

# CAPÍTULO 4

## INTERAÇÃO DE ROBÔ NO AMBIENTE

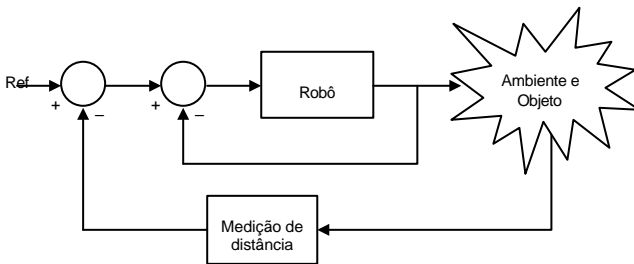
Anna Helena Reali Costa  
Jun Okamoto Jr.

### 4.1 - INTRODUÇÃO

A simples menção da palavra “automação” ou da palavra “automático” traz à mente a noção de que a máquina deve sentir o ambiente que a cerca para que ela possa executar a sua tarefa de modo automático. Em outras palavras, a máquina, equipamento ou processo deve utilizar sensores para fornecer informações para o seu sistema de controle interno de modo a executar sua tarefa sem o auxílio de um ser humano.

De uma maneira genérica estes sensores devem fornecer informações ao computador de controle sobre distâncias entre máquina e objeto, posição da própria máquina, contato entre máquina e objeto, força que a máquina exerce sobre objeto, ou ainda cor do objeto, textura do objeto (DU BUF, KARDAN, SAPNN, 1990), cheiro do objeto (WILKINSON, GUANIO, 1998), dentre outras.

Em robótica, quando se fala em sensores, normalmente refere-se a sensores que auxiliam o robô na execução de suas tarefas, isto é, que são utilizados para o robô determinar parâmetros do ambiente que o cerca e do objeto que se deseja manipular. É conveniente se distinguir entre estes sensores e os que são utilizados para o robô determinar seus parâmetros de movimento, ou seja, sua posição, velocidade, aceleração e força. Os sensores do primeiro caso são chamados de sensores externos do robô e os do segundo caso são chamados de sensores internos do robô.



**Figura 4.1 - Controle de robô com duas malhas de realimentação**

É claro que o parâmetro que está sendo medido determina o tipo de sensor a ser utilizado, porém, a principal diferença entre um sensor interno e um externo está na velocidade de resposta que o sensor deve ter. A Figura 4.1 mostra um diagrama de blocos do sistema de controle de um robô onde estão presentes duas malhas de realimentação de informações sensoriais, uma interna e outra externa.

A malha de realimentação interna requer o controle em tempo real da posição do robô que opera em períodos de amostragem da ordem de milissegundos, ou seja, o sensor que está sendo

utilizado para medir a posição do robô deve ser consultado na frequência do período de amostragem e deve produzir um valor válido num tempo menor que o do período de amostragem.

Por outro lado, a malha de realimentação externa utiliza informações do ambiente adquiridas pelo sensor para a execução da tarefa num nível mais alto. Por exemplo, no caso da Figura 4.1, com uma realimentação de medida de distância até o objeto a tarefa de alto nível pode ser manter constante a distância até o objeto, como no caso de acompanhar um objeto numa esteira transportadora. Enquanto se faz o planejamento da tarefa sendo executada na malha de realimentação externa, a malha de realimentação interna mantém o robô na trajetória planejada. Assim, dentro de um período de amostragem da malha de realimentação externa devem existir muitos períodos de amostragem da malha de realimentação interna, de modo que a velocidade de aquisição de informações pelos sensores externos é menor do que pelos sensores internos.

Os sensores externos por sua vez podem fornecer os mais variados tipos de informação para que o robô execute de maneira não supervisionada tarefas com alto grau de complexidade. Com as informações deste sensores pode-se montar mapas de ambientes, criar modelos de objetos tanto bidimensionais como tridimensionais, tomar decisões que alteram o movimento do robô, dentre outras.

A seguir será mostrado o funcionamento de alguns tipos de sensores internos e externos, indicando sua aplicação nas diversas tarefas de um robô.

## 4.2 - SENSORES INTERNOS

### 4.2.1 - Posição

A medição da posição de um robô é feita normalmente no eixo dos atuadores que movimentam os links do robô. Diversos dispositivos podem ser utilizados para esta medição de posição. Considerando-se o caso de atuadores de rotação, estes dispositivos devem ser capazes de fornecer um valor proporcional ao ângulo de rotação do eixo. Um dos métodos de medição de posição angular mais comuns em robótica é o *encoder* óptico incremental.

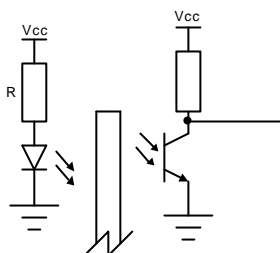
A Figura 4.2 mostra um exemplo de um *encoder* óptico incremental. O princípio de



**Figura 4.2** - Exemplo de *encoder* óptico incremental

funcionamento baseia-se na interrupção de um feixe luminoso que incide sobre um sensor óptico por meio de um disco fixo ao eixo do atuador. Este disco possui regiões transparentes e opacas para o feixe luminoso. Estas regiões alternam-se de modo que com o circuito mostrado na Figura 4.3 tem-se 50 pulsos por volta com o *encoder* da Figura 4.2. O número de pulsos por volta do

disco, ou seja, para 360° de rotação do eixo do motor pode variar com o tipo de disco e com o tamanho dos sensores ópticos utilizados. Na prática os *encoders* ópticos podem ser encontrados com até 2000 pulsos por volta ou até maiores. A Figura 4.4 mostra um exemplo de *encoder* comercial com 500 pulsos por volta.

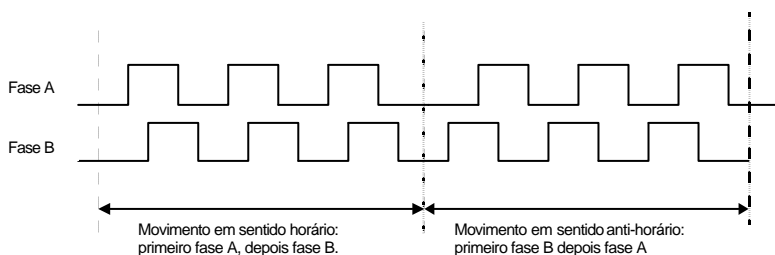


**Figura 4.3 -** Circuito para detecção da posição do encoder

Para a determinação da posição angular do eixo do atuador os pulsos provenientes do circuito da Figura 4.3 devem ser contabilizados num contador digital, de modo que a saída deste contador terá um valor proporcional a uma variação do ângulo do eixo do atuador.



**Figura 4.4 -** Exemplo de encoder óptico comercial (pedir permissão para HP)

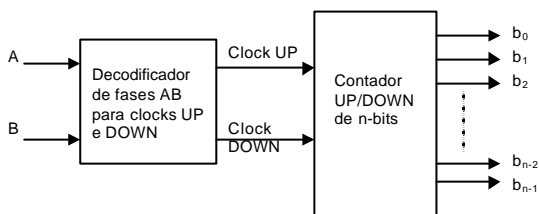


**Figura 4.5 –** Sinais dos sensores montados em quadratura

A utilização de um único sensor no *encoder* permite a determinação do ângulo de rotação do eixo do atuador, mas não determinação da direção de rotação. Neste caso é necessário a utilização de um sensor adicional montado em quadratura com o outro sensor. A Figura 4.5 mostra os sinais produzidos por este par de sensores montados em quadratura para o movimento de um único disco.

