R^{\frac{1}{2}} x_R = R M_L^{\frac{1}{2}} x_L \quad (2.8)$$

Matriks  $M'_L$  dan  $M'_R$  dalam bingkai yang dinormalisasi sama dengan matriks rotasi murni. Dengan kata lain, pola intensitas dalam bingkai yang dinormalisasi bersifat isotropik

**Isotropy measure** Berikut ini adalah menginterpretasikan *second moment matrix*, yang disajikan di atas, dalam kaitannya dengan ukuran isotropi. Tanpa kehilangan keumuman, kami menganggap bahwa struktur anisotropik lokal adalah struktur isotropik yang ditransformasi *affine*. Ini memberikan solusi untuk masalah deformasi affine pola lokal bila dilihat dari sudut yang berbeda. Struktur isotropik yang dideformasi oleh transformasi affine menjadi anisotropik. Untuk mengimbangi deformasi affine, diharuskan menemukan transformasi yang membawa pola anisotropik ke isotropik. Perlu diperhatikan bahwa mempertahankan rotasi isotropi atau anisotropi dari patch citra. Oleh karena itu, deformasi afin struktur isotropik dapat ditentukan hingga faktor rotasi. Faktor ini dapat dipulihkan dengan metode lain. *second moment matrix*  $\mu(x, \sigma_I, \sigma_D)$  dapat diartikan sebagai ukuran isotropi yang diterapkan pada titik  $x$  dalam lingkungan lokal berukuran  $\sigma_I$ . Isotropi lokal dapat diukur oleh nilai eigen matriks  $\mu$ . Jika nilai eigennya sama, dapat dianggap titik tersebut isotropik. Untuk memperoleh ukuran yang dinormalisasi menggunakan rasio nilai eigen:

$$Q = \frac{\lambda_{\min}(\mu)}{\lambda_{\max}(\mu)} \quad (2.9)$$



Nilai  $Q$  bervariasi dalam kisaran  $|0 \cdots 1|$  dengan 1 untuk struktur isotropik yang sempurna. Pengukuran ini dapat memberikan respons yang sedikit berbeda untuk skala yang berbeda karena matriks  $\mu$  ditentukan oleh dua parameter skala. Skala-skala ini harus dipilih secara independen dari resolusi gambar.

Teknik pemilihan skala yang dijelaskan pada bagian berikutnya, memberikan kemungkinan untuk menentukan skala integrasi yang terkait dengan struktur citra lokal. Skala derivasi dan integrasi dapat dihubungkan dengan  $\sigma_D = s\sigma_I$ , dengan  $s$  adalah faktor konstanta. Untuk alasan yang jelas skala derivasi harus selalu lebih kecil dari skala integrasi. Faktor  $s$  tidak boleh terlalu kecil, jika tidak smoothing terlalu signifikan terhadap derivasi. Di sisi lain  $s$  harus cukup kecil, sehingga  $\sigma_I$  dapat merata-ratakan matriks kovarians  $\mu(x, \sigma_D, \sigma_I)$  di lingkungan sekitar. Idenya adalah untuk menekan noise tanpa menekan bentuk anisotropik dari struktur gambar yang diamati. Pendekatan yang lebih canggih adalah memilih skala derivasi  $\sigma_D$  secara independen dari skala  $\sigma_I$ . Mengingat skala integrasi dapat mencari skala  $\sigma_D$ , dimana respon dari ukuran isotropi mencapai maksimum lokal. Dengan demikian, bentuk yang dipilih untuk struktur yang diamati kurang anisotropik. Pendekatan serupa untuk memilih skala lokal tetapi dia mengusulkan pemilihan skala yang anisotropinya dinormalisasi diasumsikan maksimum melebihi skala.

$$Q_A = \frac{\sqrt{\text{trace}^2 \mu - 4 \det \mu}}{\text{trace} \mu} \quad (2.10)$$

Ukuran ini juga dapat dinyatakan oleh nilai eigen:

$$Q_A = \left| \frac{\lambda_{\max}(\mu) - \lambda_{\min}(\mu)}{\lambda_{\max}(\mu) + \lambda_{\min}(\mu)} \right|$$

Perhatikan kesamaan antara  $Q$  dan  $Q_A$ . Meskipun,  $Q_A$  cenderung nol jika titiknya menjadi lebih isotropik. Berlawanan dengan pendekatan, pola gambar tidak dinormalisasi afin dalam prosedur iteratif yang memperkirakan bentuk anisotropik. Selanjutnya, dalam percobaan selanjutnya, melihat bahwa prosedur ini menyimpang lebih sering jika skala lokal dipilih dengan ukuran  $Q_A$  maksimum.

## 2.2 Automatic scale selection

Skala invariansi merupakan salah satu tujuan dalam pekerjaan ini, oleh karena itu pada bagian ini kami fokus pada metode untuk menentukan skala struktur citra lokal. Pemilihan skala otomatis dan sifat dari skala yang dipilih telah dipelajari secara ekstensif. Idenya adalah untuk memilih skala karakteristik, dimana fungsi tertentu mencapai skala ekstrem. Karakteristik nama sedikit arbitrer karena struktur lokal dapat eksis pada rentang skala dan dalam rentang ini tidak ada skala persepsi yang disukai. Namun, untuk deskriptor tertentu sebuah skala dapat diberi nama karakteristik, jika deskriptor yang dihitung pada skala ini dapat menyampaikan informasi lebih banyak dibandingkan dengan deskriptor pada skala lain.

### 2.2.1 Scale-space maxima

Berikut ini disajikan hubungan analitis antara skala maksimum ruang dan skala struktur citra lokal. Biarkan  $F$  menjadi fungsi yang dinormalisasi sehubungan dengan skala, yang digunakan untuk membangun ruang-skala. Himpunan respons untuk titik  $x$  kemudian ditentukan oleh  $F(x, \sigma_n)$  dengan  $\sigma_n = \sigma_0 s^n$ . Himpunan ini disebut jejak skala. Faktor  $\sigma_n$  adalah skala awal pada tingkat resolusi terbaik dan  $\sigma_n$  menunjukkan tingkat representasi skala-ruang yang berurutan. Parameter memungkinkan peningkatan skala secara eksponensial untuk memperoleh perubahan informasi yang seragam antara level skala.

Baris atas dari frame pada gambar 2.2 menunjukkan konfigurasi sinyal teoritis, dimana Laplacian dan gradien dihitung. Gambar menyajikan kernel diferensiasi di sekitar sudut atau tepi. Di baris tengah kami menampilkan jejak skala untuk konfigurasi sinyal yang sesuai. Perhatikan bahwa untuk beberapa di antaranya, gradien tidak mencapai titik ekstrem. Hal ini terjadi untuk titik di dekat sudut dan tepi jika tidak ada perubahan sinyal lain di sekitar. Perhatikan bahwa ekstrem dari gradien juga kurang berbeda dari yang diperoleh untuk Laplacian. Dapat berharap bahwa dalam gambar nyata, Laplacian mencapai skala ekstrem lebih sering daripada gradien

Seperti yang ditunjukkan untuk beberapa konfigurasi sinyal yang ditampilkan pada Gambar 2.2 bahwa maksimum turunan yang dinormalisasi terkait dengan jarak dari perubahan sinyal. Kondisi yang diperlukan untuk menemukan skala ekstrem lokal adalah  $\frac{\partial}{\partial \sigma} F_{norm} = 0$  Diberikan sebuah fungsi yang merepresentasikan step-edge yang ditampilkan pada Gambar 2.2 :

$$f_{step-edge}(x) = \begin{cases} 0 & x < x_0 \forall y \\ 1 & x \geq x_0 \forall y \end{cases}$$

Dan dapat menghitung konvolusi sudut dengan Laplacian yang dinormalisasi berpusat pada titik  $(0, 0)$ :

$$f_{xx_{norm}}(x, \sigma) = \frac{x}{\sigma \sqrt{2\pi}} e^{\frac{-x^2}{2\sigma^2}}$$