Estes sinais são denominados fases A e B. Se o eixo do motor girar no sentido horário os sinais apresentarão as bordas de subida na seqüência A-B-A-B e se o motor girar no sentido anti-horário a seqüência de bordas de subida será B-A-B-A.

Para se utilizar os sinais em quadratura com um contador digital é necessário se converter estes sinais para sinais de incremento e decremento do contador. Por exemplo, se for detectada a seqüência ABAB o circuito fornece pulsos para incrementar o contador e não para decrementar. Se a seqüência B-A-B-A for detectada o sinal produzido é para decrementar e não incrementar. Contabilizando-se os pulsos nas duas direções, o saldo será correspondente a posição angular do eixo do motor. O circuito completo para determinação da posição angular do eixo de um motor é mostrado na Figura 4.6.



**Figura 4.6** - Diagrama em blocos do circuito completo de encoder para medição de posição e direção de rotação

## 4.2.2 - Velocidade

A medida de velocidade de um robô, assim como a medida de posição, é normalmente feita no eixo do atuador. O tacogerador é um dispositivo acoplado ao eixo do motor que fornece um valor de tensão analógico proporcional a velocidade angular do eixo do motor. O funcionamento de um tacogerador é o mesmo de um gerador de corrente contínua onde a tensão de saída do gerador aumenta com o aumento da velocidade.

Contudo, a utilização de sinais analógicos em aplicações de controle digital vem diminuindo ao longo dos anos principalmente devido ao ruído que é introduzido na medida e a erros introduzidos devido a conversão analógico/digital. No caso da medição de velocidade, uma maneira digital de se fazer esta medição é a utilização do mesmo *encoder* utilizado para a medição de posição do eixo do atuador contabilizando-se o número de pulso por certo período de tempo fixo.

Esta medição pode ser feita contabilizando-se pulsos do *encoder* em janelas de tempo conhecidas. Por exemplo, no instante de tempo  $t_0$  marca-se o valor do *encoder*; no instante  $t_1$  lê-se o valor do *encoder* novamente e subtrai-se do valor marcado em  $t_0$ , a diferença obtida dividida pela janela de tempo é a velocidade média no período em pulsos por segundo. Considerando-se a janela de tempo uma constante como a que corrige pulsos por segundo em milímetros por segundo, temos que o valor da diferença é proporcional à velocidade.

No caso deste método o valor de velocidade sempre corresponderá a um valor médio no período e não a um valor instantâneo como o que pode ser obtido por um tacogerador. A utilização deste valor médio não se constitui num problema desde que o período de amostragem seja pequeno. Por isso, mesmo que as velocidades envolvidas não sejam elevadas é desejável um

número grande de pulsos por volta do *encoder* para que se possa trabalhar com janelas de curta duração.

### 4.2.3 - Aceleração

A aceleração nem sempre é um parâmetro medido em robótica mas sim calculado a partir da variação da velocidade num período de tempo.

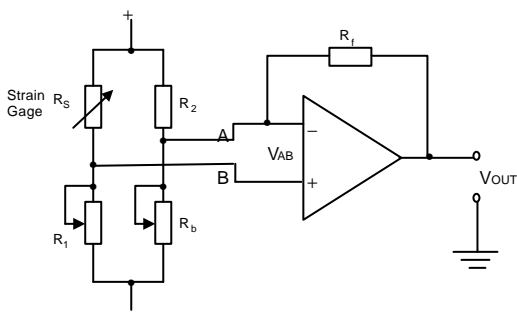
Seguindo-se o mesmo método aplicado para o cálculo da velocidade a partir dos valores lidos do *encoder*, no mesmo instante de cálculo da velocidade o próprio valor da velocidade pode ser armazenado e no próximo instante de cálculo da velocidade a diferença do valor atual com o armazenado anteriormente resulta no valor da aceleração.

### 4.2.4 - Força

A medição de força só é necessária em situações onde se deseja controlar a força que o robô exerce no objeto que está manipulando. O método mais direto de se medir forças é através da utilização de *strain gages* que produzem uma variação de resistência proporcional a uma variação de força (vide Figura 4.7). Esta variação de resistência normalmente é pequena de modo que para uma detecção mais confiável é utilizada uma ponte de Wheatstone em equilíbrio, conforme o circuito da Figura 4.8.



**Figura 4.7 - Strain Gage**



**Figura 4.8 - Montagem elétrica de um strain gauge em Ponte de Wheatstone**

## 4.3 - Sensores externos

### 4.3.1 - Presença

A detecção de presença de objetos próximos ou distantes de um sensor pode ser feita por diversas técnicas, com diversas tecnologias, dependendo da distância, do objeto e da velocidade de medição. A Tabela 4.1 resume estas informações para os tipos de sensores mais comuns em robótica.

**Tabela 4.1 - Sensores, tecnologias, distâncias de detecção**

Sensor	Tecnologia	Faixa de Detecção
Magnético	Efeito Hall	0,1 - 3 mm
Óptico	Infravermelho reflectivo (baixa potência)	1mm - 300mm
Ultra-som	Tempo-de-vôo	500mm - 10000mm
Óptico	Infravermelho ou laser reflectivo (alta potência)	3000mm - 100000mm

Dentre os sensores descritos na Tabela 4.1, os métodos de medição dos sensores de ultra-som e ópticos podem ser utilizados para medir distâncias além da presença de objetos, conforme será visto no item seguinte.

#### 4.3.1.1 - Sensor magnético por efeito Hall

O sensor magnético detecta somente objetos ferrosos em distâncias muito pequenas. O seu princípio de funcionamento baseia-se no efeito Hall onde ocorre a variação de correntes de Lorentz quando um objeto metálico está próximo do sensor. Estes sensores são utilizados em aplicações particulares para a detecção da presença de peças metálicas em correias transportadoras, montados fora do manipulador, produzem um sinal digital que pode disparar o movimento do manipulador para interagir com a peça detectada.

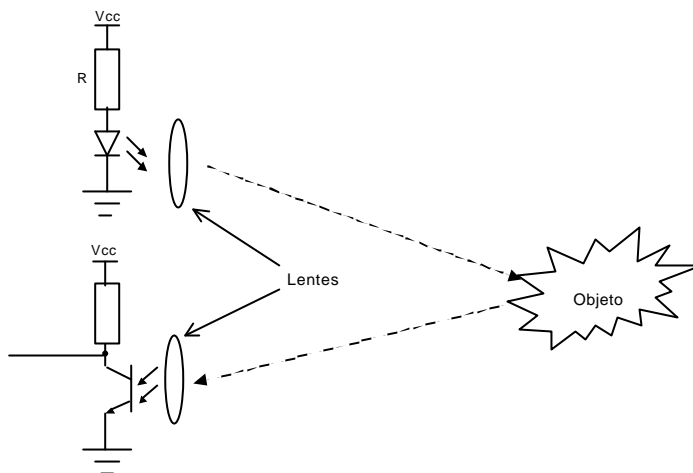


**Figura 4.9 - Sensor magnético industrial (Assemtech Europe Ltd. - <http://www.assemtech.co.uk>)**

#### 4.3.1.2 - Sensor óptico infravermelho

O sensor óptico infravermelho de baixa potência possui um emissor de luz e um receptor montados num mesmo dispositivo. A luz refletida num objeto próximo é captada pelo receptor de luz que produz um sinal digital. A Figura 4.10 mostra o arranjo elétrico para este sensor e a Figura 4.11 mostra um exemplo de um sensor industrial disponível comercialmente. Como este princípio baseia-se na reflexão de luz, a distância de detecção pode variar de acordo com as características da superfície do objeto.