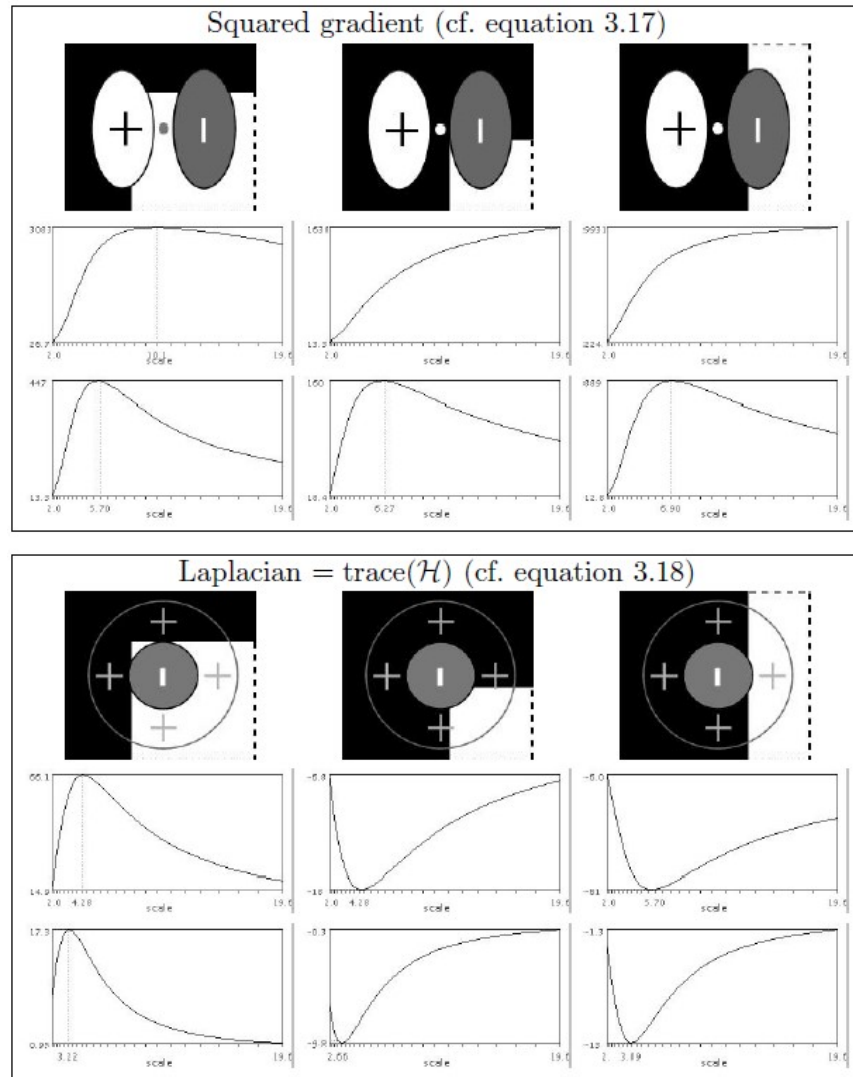
Ekstrem dari fungsi dapat ditemukan sebagai berikut:

$$\frac{\partial}{\partial \sigma} f_{xx_{norm}}(x, \sigma) = \frac{1}{\sqrt{2\pi}} e^{\frac{-x^2}{2\sigma^2}} \left( -\frac{x}{\sigma^2} + \frac{x^3}{\sigma^4} \right) = 0 \Leftrightarrow \sigma = |x_0| \quad (2.11)$$

Kondisi yang diperlukan untuk menemukan skala ekstrem lokal adalah  $\frac{\partial}{\partial \sigma} F_{norm} = 0$ . Diberikan sebuah fungsi yang merepresentasikan step-edge yang ditampilkan

$$\frac{\partial}{\partial \sigma} f_{xx_{norm}}(x, \sigma) = \frac{x^2}{\sigma^3 \sqrt{2\pi}} e^{\frac{-x^2}{2\sigma^2}} \neq 0 \forall \sigma \quad (2.12)$$

Untuk titik tertentu  $(x_0, y_0)$  Laplacian yang dinormalisasi mencapai ekstrem



**Gambar 2.3:** *Scale trace of squared gradient and Laplacian* diterapkan pada model sudut dan tepi. Kolom pertama: tunjuk di dalam sudut. Kolom kedua: titik di luar sudut. Kolom ketiga: arahkan di dekat tepi. Di bingkai: Baris atas: Model sudut dan tepi. Baris tengah: Jejak skala untuk  $\gamma = 1$ . Baris bawah: Jejak skala untuk  $\gamma = 0,5$ .

pada  $\sigma_{extremum} = |x_0|$ . Masalah dapat terjadi jika titiknya sama dengan  $x_0 = 0$ , yaitu maksimum turunan pertama. Untungnya, interest point yang dideteksi oleh operator biasanya diterapkan untuk tujuan ini, tidak terlokalisasi di  $x_0 = 0$ , tepatnya di persimpangan sudut. Hal ini disebabkan oleh skala deteksi, yang dalam praktiknya tidak boleh sama dengan 0, dan seperti yang akan ditunjukkan di bagian selanjutnya, titik minat mengubah lokasi sehubungan dengan skala deteksi. Operasi matematika terperinci dapat ditemukan di lampiran A.1. Tanggapan seperti itu terhadap tepi dan sudut juga dianalisis. Perhatikan bahwa skala linear koordinat  $x, y$  melibatkan skala linear yang sama dari  $\sigma_{extremum}$ . Dapat juga menunjukkan bahwa tidak ada skala ekstrem untuk turunan orde pertama dari sebuah step-edge.

Dengan demikian, dapat memulihkan skala fitur dengan melihat maksimum turunan orde kedua yang dinormalisasi.

### 2.2.2 Gamma normaozation

Faktor normalisasi, yang diterapkan untuk menghitung turunan skala-ruang, memiliki properti, yang dapat sangat berguna untuk parameterisasi mekanisme pemilihan skala. Pada bagian ini menganalisis pengaruh faktor normalisasi pada maxima lokal atas skala. Alih-alih menormalkan turunan dengan faktor  $\sigma^n$ , dimana  $n$  adalah urutan turunan, dan dapat menerapkan  $\sigma^{\gamma n}$ . Ketika  $\gamma = 1$  berurusan dengan invarian skala sempurna, yaitu amplitudo turunan yang dinormalisasi tidak tergantung pada resolusi sinyal. Invarian tidak selalu dipertahankan dalam kasus operator pemilihan skala berdasarkan kombinasi turunan dari pesanan yang berbeda. Jika operator tersebut digunakan, besaran jejak skala dapat berbeda untuk suatu titik yang direpresentasikan pada resolusi yang berbeda. Faktor normalisasi harus kemudian  $\gamma \neq 1$  untuk mempertahankan invarian besaran. Meskipun demikian, maxima over scale lokal tetap dipertahankan, bahkan untuk  $\gamma \neq 1$ . Diberikan  $\gamma$ -normalisasi operator deteksi, ada rentang tertentu dari nilai  $\gamma$  dimana struktur gambar spesifik telah menetapkan skala karakteristik. Ekstrem turunan atas skala memiliki kecenderungan untuk berpindah ke skala yang lebih rendah dengan penurunan faktor  $\gamma$ . Efek ini diilustrasikan pada baris tengah dan bawah pada Gambar 3.8 dan 3.9. Dengan menyetel  $\gamma < 1$  dapat memperoleh ekstrem untuk nilai  $\sigma$  yang lebih rendah. Oleh karena itu, dapat menjelajahi rentang skala yang lebih sempit saat mencari maksimum lokal. seorang peneliti menunjukkan bahwa untuk sinyal satu dimensi sederhana, periodik, seperti sinus atau kosinus, maksimum turunan normalisasi sesuai dengan panjang gelombang sinyal:

$$\sigma_{extremum} = \lambda \frac{\sqrt{\gamma m}}{2\pi} \quad (2.13)$$

dimana  $m$  adalah urutan turunan dan  $\lambda$  panjang gelombang. Ekspresi yang sesuai untuk turunan kedua yang berpusat pada fungsi Gaussian satu dimensi diberikan:

$$\sigma_{extremum} = \sigma_{Gaussian} \sqrt{\frac{\gamma}{\frac{3}{2} - \gamma}} \quad (2.14)$$

Perhatikan bahwa jika  $\gamma = \frac{3}{4}$ , maka  $\sigma_{extremum} = \sigma_{gaussian}$ . Untuk operator Laplacian yang diterapkan pada step-edge diperoleh relasi berikut:

$$\sigma_{extremum} = x \sqrt{3 - 2\gamma} \quad (2.15)$$

Oleh karena itu, tidak ada skala ekstrem untuk  $\gamma \geq \frac{3}{2}$ . Persamaan yang sesuai untuk turunan pertama adalah:

$$\sigma_{extremum} = x \sqrt{\frac{1}{1 - \gamma}} \quad (2.16)$$

Hal ini menunjukkan bahwa turunan pertama dapat mencapai suatu ekstrem tetapi faktor normalisasi harus  $\gamma < 1$ . Parameter  $\gamma$  juga dapat mengambil nilai negatif tetapi semakin rendah nilai  $\gamma$  semakin rendah besarnya ekstrem lokal dan oleh karena itu kurang khas. Perhatikan bahwa hubungan ini berlaku untuk struktur gambar teoritis yang sempurna. Dalam kasus gambar nyata, tekstur, yang sering muncul di sekitar sudut atau tepi, dapat mengubah skala jejak. Namun demikian, hubungan di atas menunjukkan kemampuan ekspresi diferensial untuk memilih skala struktur citra lokal dan juga menunjukkan pengaruh faktor normalisasi  $\gamma$  pada maksimum skala-ruang. Hasil percobaan membuktikan hubungan ini.

### 2.2.3 Differential expressions for scale selection

Turunan yang dihitung dalam koordinat Cartesian umumnya tidak terkait dengan struktur gambar, oleh karena itu operator struktur yang berguna dibuat dari kombinasi beberapa turunan Gaussian. Pada bagian ini menyajikan operator, yang sering digunakan dalam konteks pemilihan fitur lokal skala. Operator pemilihan skala setidaknya harus invarian terhadap rotasi untuk mempertahankan invarian minimum. Invariansi iluminasi kurang kritis karena fitur dilokalkan pada fungsi ekstrem lokal. Namun, seseorang harus berhati-hati karena saturasi dapat menimbulkan kesalahan. Lokalisasi ekstrem tidak bergantung pada perubahan iluminasi affine, hanya besaran respons yang berubah. Pemilihan skala dengan menggunakan gradient magnitude juga telah digunakan. Chomat menunjukkan bahwa operator gradien sesuai untuk memilih skala karakteristik fitur lokal dan tahan terhadap noise pada gambar.

$$squaredgradient \sigma_D^2(L_x^2(x, \sigma_D^2) + L_y^2(x, \sigma_D^2)) \quad (2.17)$$

Besarnya gradien secara alami tidak berubah terhadap rotasi dan fase dapat digunakan untuk menentukan orientasi dominan pada fitur lokal. Fungsi Laplacian simetris sirkular dan telah berhasil digunakan untuk deteksi gumpalan dan pemilihan skala otomatis.

$$\text{Laplacian } \sigma_D^2 |L_{xx}^2(x, \sigma_D^2) + L_{yy}^2(x, \sigma_D^2)| \quad (2.18)$$

Operator difference-of-Gaussian yang digunakan oleh Lowe adalah perkiraan dari Laplacian-of-Gaussian dan memungkinkan untuk mempercepat komputasi representasi ruang-skala.

$$\text{difference-of-Gaussian } |I(x) * g(\sigma_I) - I(x) * g(k\sigma_I)| \quad (2.19)$$

Pendekatan yang lebih canggih adalah memilih skala dimana jejak dan determinan matriks Hessian mengasumsikan ekstrem lokal.

$$\max(|\text{trace}(H)|) \text{ and } \max(|\det(H)|) \quad (2.20)$$

Pemilihan skala menggunakan determinan matriks Hessian digunakan. Jejak matriks Hessian sama dengan Laplacian, tetapi pemilihan maksimum determinan secara simultan menghasilkan titik-titik, yang nilai eigen matriksnya memiliki nilai yang sebanding dan besar. Titik-titik ini lebih kuat terhadap kebisingan dan perubahan iluminasi. Detektor titik bunga, yang diusulkan oleh Moravec, diperbaiki oleh Harris dan kemudian oleh Schmid et al., didasarkan pada gagasan yang sama, tetapi menggunakan komponen matriks momen kedua. Sebuah detektor yang sangat serupa juga dikembangkan oleh Förstner dan Gülch.

$$\text{Harris function } \det(\mu(x, \sigma_I, \sigma_D)) - \alpha \text{trace}^2(\mu(x, \sigma_I, \sigma_D)) \quad (2.21)$$

Namun, operator ini tidak diadaptasi untuk perubahan skala. Untuk menghadapi transformasi tersebut, Dufournaud et al. membuat parameter operator Harris berdasarkan skala. Hal ini memungkinkan titik-titik minat dideteksi pada skala yang berbeda.

## 2.3 Scale invariant detector

### 2.3.1 Harris-Laplace detector

Pada bagian ini kami mengusulkan detektor titik minat baru yang menggabungkan detektor Harris yang andal dan pemilihan skala berbasis Laplacian. Evaluasi detektor interest point yang disajikan menunjukkan keunggulan detektor Harris dibandingkan dengan pendekatan lain yang ada. Dalam eksperimen diperhatikan bahwa skala mengadaptasi fungsi Harris jarang mencapai maxima atas skala dalam representasi skala-ruang. Jika terlalu sedikit interest point yang terdeteksi, gambar tidak terwakili dengan andal. Oleh karena itu, tinggalkan ide pencarian 3D maxima dari fungsi Harris. Selanjutnya, percobaan menunjukkan bahwa fungsi LoG memungkinkan persentase tertinggi dari skala karakteristik yang

benar dapat ditemukan. Oleh karena itu, kami mengusulkan untuk menggunakan Laplacian untuk memilih skala titik yang diekstraksi dengan detektor Harris. Detektor Harris-Laplace menggunakan fungsi Harris untuk melokalkan titik di setiap tingkat representasi ruang-skala. Selanjutnya, ia memilih titik-titik yang Laplacian-of-Gaussian mencapai skala maksimum. Dengan cara ini, gabungkan kedua metode ini untuk mendapatkan detektor titik minat yang andal yang tidak berubah terhadap perubahan skala yang signifikan.

Berikut ini dijelaskan secara detail algoritma deteksi. Usulkan dua implementasi dari gagasan umum yang disajikan di atas. Yang pertama adalah algoritma cepat untuk mendeteksi lokasi titik minat dan skala wilayah karakteristik terkait. Yang kedua memberikan estimasi lokasi dan skala yang tepat dari setiap titik minat yang mungkin.

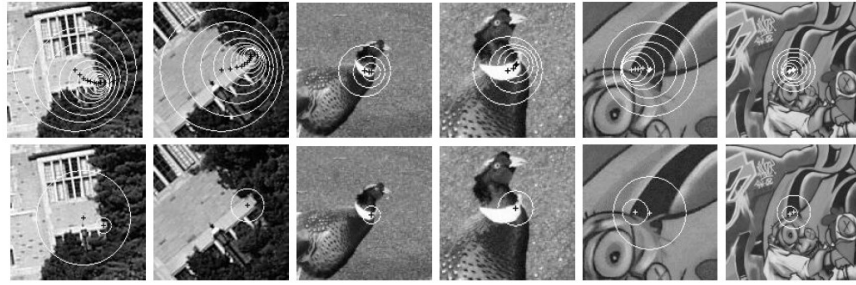
**Harris-Laplace.** Algoritma deteksi bekerja sebagai berikut. Pertama, bangun representasi ruang-skala dengan fungsi Harris untuk skala yang dipilih sembarang  $\sigma_n = s^n \sigma_0$ , dimana  $s$  adalah faktor skala antara tingkat yang berurutan. Pada setiap level representasi mengekstraksi interest point dengan mendeteksi maxima lokal di 8-neighbourhood dari sebuah titik  $x$ . Ambang batas digunakan untuk menolak maksimal sudut kecil, karena kurang stabil dalam kondisi tampilan yang berubah-ubah:

$$\det(\mu(x, \sigma_n)) - \alpha \text{trace}^2(\mu(x, \sigma_n)) > \text{threshold}_H \quad (2.22)$$

Matriks  $\mu(x, \sigma_n)$  sebenarnya dihitung dengan skala integrasi  $\sigma_I = \sigma_n$  dan skala lokal  $\sigma_D = k\sigma_n$ , dengan  $k$  adalah faktor konstan. Untuk mendapatkan kumpulan poin yang kompak dan representatif, verifikasi untuk setiap kandidat poin yang ditemukan pada level yang berbeda apakah itu membentuk maksimum dalam skala dimensi  $F(x, \sigma_n) > F(x, \sigma_I)$  dengan  $l \in n-1, n+1$  dan  $F(x, \sigma_n) > \text{threshold}$ . Laplacian-of-Gaussian digunakan untuk menemukan maxima over scale. Menolak poin yang Laplacian tidak mencapai ekstrem atau responsnya di bawah ambang batas.