Com este tipo de sensor é possível detectar a presença de objetos próximos ao sensor onde a distância até o objeto pode ser ajustada dentro de uma faixa alterando-se o ganho do receptor.



**Figura 4.10 - Eletrônica para detecção óptica de presença.**



**Figura 4.11 - Exemplo de sensor comercial (Balluff - <http://www.balluff.com>).**

## 4.3.2 - Distância

Em algumas aplicações é necessário se medir distâncias entre um sensor e objetos de interesse na área de trabalho do robô. Com algumas das tecnologias vistas anteriormente para a detecção de objetos é possível a medição de distâncias dentro de certos limites.

### 4.3.2.1 - Sensor de Ultra-som

A medição de distância com um sensor de ultra-som é feita pelo método que é conhecido por tempo-de-vôo do eco refletido num objeto produzido por um pulso de ultra-som emitido na direção deste objeto. O transdutores de ultra-som podem ser construídos por diversas tecnologias para a propagação no ar. Em aplicações na área da robótica os sistemas de medição de distâncias com ultra-som mais comuns são baseados em transdutores da Polaroid (vide Figura 4.12) com os quais é possível medir distâncias entre 70 cm e 10 metros.



**Figura 4.12 – Três diferentes modelos de transdutores de ultra-som da Polaroid.**

### 4.3.2.2 - Sensor Óptico

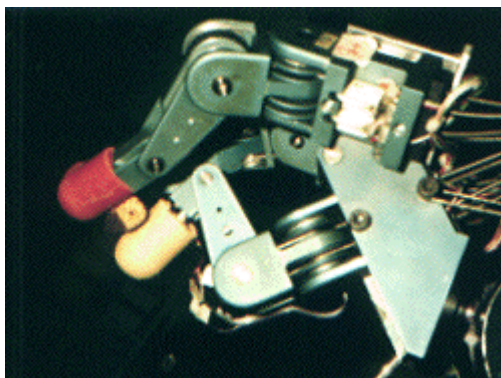
Sensores ópticos para medição de distância podem empregar diversos métodos tais como:

- Triangularização da luz - onde a luz é enviada até o objeto por caminhos diferentes o que permite a triangularização da luz e conseqüentemente a estimativa da distância até o objeto. Este método é adequado para distâncias curtas, até 50 mm, com uma precisão da ordem de milímetros.
- Diferença de fase - a luz modulada é enviada até o objeto e a sua reflexão é captada no sensor. A diferença de fase entre a luz emitida e a recebida dá uma estimativa da distância até o objeto. Este método pode ser empregado tanto com emissores de baixa potência como de alta potência o determina a faixa de distâncias que podem ser medidas.
- Tempo-de-vôo - um pulso de luz de alta potência é emitido em direção a um objeto e espera-se o retorno do pulso refletido. O tempo decorrido entre a emissão e a recepção é proporcional a distância até o objeto. A utilização de luz coerente (laser) permite a medição de distâncias grandes, contudo para aplicações em robótica a utilização de luz infravermelha não coerente, focalizada por lentes é suficiente para distâncias até cerca de 20 metros.

### 4.3.3 - Tato

A detecção de contato com o objeto e o valor da pressão exercida num objeto podem ser úteis na manipulação de objetos deformáveis e frágeis. Diversas tecnologias estão disponíveis para a implementação de sensores de tato. Algumas de fácil aplicação industrial e outras disponíveis mais a nível experimental.

Borrachas condutoras podem ser colocadas nas pontas da garra do robô de maneira que ao se fazer o contato entre a garra e o objeto a resistência da borracha começa a diminuir com o aumento da pressão. Fazendo-se uma montagem em ponte de Wheatstone, como no caso dos *strain gauges*, pode-se detectar variações bem pequenas na resistência da borracha o que dará um valor inversamente proporcional a força exercida pela garra no objeto. A Figura 4.13 mostra uma garra de robô com três dedos desenvolvido pela Universidade de Stanford em conjunto com o JPL da Nasa com a qual é possível determinar a força de contato dos dedos com o objeto e assim controlar o seu movimento.



**Figura 4.13** – Garra com três dedos de Stanford JPL com sensor de tato.



## 4.4 - VISÃO ROBÓTICA

Visão se mostra como um dos sentidos mais importantes para a interação de um agente (humano ou robótico) com o mundo. Visão computacional, também chamada de entendimento de imagens ou visão de máquinas, descreve a *determinação automática* da estrutura e propriedades do ambiente tridimensional no qual o agente está inserido, ambiente este possivelmente dinâmico, a partir de uma única ou múltiplas imagens bidimensionais do ambiente. As imagens podem ser monocromáticas ou coloridas, capturadas por uma ou múltiplas câmaras que, por sua vez, podem ser estacionárias ou móveis (NALWA, 1993).

A estrutura e propriedades a serem extraídas do ambiente podem ser geométricas (formas, dimensões, localizações de objetos, etc) ou materiais (cores, texturas, etc). Além disso, se o ambiente estiver mudando durante o processo de imageamento, deseja-se também inferir a natureza desta mudança, assim como fazer inferências acerca do futuro.

Visão não é uma tarefa simples, uma vez que o processo de formação de imagens consiste num mapeamento muitos-para-um: uma grande variedade de superfícies, com diferentes materiais e condições de iluminação, pode produzir imagens idênticas (MARR, 1982).

A abordagem básica adotada neste capítulo consiste em primeiro entender como os estímulos sensoriais são criados pelo ambiente e então questionar: “se os estímulos sensoriais são criados de tal e tal forma pelo ambiente, então como deveria ser o mundo para que produzisse tais estímulos?” (RUSSEL& NORVIG, 1995).

De certa forma, pode-se pensar em visão computacional como a tentativa de “desfazer” parcialmente o processo de formação da imagem, daí ser considerado um *problema inverso*. Infelizmente, não existe uma inversão própria e correta. No entanto, algumas restrições podem ser impostas para resolver satisfatoriamente a ambigüidade inerente ao processo visual. Além disso, o agente não precisa saber “tudo” acerca do ambiente para poder atuar com sucesso. Dependendo da aplicação, somente alguns aspectos tornam-se necessários. Por exemplo, em tarefas de manipulação, as ações de segurar peças, encaixar, inserir, etc. que são realizadas, necessitando somente de informações de forma e localização das peças a serem manipuladas; em tarefas de navegação, ações como encontrar caminho livre, evitar obstáculos, calcular velocidade e orientação, etc. que são necessárias, e assim por diante. Nenhuma destas tarefas requer a determinação da descrição completa do ambiente para ser realizada com sucesso.

### 4.4.1 - Formação de Imagens

O olho humano forma uma imagem invertida da cena em sua retina. A retina, por sua vez, captura a imagem, codifica-a e transmite esta imagem codificada ao cérebro. A câmara de um sistema de visão computacional executa um papel análogo ao do olho no sistema visual humano.