**Extended Harris-Laplace.** Untuk beberapa titik, maksimum jejak skala tidak sesuai dengan skala deteksi yang diatur secara arbitrer. Poin-poin ini ditolak karena kurangnya maksimum, atau lokasi dan skalanya tidak akurat. Algoritma Harris-Laplace dapat diperluas untuk mencari lokasi  $x$  dan skala  $\sigma_I$  dari suatu interest point dengan akurasi tinggi. Detektor dapat diinisialisasi dengan titik Harris multi-skala. Selanjutnya, untuk setiap titik dapat menerapkan algoritma iteratif yang secara bersamaan mendeteksi lokasi dan skala titik-titik tersebut. Sebuah metode iteratif langsung untuk deteksi fitur dapat dinyatakan sebagai berikut. Untuk titik awal tertentu  $x$  dengan skala  $\sigma_I$ :

1. temukan skala ekstrim lokal untuk titik  $x^{(k)}$ , jika tidak tolak titik tersebut. Kisaran skala dapat dibatasi oleh  $\sigma_I^{(k|1)} = s\sigma_I^{(k)}$  dengan  $s \in [0.7, \cdot, 1.4]$ .
2. mendeteksi spasial lokasi  $x^{k|1}$  dari maksimum ukuran Harris yang terdekat  $x^k$  untuk dipilih  $\sigma_I^{k|1}$ .



**Gambar 2.4**

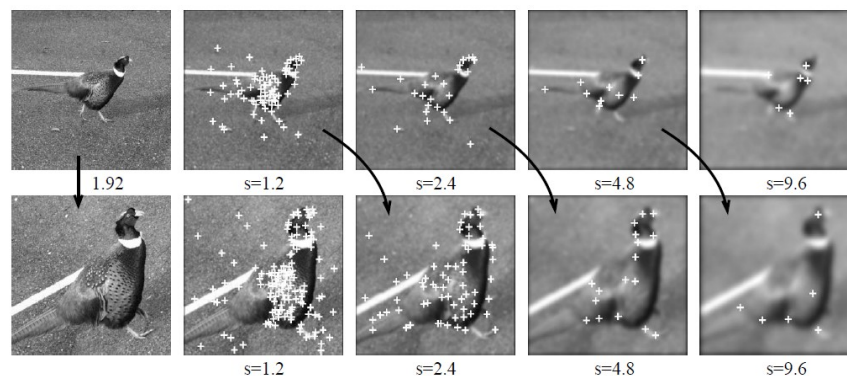
3. lanjut ke langkah 1 jika  $\sigma_I^{k|1} \neq \sigma_I^{(k)}$  atau  $x^{(k|1)} \neq x^{(k)}$ .

Titik awal dapat dideteksi dengan perubahan skala yang lebih besar antara dua tingkat representasi berurutan, yaitu  $s = 1.4$ . Interval skala yang lebih kecil yaitu  $s = 1.12$ , dalam algoritma iteratif memberikan perkiraan lokasi dan skala yang lebih baik. Seperti yang dapat dibayangkan, titik awal yang terdeteksi pada struktur lokal yang sama tetapi pada level representasi yang berbeda harus menyatu ke lokasi yang sama dan skala yang sama. Sangat mudah untuk menemukan titik yang mirip menggunakan koordinat titik dan skala. Untuk mewakili struktur, hanya dapat menyimpan salah satunya. Pendekatan ini memberikan titik-titik, yang lokasi dan skalanya diperkirakan dengan akurasi tinggi. Itu juga menemukan parameter yang benar untuk poin, yang ditolak oleh ukuran Laplacian dalam pendekatan Harris-Laplace. Namun, algoritma iteratif yang diterapkan untuk setiap titik awal memakan waktu lebih lama dibandingkan dengan pendekatan Harris-Laplace.

### 2.3.2 Scale covariant points

Pada gambar di atas kami menyajikan beberapa contoh titik yang terdeteksi dengan metode Harris-Laplace. Baris atas menunjukkan titik yang terdeteksi dengan detektor Harris multiskala. Skala deteksi diwakili oleh lingkaran di sekitar titik dengan radius  $3\sigma_I$ . Perhatikan, bagaimana titik minat, yang terdeteksi untuk struktur gambar yang sama, mengubah lokasinya dalam arah gradien relatif terhadap skala deteksi. Seseorang dapat menentukan rantai poin dan memilih hanya satu dari mereka untuk mewakili struktur lokal. Titik serupa terletak di lingkungan kecil dan dapat ditentukan dengan membandingkan deskriptornya. Namun, untuk struktur lokal yang ada dalam berbagai skala, konten informasi dapat berubah. Dalam pendekatan, ukuran LoG digunakan untuk memilih titik representatif untuk struktur tersebut. Selain itu, Laplacian memungkinkan titik-titik karakteristik yang sesuai untuk dipilih (baris bawah) bahkan jika transformasi antar citra signifikan. Kadang-kadang, dua atau lebih titik dipilih, tetapi tanpa pengetahuan sebelumnya tentang perubahan skala antara gambar, harus menyimpan semua titik yang dipilih. Lokasi dan skala titik-titik benar sehubungan dengan transformasi antar gambar.





**Gambar 2.5**

Gambar di atas menunjukkan representasi skala-ruang untuk dua citra dengan titik-titik yang dideteksi dengan metode Harris-Laplace. Untuk setiap level skala objek, kami menyajikan titik-titik yang dipilih. Ada banyak korespondensi point-to-point antara level yang rasio skalanya sesuai dengan perubahan skala nyata antara gambar (ditunjukkan dengan pointer). Selain itu, sangat sedikit titik yang terdeteksi di lokasi yang sama tetapi pada level yang berbeda. Oleh karena itu, titik-titik merupakan karakteristik pada bidang citra dan dalam dimensi skala.

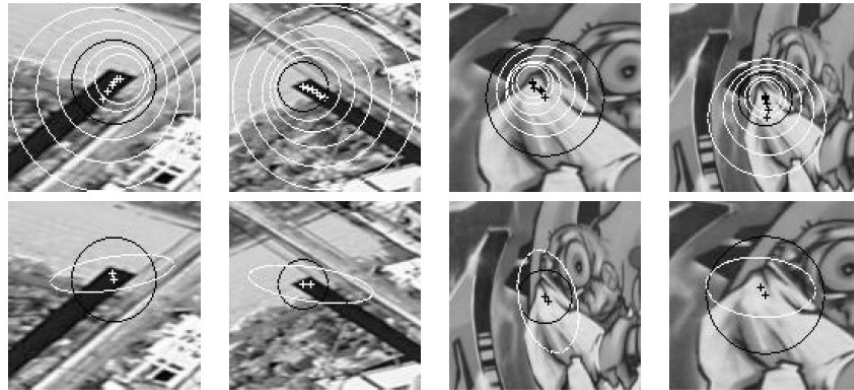
## 2.4 Affine invariant detector

Sebuah detektor invarian affine dapat dilihat sebagai generalisasi dari detektor invarian skala. Dalam kasus transformasi affine penskalaan bisa tidak seragam, yang berbeda di setiap arah. Penskalaan yang tidak seragam berpengaruh terhadap lokalisasi, skala dan bentuk struktur lokal yang khas. Oleh karena itu, detektor skala invarian gagal dalam kasus transformasi affine yang signifikan.