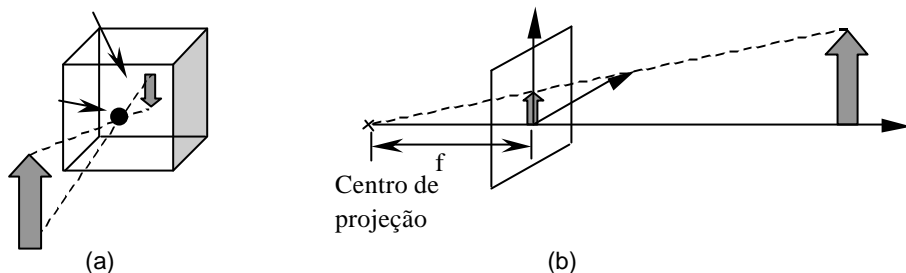
A formação de uma imagem ocorre quando um sensor registra radiação que interagiu com objetos físicos (BALLARD&BROWN, 1982). Primeiro estudaremos rapidamente a geometria da formação da imagem. Depois, a relação entre a quantidade de luz que irradia de uma superfície e a quantidade de luz observada na imagem desta superfície. Finalmente, estudaremos o processo de sensoreamento desta imagem, transformando-a em uma representação passível de ser armazenada, processada e analisada por um computador.

#### 4.4.1.1 - Geometria do imageamento

O dispositivo de imageamento mais simples existente é a *câmara “pinhole”*. Idealmente, esta câmara possui um orifício extremamente pequeno (“pinhole”), por onde entra luz na câmara, formando uma imagem geometricamente invertida na face oposta da câmara. A geometria da formação da imagem numa câmara “pinhole” é bem modelada pela *projeção perspectiva*. Projeção perspectiva é a projeção de uma entidade tridimensional em uma superfície

bidimensional, determinada por retas que passam por um único ponto, chamado centro de projeção.

A Figura 4.14 mostra a imagem formada numa câmara “pinhole” e sua correspondente configuração pela projeção perspectiva, com a imagem já representada do mesmo lado da cena, em relação ao centro de projeção. A distância do centro de projeção ao plano de imagem (ou plano focal) é chamada distância focal  $f$ . Considere o plano de imagem mostrado na Figura 4.14; um ponto  $(x_c, y_c, z_c)$  da cena é mapeado no ponto  $(x_i, y_i, 0)$  no plano de imagem. Os pontos  $x$  e  $y$  derivam das seguintes relações:  $x_i/f = x_c/(f - z_c)$  ;  $y_i/f = y_c/(f - z_c)$  (MARR, 1982).



**Figura 4.14** - (a) Câmera “pinhole” e (b) seu respectivo modelo geométrico pela Projeção Perspectiva.

Para que informações geométricas 3D sejam derivadas das imagens 2D, torna-se necessário que se determine a relação entre a posição de um ponto na cena e a posição em sua imagem. Esta determinação recebe o nome de *calibração da câmara*. Tipicamente, a calibração da câmara se dá usando um objeto de calibração conhecido, cuja imagem exibe um grande número de pontos distintos, facilmente identificáveis e precisamente localizados como por exemplo, uma grade retilínea planar (BALLARD, BROWN, 1982; SHIRAI 1987).

#### 4.4.1.2 - Radiometria

No segmento anterior vimos a relação geométrica entre o ambiente e a imagem. Agora será abordada a relação radiométrica entre eles.

O *brilho* de um ponto  $p$  na imagem é proporcional à quantidade de luz direcionada à câmara por uma superfície infinitesimal  $S_p$ , cuja projeção seja  $p$ . Esta quantidade de luz, por sua vez, depende das propriedades de reflectância de  $S_p$ , além da posição e distribuição das fontes de luz. Como as outras superfícies da cena também podem ser fontes indiretas de luz (ao refletir em  $S_p$  a luz recebida), suas propriedades de reflectância também afetarão a referida quantidade de luz (RUSSEL, NORVIG, 1995).

Assim, a intensidade  $I(x_i, y_i)$  de luz refletida capturada na imagem num determinado instante  $t$  (que corresponde ao brilho de  $p$ ) é dada por:  $I(x_i, y_i) = k \cdot a(x_c, y_c, z_c) \cdot \Phi(n, s, v)$ , onde  $k$  é uma constante determinada pelas características da câmara;  $a(x_c, y_c, z_c)$  corresponde à intensidade de luz incidente total no ponto da cena  $(x_c, y_c, z_c)$  e  $\Phi(n, s, v)$ , à reflectância da superfície em  $(x_c, y_c, z_c)$  (MARR, 1982).

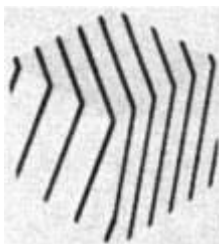
Uma superfície perfeitamente difusa apresenta o mesmo brilho para um observador, qualquer que seja a direção em que ele esteja; Já numa superfície perfeitamente especular refletirá somente na direção em que os raios refletidos estejam no mesmo plano dos raios incidentes e que satisfaça a condição de que o ângulo de reflexão seja o mesmo que o de incidência. As superfícies reais apresentam uma combinação das propriedades difusa e especular.

#### 4.4.1.3 - Sensoreamento

Até o momento foi considerada a relação entre a cena e a imagem formada no plano de imagem, chamada imagem “ótica”. Através do sensoreamento pode-se transformar esta imagem “ótica” em uma imagem “elétrica” (NALWA, 1993). Uma grande variedade de sensores pode ser utilizada nesta tarefa (BALLARD, BROWN, 1982; SHIRAI, 1987). O sensor mais largamente utilizado em visão computacional é uma câmara CCD (“charge coupled device”), que consiste numa matriz de elementos fotossensíveis integrados numa pastilha (“chip”). As câmaras enviam sinal analógico para uma placa digitalizadora de imagens, que converte este sinal para um sinal digital a ser processado. Atualmente, câmaras digitais, que já fornecem o dado digital para o processador, têm se tornado cada vez mais populares.

Deve-se também ressaltar que a possibilidade de controle do ambiente onde o sistema de visão computacional (e o agente robótico) irá atuar pode facilitar em muito o processo da visão, permitindo resolver ambigüidades, garantindo imagens de maior qualidade, possibilitando ressaltar atributos de interesse da cena, reduzindo ruídos, simplificando os algoritmos usados para a interpretação, etc. Este controle do ambiente envolve diversos fatores, desde a colocação de adesivos com código de barras em vários lugares, visando auxiliar a tarefa de localização de robôs, até o uso de sensores a laser para medir distâncias.

Uma prática bastante difundida em ambientes industriais consiste no uso de *luz estruturada*, onde o próprio sensor projeta uma estreita faixa de luz nos objetos, para facilitar o problema de determinação de sua forma e podendo, inclusive, facilitar a determinação da distância do objeto ao sensor (“range distance”). Assim, o projetor corta o objeto com um feixe planar de luz. A câmara, posicionada adequadamente, captura a imagem que, no caso, será de informação essencialmente unidimensional, pois conterá somente a estreita faixa de luz deformada segundo a geometria da cena. A figura 4.15 mostra este procedimento.



**Figura 4.15** – Técnica de luz estruturada, onde faixas de luz são projetadas num cubo.

O plano de luz pode resultar de um laser ou da projeção através de uma fenda. Adequando a câmara convenientemente, uma vez que somente a faixa de luz deve ser visível, o primeiro caso (laser) apresenta vantagens, pois o sistema pode operar com uma cena iluminada normalmente, além de conseguir um plano de luz mais estreito.

Movendo a faixa de luz (ou movendo o objeto através da faixa), pode-se conseguir uma imagem densa de distâncias. Os “laser range finders” são sensores que incorporam a fonte de laser, o controle da movimentação do feixe de luz, a câmara e todo processamento de imagem necessário para computar um mapa de distâncias dos pontos na imagem (SHIRAI, 1987).

Outra maneira de se conseguir um mapa de distâncias é pelo uso de estéreo binocular, semelhante ao do ser humano. Neste modo de aquisição de imagem de distância, as coordenadas dos planos de imagens de duas câmaras estão relacionadas por uma transformação conhecida; identificando-se a projeção de determinado ponto da cena nas duas imagens pode-se determinar, por triangularização, a distância de cada ponto da cena à câmara. Estéreo multiocular atua do mesmo modo, porém fornecendo mais informação para que se determine qual ponto de

uma imagem corresponde a qual ponto da outra imagem (SHARAI, 1987; HORN, 1986; BALLARD, BROWN, 1982).

No item anterior a intensidade (brilho) de uma imagem foi mencionada ignorando-se o fato de que a luz visível se apresenta numa faixa de comprimentos de onda  $\lambda$ , variando de 400nm para violeta a 700nm para o vermelho. Assim, é necessário considerar a composição espectral do brilho da imagem<sup>1</sup>.

Denota-se brilho de uma imagem por  $I(x,y,t,\lambda)$ , onde  $(x,y)$  representa as coordenadas dos pontos no plano da imagem,  $\lambda$  é o comprimento de onda no instante de tempo  $t$ . A imagem "elétrica"  $I_e(x,y)$  no instante  $t$  é dada por:  $I_e(x,y) = \int I(x,y,t,\lambda) \cdot s(\lambda) \cdot d\lambda$ , onde  $s(\lambda)$  corresponde à sensibilidade espectral do sensor (dispositivo). No olho humano existem três diferentes tipos de sensores, chamados cones, cada um com uma sensibilidade espectral  $s(\lambda)$  diferente: um para cor vermelha, outro para azul e o terceiro, para verde. As câmaras coloridas tipicamente incorporam estas três diferentes funções de sensibilidade espectral. Neste item os estudos estão restritos a imagens monocromáticas, que são capturadas usando uma única função de sensibilidade espectral (imagens em níveis de cinza).

#### 4.4.2 - Imagem Digital

Um sinal que tenha um domínio contínuo e uma faixa de variação contínua é chamado de sinal *analógico*; já um sinal que tenha um domínio discreto e uma faixa de variação também discreta é chamado de sinal *digital*. *Imagem Digital* é uma matriz de inteiros, onde cada inteiro representa o brilho da imagem num tempo discreto e num ponto discreto do plano da imagem (NALWA, 1993). Um ponto desta matriz é denominado *pixel*.

A transformação da imagem elétrica numa imagem digital envolve um processo de amostragem e um de quantização. A *amostragem* discretiza o domínio da função imagem, fornecendo a cada ponto discreto o valor da função naquele ponto. A *quantização* discretiza a faixa de variação da função, mapeando o valor da função em cada ponto a um valor dentre vários valores discretos. Tipicamente os valores discretos assumidos pelos pontos numa imagem monocromática variam de 0 (preto) a 255 (branco). A Figura 4.16 mostra uma imagem e sua matriz correspondente.

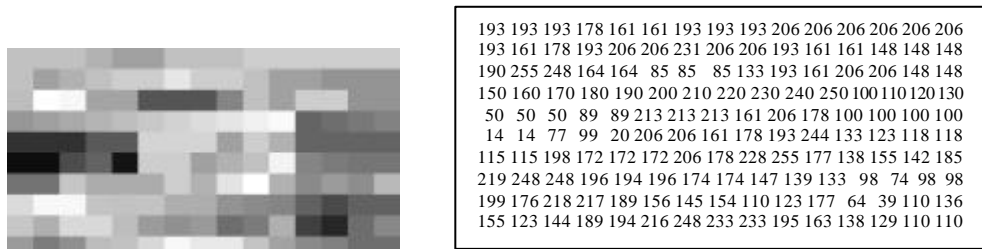


Figura 4.16 – Imagem digital e sua matriz correspondente.

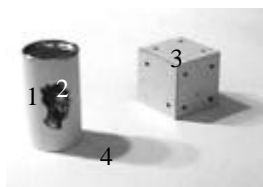
#### 4.4.3 - Processamento de imagens digitais

O processo de imageamento introduz diversos ruídos e deformações na informação contida nas imagens digitais. No entanto, esta informação não está completamente perdida devida à grande redundância espacial existente, isto é, pixels vizinhos numa imagem apresentam os

<sup>1</sup> Numa análise mais rigorosa, deve-se considerar não só a composição espectral do brilho da imagem, mas também sua dependência em relação ao tempo (NALWA, 1993).

mesmos (ou quase os mesmos) parâmetros físicos. Um grande conjunto de técnicas exploram esta redundância para recuperar a informação contida na imagem. Estas técnicas executam transformações na imagem, procurando extrair parâmetros intrínsecos tais como descontinuidades de superfícies, distância, orientação de superfícies, entre outros.

Um dos objetivos mais comuns no processamento de imagens consiste em eliminar ruídos e extrair as bordas (“edges”) contidas numa imagem. Bordas são curvas no plano de imagem através das quais existe uma “significante” mudança no brilho (RUSSEL, NORVIG, 1995). O interesse nas bordas de uma imagem consiste no fato de que elas correspondem a importantes contornos da cena. A Figura 4.17 mostra descontinuidades na cena que causam bordas na imagem: descontinuidades de reflectância, de profundidade, de orientação da superfície e de iluminação (sombras).



1. Descontinuidade de profundidade
2. Descontinuidade de reflectância
3. Descontinuidade de orientação
4. Descontinuidade de iluminação

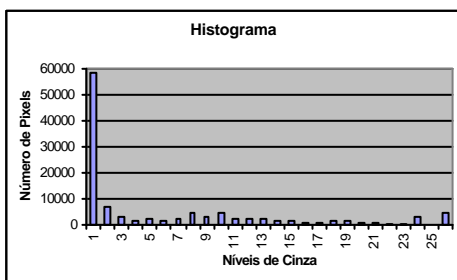
**Figura 4.17** – Diferentes tipos de bordas numa imagem.

Basicamente, o conjunto de operadores que podem ser aplicados numa imagem dividem-se em três classes: operadores pontuais, locais e globais. A seguir serão dados alguns exemplos de cada classe.

#### 4.4.3.1 - Operadores Pontuais

Operadores pontuais são aqueles nos quais os valores resultantes de brilho num pixel dependem somente de seu valor anterior. Exemplos deste operador incluem operações aritméticas entre imagens, limiarização e manipulações do histograma da imagem.

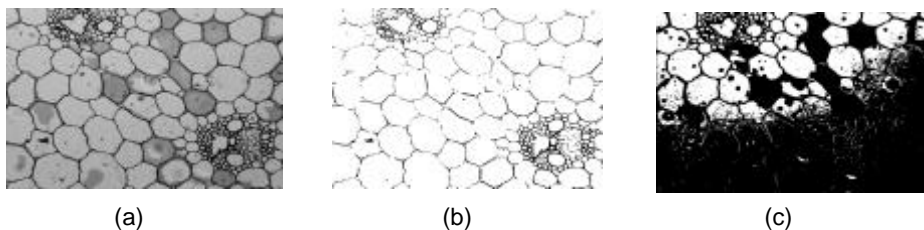
*Histograma* de uma imagem monocromática (em níveis de cinza) é uma função  $h(k)$  que fornece o número total de pixels que possuem um valor  $k$  de cinza (valor do brilho), conforme exemplificado na Figura 4.18.