### 2.4.1 Harris-Affine detector

Dalam kasus transformasi afin, perubahan skala dapat berbeda di setiap arah. Detektor Harris-Laplace yang disajikan akan gagal dalam kasus transformasi affine yang penting karena mengasumsikan perubahan skala yang seragam. Angka ?? menyajikan dua pasang titik yang terdeteksi dalam gambar dengan deformasi afin yang signifikan. Baris atas menunjukkan titik yang terdeteksi dengan detektor Harris multiskala. Skala yang dipilih dengan Laplacian ditampilkan dalam warna hitam. Jika diproyeksikan lingkungan lingkaran dari titik yang bersesuaian menggunakan transformasi affine, diperoleh daerah elips yang tidak menutupi bagian yang sama dari gambar. Dapat dilihat titik yang diproyeksikan ditampilkan di baris bawah (berwarna putih) ditumpangkan pada titik Harris-Laplace yang sesuai (berwarna hitam)

Dalam kasus transformasi halus, ketika perubahan skala tidak selalu sama di



**Gambar 2.6**

setiap arah, skala yang dipilih secara otomatis tidak mencerminkan transformasi sebenarnya dari suatu titik. Diketahui bahwa Harris maxima lokal mengubah lokasi spasial sehubungan dengan skala deteksi. Dengan demikian, kesalahan tambahan terjadi pada lokasi titik jika skala deteksi tidak sesuai dengan faktor skala antara pola gambar yang sesuai. Skala deteksi dalam arah ortogonal harus bervariasi secara independen, untuk menangani kemungkinan penskalaan affine. Misalkan kedua skala dapat disesuaikan dengan struktur gambar lokal. Oleh karena itu, kita menghadapi masalah komputasi matriks momen kedua dalam ruang skala affine Gaussian, di mana lingkungan titik lingkaran digantikan oleh elips.

Banyak hasil yang sukses dalam memperkirakan deformasi afin dengan matriks momen kedua membuktikan kegunaan matriks ini. Telah mengeksplorasi propertinya untuk memilih skala deteksi. Properti yang memadai dijelaskan di bagian 2.1.3. Untuk titik tertentu  $x$  matriks momen kedua  $\mu$  dalam ruang skala tidak seragam didefinisikan oleh:

$$\mu(x, \Sigma_I, \Sigma_D) = g(\Sigma_I) * ((\nabla L)(x, \Sigma_D)(\nabla L)(x, \Sigma_D)^T)$$

di mana  $\Sigma_I$  dan  $\Sigma_D$  adalah matriks kovarians yang menentukan integrasi dan diferensiasi kernel Gaussian. Untuk mengurangi ruang pencarian, diberlakukan kondisi  $\Sigma_I = s\Sigma_D$ , dimana  $s$  adalah skalar. Selanjutnya, untuk membatasi ruang pencarian inisialisasi detektor affine dengan interest point diekstrak oleh detektor Harris multi-skala. Detektor apa pun dapat digunakan untuk menentukan lokalisasi spasial dari titik awal, tetapi detektor Harris juga didasarkan pada *Second moment matrix*, agar cocok secara alami dalam kerangka kerja ini. Untuk mendapatkan matriks bentuk untuk setiap titik kepentingan menghitung deskriptor momen kedua dengan skala integrasi dan diferensiasi yang dipilih secara otomatis. Garis besar metode deteksi kami disajikan dalam berikut

1. lokalisasi spasial dari suatu titik perhatian pada skala dan bentuk tertentu ditentukan oleh maksimum lokal dari fungsi Harris,

2. skala integrasi dipilih pada skala ekstrim dari turunan yang dinormalisasi,
3. skala diferensiasi dipilih pada maksimum isotropi yang dinormalisasi,
4. matriks adaptasi bentuk menormalkan lingkungan titik.

Berikut ini dibahas secara rinci setiap langkah dari algoritma.

**Shape adaptation matrix.** metode adaptasi bentuk iteratif bekerja pada domain citra yang ditransformasikan. Seperti yang disajikan di bagian 2.1, alih-alih menerapkan kernel affine Gaussian, kami mengubah gambar dan menerapkan kernel yang seragam. Itu memungkinkan penggunaan implementasi rekursif filter Gaussian seragam untuk menghitung  $L_x$  dan  $L_y$ . Matriks momen kedua dihitung menurut persamaan ???. Jendela lokal  $W$  berpusat pada titik minat  $x$  dan ditransformasikan oleh matriks:

$$U^{k-1} = (\mu^{-\frac{1}{2}})^{k-1} \cdot (\mu^{-\frac{1}{2}})^1 \dots U^0 \quad (2.23)$$

dalam langkah ( $k$ ) dari algoritma iteratif. Berikut ini kami menyebut operasi ini sebagai transformasi-U. Perhatikan, bahwa matriks  $\mu$  baru dihitung pada setiap iterasi dan matriks  $U$  adalah gabungan dari akar kuadrat dari matriks momen kedua. Kami memastikan bahwa gambar asli diambil sampelnya dengan benar dengan menyetel nilai eigen yang lebih besar  $\lambda_{max}(U) = 1$ . Ini berarti patch gambar diperbesar ke arah  $\lambda_{min}(U)$ . Untuk titik tertentu integrasi dan skala lokal menentukan matriks momen kedua  $\mu$ . Parameter skala ini secara otomatis terdeteksi di setiap langkah iterasi. Dengan demikian, matriks  $\mu$  yang dihasilkan tidak bergantung pada skala awal dan resolusi gambar.

**Integration scale** Untuk titik spasial tertentu, kami secara otomatis memilih skala karakteristiknya. Untuk mempertahankan invarian terhadap perubahan ukuran, kami memilih skala integrasi  $\sigma_I$  di mana Laplacian yang dinormalisasi mencapai skala maksimum lokal. Dalam kasus perubahan skala yang lemah, cukup menjaga  $\sigma_I$  konstan selama iterasi. Di hadapan deformasi afin penting perubahan skala sangat berbeda di setiap arah. Dengan demikian, skala karakteristik yang terdeteksi pada gambar asli dan versi transformasi U-nya dapat berbeda secara signifikan. Oleh karena itu, sangat penting untuk memilih skala integrasi setelah menerapkan transformasi U. Kami menggunakan prosedur yang mirip dengan yang dijelaskan untuk versi perluasan detektor Harris-Laplace, di bagian 2.3.1. Hal ini memungkinkan titik-titik awal menyatu menuju titik di mana skala dan matriks momen kedua tidak berubah lagi. Perhatikan, bahwa skala ekstrem harus memiliki jenis yang sama selama iterasi. Jika tidak, metode dapat beralih antara maksimum dan minimum jika terdapat kedua jenis ekstrem dalam rentang skala yang dipindai.

**Differentiation scale** Skala diferensiasi lokal kurang kritis dan dapat diatur secara proporsional dengan skala integrasi  $\sigma_D = s\sigma_I$ , dimana  $s$  adalah faktor konstan. Namun, kami mengusulkan untuk mendasarkan skala turunan pada ukuran isotropi yang diperkenalkan di bagian 2.1.3. Faktor  $s$  umumnya dipilih dari rentang  $[0.5, \cdot, 0.7]$ . Solusinya adalah memilih skala diferensiasi yang isotropi lokalnya

diasumsikan maksimum pada rentang skala ini. Mengingat skala integrasi  $\sigma_I$  kami memilih  $s \in [0.5, \cdot, 0.7]$  dimana ukuran  $Q$  mengasumsikan maksimum. Solusi ini dimotivasi oleh fakta bahwa skala lokal memiliki pengaruh penting pada konvergensi matriks momen kedua. Prosedur iteratif konvergen menuju matriks dengan nilai eigen yang sama. Semakin kecil perbedaan antara nilai eigen  $\lambda_{max}(\mu)$ ,  $\lambda_{min}(\mu)$  dari matriks awal, semakin dekat solusi akhir dan prosedur semakin cepat konvergen. Perhatikan bahwa ukuran Harris sudah memilih titik dengan dua nilai eigen besar. Perbedaan besar antara nilai eigen menyebabkan penskalaan besar dalam satu arah oleh transformasi  $U$ . Intinya tidak konvergen ke solusi yang stabil karena kebisingan. Pemilihan skala lokal memungkinkan diperoleh rasio nilai eigen yang masuk akal dan titik-titik konvergen, yang tidak akan konvergen jika rasionya terlalu besar.