**Figura 4.18** - Exemplo de histograma de uma imagem monocromática.

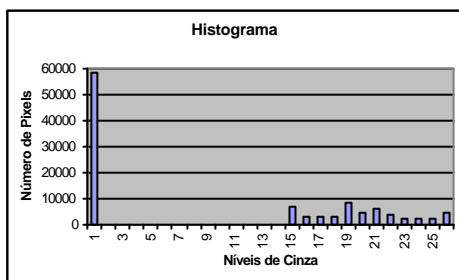
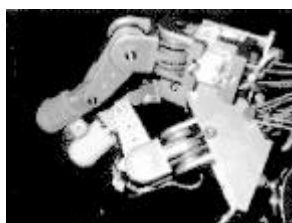
*Limiarização* é um dos processos mais simples de dividir uma imagem em regiões. A *binarização* é o caso particular da limiarização, onde a imagem é dividida em duas regiões (0 ou 1, branco ou preto), de acordo com um valor de limiar  $L$ ; assim, pixels com valor de cinza acima de  $L$  recebem valor 1 (ou branco) e recebem 0 (ou preto) no caso contrário. Tipicamente, o valor de limiar é estabelecido a partir da análise do histograma da imagem: se o objeto de interesse

apresentar grande contraste com o fundo, o histograma se configurará bimodal e o valor de cinza para o limiar pode ser estabelecido no ponto de mínimo entre os dois máximos do histograma. A Figura 4.19 apresenta exemplos de imagens que foram submetidas ao processo de limiarização, com diferentes valores de  $L$ .



**Figura 4.19** – Exemplos de limiarização, transformando uma imagem monocromática numa imagem binária, através do uso de diferentes valores de limiar: (a) imagem original; (b) imagem original após aplicação de limiar no nível de cinza 50 e (c) imagem original após aplicação de limiar no nível de cinza 180.

Manipulações do histograma incluem esticamento do histograma, equalização do histograma, entre outros. O esticamento do histograma realiza uma operação de normalização onde ajusta-se os valores mínimos e máximos dos níveis de cinza existentes numa imagem para os novos valores mínimos e máximos estipulados para o histograma. A equalização do histograma visa aproximar o histograma cumulativo a uma reta (ou qualquer função que se queira), para adquirir distribuição uniforme de níveis de cinza. Estas operações visam melhorar o contraste na imagem, conforme exemplifica a Figura 4.20.



**Figura 4.20** – Equalização do histograma da imagem da Figura 4.16.

#### 4.4.3.2 - Operadores Locais

Operadores locais são aqueles que se utilizam de uma janela (ou máscara) envolvendo o pixel de interesse e o valor resultante deste pixel é função dos valores do referido pixel e seus vizinhos incluídos na janela. Os operadores locais podem ser operadores lineares ou não lineares.

Uma operação muito importante em processamento de imagens é a convolução. *Convolução*, denotada por  $*$ , é a operação que computa a integral ponderada (ou soma, no caso de funções discretas) de uma função em relação a outra função que primeiramente foi refletida em relação à origem e então deslocada. No caso de funções discretas bidimensionais, como imagens digitais, vem:  $H(m,n) * G(m,n) = \sum_i \sum_j H(i,j) G(m-i,n-j) = \sum_i \sum_j H(m-i,n-j) G(i,j)$ , com  $i$  e  $j$  variando dentro dos limites da imagem.

#### 4.4.3.3 - Operadores Locais Lineares

Exemplos típicos de operadores locais lineares são o operador de média, operador gradiente e o operador laplaciano.

O operador de média suaviza a imagem, pois substitui o valor de cada pixel pelo valor da média dos valores dos pixels incluídos na janela envolvente. Este operador pode ser conseguido através da convolução discreta da imagem original por uma máscara de valores 1. Na operação de convolução, a máscara é colocada sobre cada pixel (excluindo os pixels da borda da imagem por apresentarem problemas de vizinhança) e a operação de soma dos produtos de cada pixel (pixel atual e seus vizinhos incluídos na máscara) pelo correspondente ponto da máscara é feita; o resultado desta operação será o novo valor do pixel. Assim, por exemplo, o novo valor do pixel I(1,1) será:  $1 \times 1 + 1 \times 2 + 1 \times 3 + 1 \times 1 + 1 \times 1 + 1 \times 2 + 1 \times 2 + 1 \times 3 + 1 \times 1 = 25$  (figura 4.21).

O operador gradiente e o laplaciano, por sua vez, têm o efeito oposto ao de média, já que acentuam as diferenças dos valores dos pixels, isto é, as descontinuidades dos valores de cinza. O gradiente resulta em dois componentes: (i) magnitude  $G = (D_x^2 + D_y^2)^{1/2}$ , e (ii) orientação  $\phi = \tan^{-1}(D_y/D_x)$ , com  $D_x \neq 0$ . Uma aproximação muito usada é considerar  $G = D_x + D_y$ . A Figura 4.21 apresenta um exemplo do operador média e do laplaciano, usando máscaras 3x3. Já a Figura 4.22 mostra algumas máscaras muito usadas para o operador gradiente: o operador de Robert, o operador Prewitt e o operador Sobel.

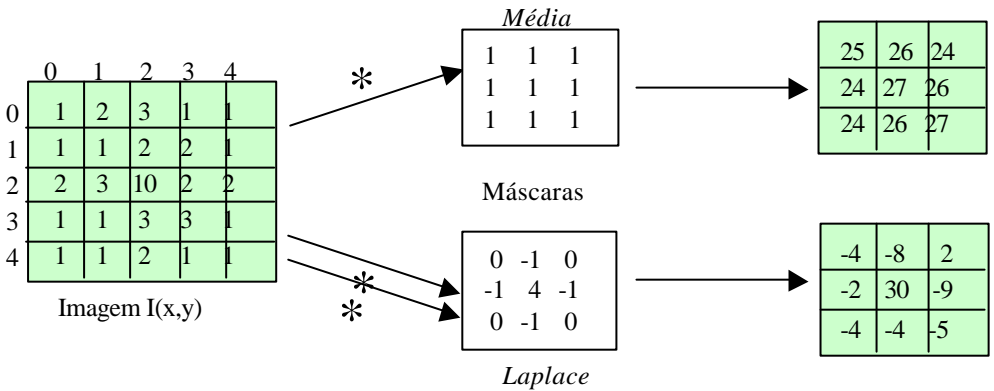


Figura 4.21 – Exemplo de operadores lineares: média e laplaciano utilizando máscaras 3x3.