**Spatial localization** telah menunjukkan bagaimana maxima lokal dari ukuran Harris ubah lokasinya jika skala deteksi berubah. Bisa juga mengamati efek ini, ketika perubahan skala berbeda di setiap arah. Deteksi dengan skala yang berbeda dalam arah  $x$  dan  $y$  diganti dengan mengadaptasi gambar dan kemudian menerapkan skala yang sama di kedua arah. Normalisasi affine dari suatu lingkungan titik sedikit menggeser maksima spasial dari fungsi Harris. Akibatnya, kami mendeteksi ulang maksimum di jendela normalisasi affine  $W$ . Dengan demikian, kami memperoleh vektor perpindahan ke maksimum terdekat dalam domain gambar yang dinormalisasi- $U$ . Lokasi titik awal dikoreksi dengan vektor perpindahan yang ditransformasikan kembali ke domain gambar asli:

$$x^k = x^{k-1} + U^{k-1} \dots (x_w^k - x_{w-1}^{k-1})$$

dimana  $x_w$  adalah titik di koordinat gambar yang diubah- $U$ .

**Convergence criterion.** Bagian penting dari prosedur iterasi adalah kriteria penghentian. Ukuran konvergensi dapat didasarkan pada matriks  $U$  atau  $\mu$ . Jika kriteria didasarkan pada  $\mu$  yang dihitung dalam setiap langkah iterasi, kami mensyaratkan matriks ini cukup dekat dengan rotasi murni. Ini menyiratkan bahwa  $\lambda_{max}(\mu)$  dan  $\lambda_{min}(\mu)$  adalah sama. Dalam praktiknya kami mengizinkan kesalahan kecil  $e_C = 0.05$ .

$$\frac{\lambda_{max}(\mu) - \lambda_{min}(\mu)}{\lambda_{max}(\mu)} < \epsilon_C \quad (2.24)$$

Kemungkinan lain adalah menguraikan matriks  $U = R^T.D.R$  menjadi rotasi  $R$  dan penskalaan  $D$  dan membandingkan transformasi yang berurutan. Kami mengizinkan titik jika transformasi  $R$  dan  $D$  berturut-turut cukup mirip. Kedua kriteria terminasi tersebut memberikan hasil akhir yang sama. Poin penting lainnya adalah menghentikan iterasi jika terjadi divergensi. Dalam teori ada kasus tunggal ketika rasio nilai eigen cenderung tak terhingga. Oleh karena itu, titik tersebut harus ditolak jika rasionya terlalu besar (yaitu  $e_I = 6$ ), jika tidak maka akan menyebabkan struktur memanjang yang tidak stabil.

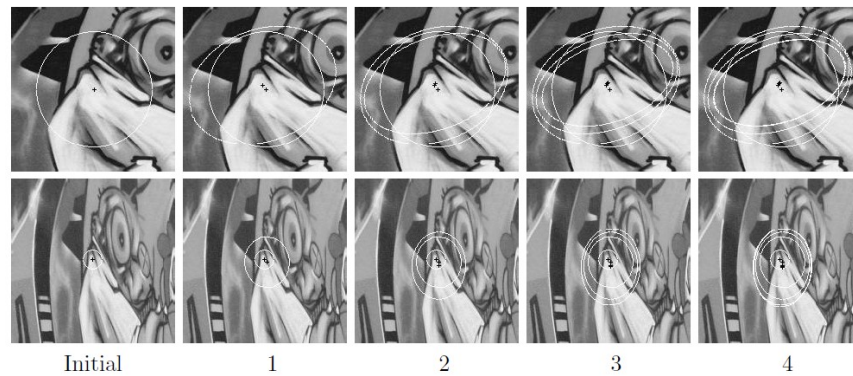
$$\frac{\lambda_{max}(D)}{\lambda_{min}(D)} > \epsilon_\iota \quad (2.25)$$

Properti konvergensi dari algoritma adaptasi bentuk dipelajari secara ekstensif. Terlihat bahwa selain kasus tunggal, titik konvergensi selalu unik. Secara umum prosedur konvergen asalkan estimasi awal deformasi affine cukup dekat dengan deformasi yang sebenarnya dan skala integrasi dipilih dengan benar sehubungan dengan ukuran struktur gambar lokal.

**Detection algorithm** mengusulkan prosedur iteratif yang memungkinkan titik-titik awal untuk bertemu dengan titik-titik kovarian affine, yaitu titik-titik yang secara kovarian berubah dengan sudut pandang. Untuk menginisialisasi algoritme kami, kami menggunakan poin yang diekstraksi oleh detektor Harris multi-skala. Titik-titik ini tidak terdeteksi dengan cara invarian affine karena kernel Gaussian yang tidak diadaptasi, tetapi memberikan perkiraan lokalisasi dan skala untuk pencarian lebih lanjut untuk titik minat kovarian affine. Untuk titik minat awal tertentu  $x^{(0)}$  kami menerapkan prosedur berikut:

1. menginisialisasi  $U^{(0)}$  ke matriks identitas,
2. menormalkan jendela  $W(x_w) = I(x)$  berpusat di  $U^{(k-1)}x_w^{(k-1)} = x^{(k-1)}$ ,
3. pilih skala integrasi  $\sigma_I$  dalam  $x_w^{(k-1)}$ ,
4. pilih skala diferensiasi  $\sigma_D = s\sigma_I$ , yang memaksimalkan  $\frac{\lambda_{min}(\mu)}{\lambda_{max}(\mu)}$  dengan  $s \in [0.5, \cdot, 0.7]$  dan  $\mu = \mu(x_w^{(k-1)})$ ,  $\sigma_D, \sigma_I$
5. mendeteksi lokalisasi spasial  $x_w^{(k)}$  dari maksimum ukuran Harris terdekat dengan  $x_w^{(k-1)}$  dan hitung lokasi titik minat  $x_w^{(k-1)}$
6. menghitung  $\mu_i^{(k)} = \mu^{-\frac{1}{2}}(x_w^{(k)}, \sigma_D, \sigma_I)$
7. transformasi gabungan  $U^{(k)} = \mu^{-\frac{1}{2}}U^{(k-1)}$  dan normalisasi  $U^{(k)}$  ke  $\lambda_{max}(U^{(k)}) = 1$
8. lanjut ke langkah 2 jika  $1 - \frac{\lambda_{min}(\mu_i^{(k)})}{\lambda_{max}(\mu_i^{(k)})} > e_C$

Meskipun perhitungan mungkin tampak sangat memakan waktu, perhatikan bahwa sebagian besar waktu dihabiskan untuk menghitung  $L_x$  dan  $L_y$ , yang dilakukan hanya sekali dalam setiap langkah jika hubungan antara integrasi dan skala lokal konstan. Perulangan iterasi dimulai dengan memilih skala integrasi karena kami menyadari bahwa bagian algoritme ini paling kuat untuk kesalahan pelokalan kecil dari suatu titik minat. Namun, skala  $\sigma_I$  berubah jika bentuk tambalan diubah. Diberikan solusi perkiraan awal, algoritme yang disajikan memungkinkan seseorang untuk secara iteratif memodifikasi bentuk, skala, dan lokasi spasial suatu titik dan menyatu dengan struktur lokal, yang ditentukan meskipun transformasi afin