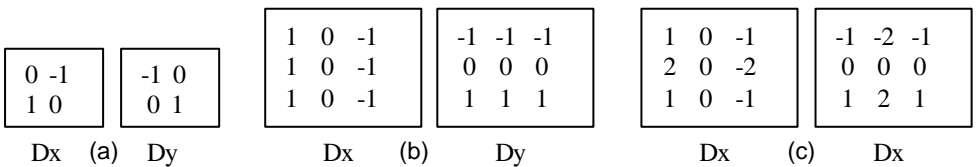


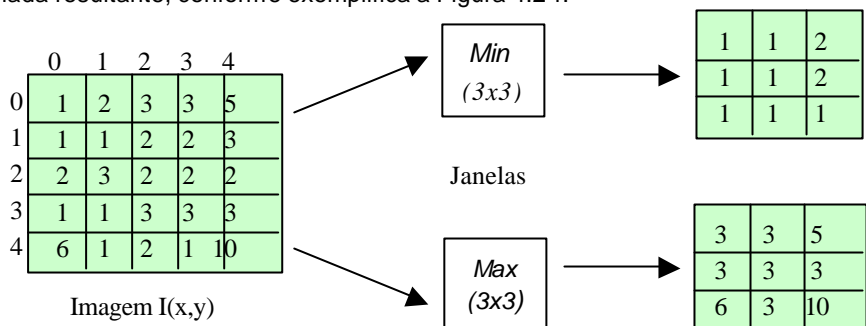
Figura 4.22 – Operadores gradientes: máscaras de (a) Robert; (b) Prewitt e (c) Sobel.

#### 4.4.3.4 - Operadores Locais Não-lineares

Exemplos de operadores locais não-lineares são o operador Min, operador Max e o operador mediano. O operador Min é usado para eliminar picos de altos valores de brilho, sem suavizar a imagem inteira. Sua aplicação resulta também num aumento de manchas de baixos valores de brilho. Ele consiste em colocar uma janela ao redor de cada pixel e analisar todos os valores contidos na janela, buscando o mínimo valor, que será adotado como novo valor do pixel central da janela.

O operador Max é o oposto do Min. Ele é usado para eliminar pequenas manchas de baixos valores de cinza e aumenta os picos de altos valores de cinza. Ele busca, dentro da janela, o maior valor, o qual será designado como novo valor para o pixel analisado. A Figura 4.23 exemplifica a aplicação dos operadores Min e Max (as bordas da imagem foram excluídas).

O operador mediano une as funções de Min e Max, evitando suas desvantagens, porém com um acréscimo considerável no custo computacional. Consiste em ordenar os valores contidos na janela ao redor do pixel analisado, assinalando como novo valor do pixel o valor mediano da lista ordenada resultante, conforme exemplifica a Figura 4.24.



(a)

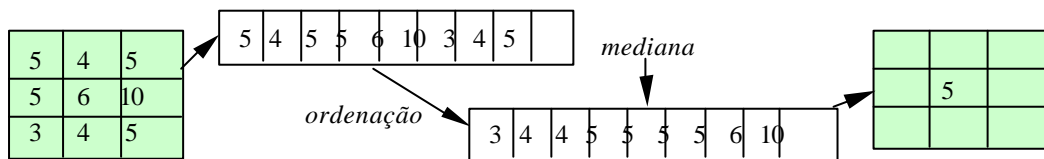


(b)



(c)

**Figura 4.23** – Exemplo dos operadores não-lineares Min e Max para janela 3x3 e sua aplicação: (a) imagem original; (b) após operador Min e (c) após operador Max.



**Figura 4.24** – Exemplo do operador mediano para janela 3x3.

#### 4.4.4 - Operadores Globais

Um exemplo típico de operador global é a *Transformada de Fourier*. A imagem é uma função que varia espacialmente. Deste modo, pode-se aplicar a transformada de Fourier (discreta) para que a imagem de intensidades seja transportada para o domínio da frequência espacial, sendo que baixas frequências espaciais correspondem a uma imagem com o nível de cinza variando suavemente e altas frequências espaciais, a variações bruscas no nível de cinza da imagem. Diversas filtragens e manipulações na imagem podem ser feitas no domínio da frequência espacial (GONZALEZ, WOODS, 1993; BALLARD, BROWN, 1982).

#### 4.4.5 - Segmentação de Imagens

Os operadores apresentados no item anterior visam basicamente eliminar ruídos e ressaltar bordas na imagem. Combinando operações de suavização da imagem (aplicando operador de média, por exemplo) para reduzir ruídos e operações de diferenciação (operador laplaciano, por exemplo), pode-se evidenciar as bordas numa imagem. Bordas correspondem a linhas divisórias



entre regiões homogêneas numa imagem (regiões com pouca variação no nível de cinza), as quais geralmente correspondem a superfícies de interesse na cena. A determinação de regiões homogêneas numa imagem é denominada *segmentação* da imagem e constitui, portanto, um passo muito importante na direção da interpretação semântica da cena.

Infelizmente, o uso exclusivo de detecção de bordas pode não ser adequado à segmentação da imagem por dois motivos (RUSSEL, NORVIG, 1995): (i) alguma parte da borda correspondente a um contorno de superfície pode apresentar baixo contraste e não ser detectada; (ii) muitas bordas detectadas podem ser ruídos, representar sombras ou marcas sem importância nas superfícies. Desta forma, métodos complementares de segmentação tornam-se necessários.

Basicamente, tem-se duas abordagens para a segmentação: (i) por análise das *descontinuidades* (nos níveis de cinza) existentes na imagem – que correspondem à detecção de bordas e (ii) por análise das *similaridades* entre elementos próximos.

Um ponto muito importante no processo de segmentação consiste na estrutura de dados usada para representar os grupos homogêneos determinados na imagem. A definição destas estruturas está intimamente ligada à abordagem adotada para a segmentação.

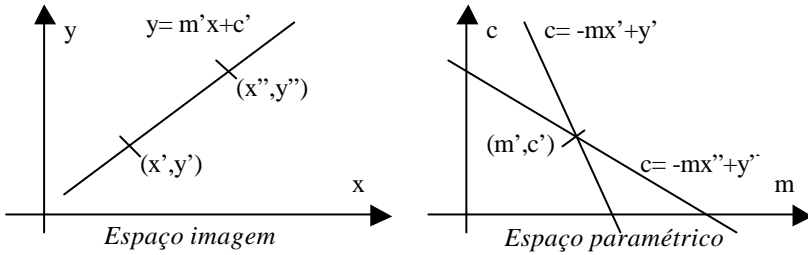
#### 4.4.5.1 - Segmentação por Descontinuidade

Na segmentação por descontinuidade, a imagem é particionada segundo variações bruscas nos valores dos pixels. No item anterior foram mostrados alguns operadores que podem evidenciar pixels que pertencem às bordas. No entanto, estes pixels precisam ser adequadamente associados (conectados) e representados de modo eficiente, visando não só descrever a imagem em termos de suas bordas, mas também facilitar o processo de correspondência entre as estruturas extraídas da imagem e as armazenadas em modelos dos objetos do domínio.

O método de conexão de Robert aplica-se na imagem resultante após a aplicação do operador de Robert (veja Figura 4.22). O primeiro passo consiste em selecionar candidatos através da análise de janelas 4x4, selecionando os pixels de maior magnitude do gradiente. A seguir, para cada candidato, uma região é analisada para determinar a direção (do gradiente) predominante; caso não exista predominância, o candidato é descartado. Os candidatos são conectados se estiverem dentro da região de análise (4x4) e se a diferença entre suas direções não passar de um limite preestabelecido.

Outro método muito usado é a Transformada de Hough, usada quando não se sabe a localização de uma borda, mas sua forma pode ser descrita por uma curva paramétrica (aqui consideraremos somente retas). A vantagem do método é que o resultado não é muito afetado com ruídos e quebras nas curvas. Aplica-se o método numa imagem de pontos candidatos (por exemplo, imagem diferenciada). A idéia básica consiste em que um ponto  $(x',y')$  no espaço da imagem corresponde à reta  $c=-mx'+y'$  no espaço paramétrico. Um segundo ponto  $(x'',y'')$  na imagem, pertencente a uma reta (borda) que passa também por  $(x',y')$ , corresponde a outra reta  $c=-mx''+y''$  no espaço paramétrico. Como  $(x',y')$  e  $(x'',y'')$  pertencem à mesma reta na imagem, as retas  $c=-mx'+y'$  e  $c=-mx''+y''$  se cruzarão no ponto  $(m',c')$  no espaço paramétrico, conforme exemplifica a Figura 4.25.