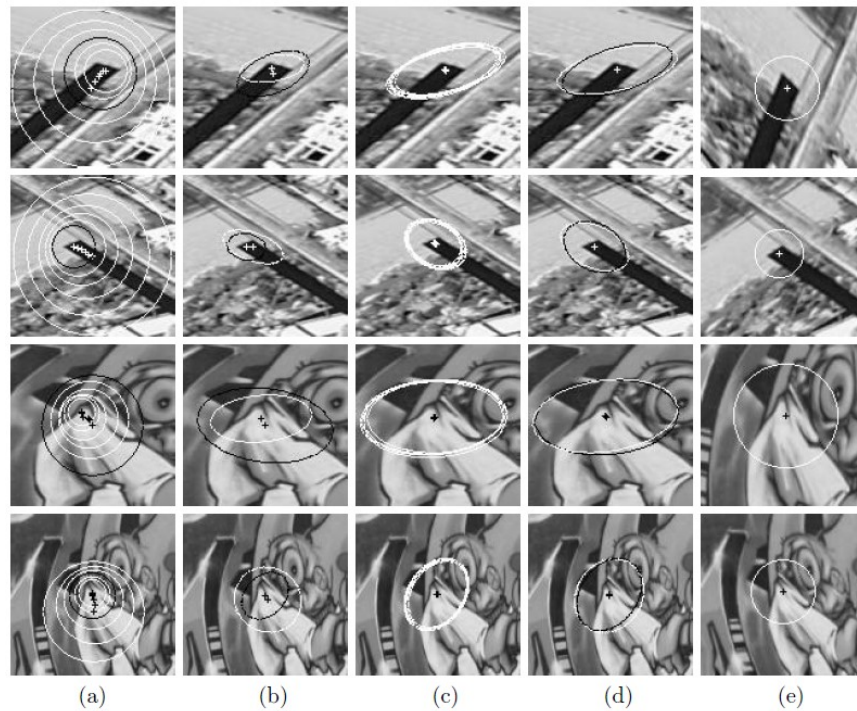
**Gambar 2.7**

berubah-ubah. Gambar 10 menunjukkan titik-titik afin yang terdeteksi dalam langkah-langkah berurutan dari prosedur berulang. Setelah iterasi keempat lokasi, skala dan bentuk titik tidak berubah lagi. Kita dapat melihat bahwa elips menutupi wilayah gambar yang sama meskipun terjadi deformasi affine yang kuat.

**Selection of similar affine points** Asalkan wilayah yang dinormalisasi adalah isotropik, ada satu maksimum spasial dari ukuran Harris dan satu skala karakteristik untuk struktur lokal yang dipertimbangkan. Oleh karena itu, beberapa titik awal yang berkorespondensi dengan fitur yang sama tetapi terdeteksi pada level skala yang berbeda dapat menyatu menuju satu lokasi dan skala titik. Sangat mudah untuk mengidentifikasi titik-titik ini dengan membandingkan lokasinya  $(x, y)$ , skala  $\sigma_I$ , regangan  $\lambda_{min}(U)$  dan miring. Kemiringan dipulihkan dari matriks rotasi  $R$ , di mana  $U = R^T \cdot D \cdot R$ . Kami mendefinisikan titik serupa jika masing-masing parameter ini cukup dekat dengan parameter titik referensi. Terakhir, kami menghitung parameter rata-rata dan memilih titik yang paling mirip dari kumpulan titik yang diidentifikasi. Sebagai hasilnya, untuk citra tertentu kita memperoleh sekumpulan titik, di mana masing-masing mewakili lokasi dan struktur citra yang berbeda.

#### 2.4.2 Affine covariant points

Gambar di atas menyajikan dua contoh karakteristik struktur lokal. Kolom (a) ditampilkan titik yang digunakan untuk inisialisasi, yang dideteksi oleh detektor Harris multi-skala. Lingkaran di sekitar titik menunjukkan skala pendeteksian, di mana jari-jari lingkaran adalah  $3\sigma_I$ . Lingkaran berwarna hitam menunjukkan titik yang dipilih oleh detektor Harris-Laplace. Perhatikan bahwa ada perpindahan penting antara titik yang terdeteksi pada skala yang berbeda dan lingkaran pada gambar yang sesuai (baris atas dan bawah) tidak menutupi bagian gambar yang sama. Pada kolom (b) kami menunjukkan titik-titik (berwarna hitam) yang terdeteksi dengan menerapkan prosedur iteratif pada titik-titik Harris-Laplace. Skala dan lokasi titik konstan selama iterasi. Wilayah terkait yang diproyeksikan ditampilkan dalam warna putih dan dengan jelas menunjukkan perbedaan dalam lokalisasi dan bentuk wilayah. Skala awal tidak terdeteksi dengan benar karena operator Laplacian yang



**Gambar 2.8**

tidak diadaptasi secara seragam. Demikian pula, lokasi titik berbeda dalam 3-4 piksel. Dalam pendekatan kami, titik-titik yang sesuai dengan struktur fisik yang sama, tetapi terdeteksi di lokasi yang berbeda karena skala, menyatu ke lokasi titik yang sama. Oleh karena itu, jumlah poin bunga efektif berkurang. Titik-titik kovarian affine, tempat titik-titik awal bertemu disajikan dalam kolom (c). Poin-poin ini diperoleh dengan menerapkan algoritme yang dijelaskan di bagian sebelumnya. Kita dapat melihat bahwa metode konvergen dengan benar meskipun lokasi dan skalanya titik awal relatif jauh dari titik konvergensi. Konvergensi pada umumnya diperoleh dalam waktu kurang dari 10 iterasi. Perbedaan kecil antara daerah di kolom (d) disebabkan oleh ketidaktepatan estimasi skala dan kesalahan  $e_C$ . Kolom (e) menunjukkan titik "rata-rata" yang dinormalisasi dengan perkiraan matriks untuk menghilangkan peregangan dan kemiringan. Kita dapat melihat dengan jelas bahwa wilayah berkorespondensi antara dua gambar (baris atas dan bawah).

## **BAB III**

### **METODE PENELITIAN**

#### **3.1 Deskripsi Sistem**

Sistem bertujuan untuk melakukan penghitungan panjang serta berat objek ikan yang menggunakan metode *Harris-corners* dan *Gaussian Laplace*. Penelitian berfokus dalam menghasilkan berupa data hasil panjang dan berat objek ikan. Dataset citra ikan diambil secara langsung menggunakan *Smartphone* pada sebuah penangkaran ikan. Kemudian bahasa yang akan digunakan dalam membangun sistem adalah Python versi 3.

Tahapan dalam penghitungan panjang dan berat ikan menggunakan *Harris-Corner Detection* adalah Pemrosesan Citra, *Smoothing Gaussian*, Perhitungan gradien, autokorelasi matriks gradien, *Harris Respons*, melakukan threshold-ing dan *Non-Maximum Suppression*, menentukan koordinat sudut, perhitungan panjang dan lebar, Penghitungan berat, dan terakhir adalah mengasosiasikan data.

#### **3.2 Perancangan Sistem**

Pada bagian ini akan membahas tentang proses yang akan dilakukan untuk mengetahui tahapan yang ditempuh dalam membangun sistem penghitungan berat dan panjang ikan. Tahapan pertama yang dilakukan adalah melakukan *input* gambar yang akan dibaca sistem, setelah itu citra akan diproses menggunakan *Smoothing gaussian* untuk mengurangi *noise*. Kemudian perhitungan gradien dilakukan bertujuan untuk menghitung setiap perubahan intensitas yang berada di sekitar titik, hasil tersebut akan diolah kembali menggunakan autokorelasi matriks gradien, dan dilanjutkan dengan menjalankan *Harris respons*. *Harris respons* bertujuan untuk mengetahui bahwa titik tersebut adalah sebuah sudut, sisi, atau tepi berdasarkan pada hasil respon harris. Setelah mendapatkan titik yang merupakan sebuah sisi, titik-titik tersebut akan dipilih. Dimana hasil dari pemilihan tersebut akan menjadi cara untuk dapat menghitung panjang, lebar dan berat ikan.

##### **3.2.1 Pemrosesan Citra**

Gambar yang akan digunakan berformat .jpg atau .png, dengan latar belakang yang telah dihilangkan dan digantikan dengan warna solid. Gambar tersebut akan diubah dari format RGB menjadi citra *grayscale*. Perubahan ini diperlukan untuk mempermudah proses pendeteksian sudut.



### 3.2.2 Smoothing Gaussian

Metode *Smoothing Gaussian* digunakan untuk mengurangi *noise* pada gambar sebelum melanjutkan ke perhitungan lebih lanjut. *Noise* adalah gangguan yang muncul dalam gambar digital yang dapat mengaburkan detail atau informasi penting. *noise* dapat disebabkan oleh berbagai faktor, seperti kualitas sensor kamera yang rendah, pencahayaan yang buruk, atau kesalahan dalam proses pemindahan data. Dalam gambar, *noise* biasanya terlihat sebagai bintik-bintik acak atau intensitas piksel yang tidak diinginkan. Keberadaan *noise* ini dapat menjadi masalah dalam proses pendeteksian sudut, karena *noise* dapat menyebabkan munculnya sudut palsu yang tidak sesuai dengan gambar asli.

Langkah pertama dalam *smoothing Gaussian* adalah membentuk *Gaussian kernel*. Pemilihan ukuran kernel sangat berpengaruh terhadap hasil *smoothing*, khususnya pada tingkat kehalusan gambar dan jumlah *noise* yang berkurang. Semakin besar kernel yang digunakan, semakin banyak *noise* yang dihilangkan, tetapi hal ini dapat menyebabkan detail gambar ikut terhapus. Sebaliknya, jika kernel yang digunakan berukuran kecil, detail gambar akan lebih terjaga, tetapi *noise* mungkin masih tetap ada. Setelah menentukan ukuran kernel yang akan digunakan kernel dapat dihitung dengan menggunakan persamaan (2.1).

Langkah selanjutnya adalah mengaplikasikan kernel pada gambar. Proses ini dilakukan dengan menerapkan kernel secara berulang pada setiap piksel di seluruh gambar. Namun, sebelum langkah tersebut, gambar perlu di-*padding* terlebih dahulu. *Padding* adalah proses menambahkan piksel pada tepi gambar sebelum pemrosesan dengan kernel atau model dimulai. Tujuan dari *padding* adalah untuk mempertahankan ukuran gambar setelah diterapkannya operasi tertentu, seperti konvolusi. Setelah *padding* diterapkan, konvolusi akan dilakukan dengan menggunakan persamaan (2.2).

### 3.2.3 Perhitungan Gradien

Gambar yang telah diperhalus akan dilanjutkan dengan menghitung gradient pada setiap *pixel*-nya. Tujuan dari penghitungan gradient adalah untuk mengetahui perubahan intensitas pada setiap piksel, yang pada umumnya akan menunjukan tepi atau kontur dalam sebuah gambar. Penghitungan gradient juga memerlukan kernel untuk dapat bekerja, terdapat banyak kernel yang dapat digunakan untuk menghitung gradient, contohnya kernel sobel.

Kernel Sobel digunakan untuk menghitung estimasi gradient dengan memberikan bobot yang lebih besar pada piksel tetangga terdekat di sekitar piksel pusat. Kernel sobel terdiri dari dua matriks  $3 \times 3$ , yang bertujuan untuk menghitung gradient pada dua arah yaitu:

- **Gradien Horizontal (Gx):** Mengukur perubahan intensitas sepanjang sumbu x.
- **Gradien Vertical (Gy):** Mengukur perubahan intensitas sepanjang sumbu y.

Berikut adalah kernel yang akan digunakan dalam operator sobel :

1. Kernel Sobel Horizontal

$$Sx = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

2. Kernel Sobel Vertical

$$Sy = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

Perhitungan gradient juga akan menggunakan rumus (2.2) tetapi gaussian filter akan diganti dengan kernel sobel horizontal (Gx) dan kernel sobel vertical (Gy). Maka rumusannya akan seperti berikut :

$$\begin{aligned} Ix &= Sx * I(x) \\ Iy &= Sy * I(x) \end{aligned} \tag{3.1}$$

### 3.2.4 Autokorelasi

Setelah hasil dari perhitungan gradien didapatkan, hasil tersebut akan dibuat menjadi matriks kembali dengan rumus (2.3). Nilai gradien dihitung untuk mendapatkan koefisien matriks autokorelasi. Setelah koefisien didapatkan, koefisien tersebut akan dikonvolusi dengan filter gaussian. Baru setelah itu hasil dari konvolusi digunakan untuk membentuk matriks  $M$  yang akan digunakan untuk perhitungan selanjutnya. Bentuk dari matriks  $M$  sebagai berikut :

$$\begin{aligned}
 \Sigma Ix &= g(\sigma) * Ix^2 \\
 \Sigma IxIy &= g(\sigma) * Ix * Iy \\
 \Sigma Iy &= g(\sigma) * Iy^2 \\
 M &= \begin{bmatrix} \Sigma Ix & \Sigma IxIy \\ \Sigma IxIy & \Sigma Iy \end{bmatrix}
 \end{aligned} \tag{3.2}$$

### 3.2.5 Harris Respon

Matriks  $M$  digunakan untuk menghitung respon harris, menggunakan rumus harris yaitu:

$$R(x, y) = Det(M) - k * (trace(M))^2 \tag{3.3}$$

Dimana  $Det(M)$  :

$$Det(M) = \Sigma Ix * \Sigma Iy - \Sigma IxIy^2$$

Dan  $trace(M)$

$$trace(M) = \Sigma Ix + \Sigma Iy$$

### 3.2.6 Thresholding dan *Non-Maximum suppression*

Setelah nilai harris didapatkan tahapan langkah selanjutnya adalah melakukan Thresholding. Thresholding adalah cara untuk memilih hasil deteksi dengan cara menetapkan nilai ambang batas. Jika nilai harris melebihi dari ambang batas, maka piksel tersebut akan dianggap sebagai potensial sudut, Sebaliknya jika nilai harris lebih kecil maka piksel tersebut akan diabaikan. Tujuannya supaya mengurangi piksel yang perlu menjadi diproses lebih lanjut.

$$Corner(x, y) = \begin{cases} 1 & \text{jika } R(x, y) > T \\ 0 & \text{jika } R(x, y) \leq T \end{cases} \quad (3.4)$$

Setelah melewati Thresholding, setiap piksel yang telah lolos akan diperiksa kembali dengan *Non-Maximum Suppression* atau NMS. NMS bertujuan untuk mengelolah kembali piksel yang saling berdekatan satu sama lain. NMS akan memilih piksel pada *range* tertentu, piksel dengan nilai  $R$  terbesar akan terpilih sebagai puncak lokan dan dipertahankan sebagai sudut, dan yang lainnya akan diabaikan.

$$NMS(x, y) = \begin{cases} 1 & \text{jika } R(x, y) = MaxLocalR(i, j) \\ 0 & \text{lainnya} \end{cases}$$

Dimana  $MaxLocal R(i, j)$  adalah nilai maksimum yang berada disekitar piksel  $(x, y)$ .

### 3.2.7 Memilih Piksel

Setelah sudut telah melalui Thresholding dan NMS, sudut-sudut tersebut akan dipilih berdasarkan kriteria yang telah ditentukan.

1. Sudut berada pada bagian ekor.
2. Sudut berada pada bagian mulut.
3. Sudut berada pada bagian sirip atas.
4. Sudut berada pada bagian bawah atau perut.

Keempat kriteria tersebut dibutuhkan untuk mempermudah dalam penghitungan panjang dan lebar.

### 3.2.8 Menghitung Panjang dan Berat

Untuk menghitung panjang dan lebar asli sebuah benda dalam gambar dibutuhkan dua hal, yaitu skala dan panjang benda pada gambar. Skala dapat dicari dengan dua cara yaitu menggunakan benda referensi, atau dengan menggunakan resolusi pada gambar tersebut.

Kedua cara tersebut memiliki kelebihan dan kekurangan masing-masing, contohnya menggunakan benda referensi. Metode ini memerlukan benda lain dalam gambar dan mengetahui panjang dari benda lain tersebut, jika tidak ada benda lain pada gambar diperlukan pengambilan gambar pada jarak yang sama dengan benda yang ingin dicari panjangnya. Kelebihan dari metode ini adalah akurasi yang tinggi, metode selanjutnya juga sama menggunakan resolusi gambar dapat mempermudah penghitungan, namun akurasi dari penghitungan panjang akan berkurang.

Setelah diketahui panjang dan lebar, langkah berikutnya adalah menghitung berat. Penghitungan berat pada gambar akan dilakukan dengan menggunakan interpolasi dengan data yang telah diambil. Menggunakan rumus sebagai berikut:

$$W_n = W_x + (L_n - L_x / L_y - L_x) * (W_y - W_x) \quad (3.5)$$

Dimana

- $W_n$  berat yang dicari.
- $W_x$  berat sebelum.
- $W_y$  berat sesudah.
- $L_n$  panjang dari berat yang dicari.
- $L_x$  panjang dari berat sebelum.
- $L_y$  panjang dari berat sesudah.

Jika diatas akan dilakukan kembali jika lebar dari ikan juga berbeda.