A detecção de retas se dá da seguinte forma: o espaço de parâmetros é discretizado, formando uma matriz acumuladora  $A(c,m)$ , inicialmente zerada. Para cada ponto candidato de borda  $(x,y)$  da imagem, fazer  $A(c,m) = A(c,m)+1$ , com  $c$  e  $m$  satisfazendo  $c=-mx+y$  (valores discretos). No final, os pontos de máximos em  $A$  representarão pontos colineares na imagem, sendo que o valor acumulado em  $A$  no ponto de máximo indica quantos pontos da imagem compõem a referida reta.



**Figura 4.25** – Transformada de Hough:  $(x',y')$  no espaço imagem corresponde a  $c=-mx'+y'$  no espaço paramétrico. Da mesma forma,  $(x'',y'')$  também tem uma reta correspondente. Se  $(x',y')$  e  $(x'',y'')$  pertencem à mesma reta na imagem, suas retas correspondentes no espaço paramétrico irão se cruzar em  $(m',c')$  e a reta na imagem será  $y = m'x+c'$ .

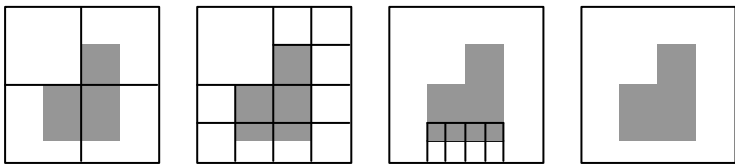
Um terceiro método usado para conectar pixels de bordas é um seguidor de bordas por busca em grafo. Um grafo é composto por nós  $\{n_i\}$  e arcos  $\langle n_i, n_j \rangle$ . Custos são associados aos arcos. A busca por uma borda consiste na busca por um caminho de menor custo entre dois nós de um grafo. Na imagem gradiente, associa-se a cada arco o custo correspondente à magnitude do gradiente e os nós do grafo, à direção do gradiente. Coloca-se restrições quanto a variações da direção do gradiente em pontos adjacentes (por exemplo, ser menor que  $\pi/2$ ). Pode-se usar busca heurística para decidir o caminho.

**4.4.5.2 - Segmentação por Similaridade**

A abordagem por similaridade para segmentação é dual da abordagem anterior. Como nenhuma apresenta resultados perfeitos, elas podem cooperar entre si para alcançar resultados melhores. A abordagem por similaridade consiste em agregar pixels com características similares (no caso, brilho) em regiões. Um ponto  $x_i$  em uma região R está conectado a  $x_j$  se existir uma seqüência  $\{x_i, \dots, x_j\}$  tal que  $x_k$  e  $x_{k+1}$  são conectados e todos os pontos estão em R. R é uma região conectada se todos os pontos x em R formarem pares de pontos conectados. Uma imagem será a união de todas as regiões  $I = \cup R_k$ . As regiões são disjuntas, isto é,  $R_i \cap R_j = 0$ , com  $i \neq j$ .

O método de limiarização descrito anteriormente (item 4.4.3.1) é provavelmente o mais simples nesta categoria. Inicialmente, pode-se suavizar a imagem. Depois, computa-se o histograma. Determina-se picos e vales. Seleciona-se o pico mais alto, determinando dois valores limites (nos vales anterior e posterior). Divide-se a imagem em subregiões, com base nestes valores. Repetir o procedimento para cada subregião, até que não existam mais picos significativos no histograma.

O método de *dividir-e-agregar* (“split-and-merge”) consiste em dividir em quatro regiões disjuntas qualquer região  $R_i$  (inicialmente, a imagem que pode ter sido anteriormente suavizada) onde o critério de homogeneidade não for satisfeito  $H(R_i)=FALSE$ . Agregar quaisquer regiões adjacentes  $R_i$  e  $R_j$  para as quais  $H(R_i \cup R_j)=TRUE$ . Repetir procedimento até que não seja mais possível fazer divisões nem agregações. A Figura 4.26 ilustra este procedimento.



**Figura 4.26** – Segmentação por similaridade usando o método de dividir-e-agregar.

O método de *rotulação* é também um método bastante popular de segmentação. Considere uma imagem binária (todos os pixels ou são pretos, ou brancos, que representaremos 0 ou 1), onde o fundo=0 e objetos=1. Rastreando a imagem de cima para baixo, da esquerda para a direita, e considerando o pixel atualmente analisado como  $x_a$ , pixel superior como  $x_s$  e pixel à esquerda como  $x_e$ , e  $f(x)$  como o brilho do pixel, vem:

```
{cor=1;
Se  $f(x_a) = 0$ , continue (é fundo, portanto, não interessa);
Senão { (é objeto, portanto, interessa)
    se ( $f(x_s)=1$  &&  $f(x_e)=0$ ), então rótulo( $x_a$ )=rótulo( $x_s$ );
    se ( $f(x_s)=0$  &&  $f(x_e)=1$ ), então rótulo( $x_a$ )=rótulo( $x_e$ );
    se ( $f(x_s)=1$  &&  $f(x_e)=1$ ), então
        {rótulo( $x_a$ )=rótulo( $x_e$ );
        se rótulo( $x_e$ )  $\neq$  rótulo( $x_s$ ), então rótulo( $x_e$ )  $\equiv$  rótulo( $x_s$ )};
    se ( $f(x_s)=0$  &&  $f(x_e)=0$ ), então {rótulo( $x_a$ )=cor; cor=cor+1}}
```

A Figura 4.27 exemplifica a aplicação deste algoritmo de rotulação numa imagem simples.

0	0	0	0	0	0	0	0
0	1	1	0	0	1	1	0
0	1	1	1	1	1	1	0
0	0	1	1	1	1	0	0
0	0	0	0	0	0	0	0
0	1	1	0	0	1	1	0
0	1	1	0	0	1	1	0
0	0	0	0	1	1	1	0
0	0	0	0	0	0	0	0

Imagem binária original

0	0	0	0	0	0	0	0
0	1	1	0	0	2	2	0
0	1	1	1	1	1	1	0
0	0	1	1	1	1	0	0
0	0	0	0	0	0	0	0
0	3	3	0	0	4	4	0
0	3	3	0	0	4	4	0
0	0	0	0	5	5	5	0
0	0	0	0	0	0	0	0

Imagem rotulada

Lista de equivalências:

- Rótulo 1  $\equiv$  Rótulo 2
- Rótulo 5  $\equiv$  Rótulo 4

Figura 4.27 – Exemplo simples de rotulação.

#### 4.4.6 - Reconhecimento de Objetos

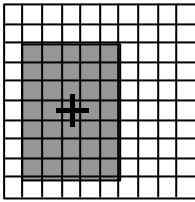
Reconhecimento consiste em identificar e determinar a posição e orientação de cada objeto na cena em relação à câmara e, com as informações adquiridas na fase de calibração, determinar sua localização em relação a um sistema de coordenadas do ambiente (ou do robô), para tarefas de manipulação, por exemplo.

No item anterior algumas técnicas de segmentação de imagens foram descritas. As regiões extraídas da imagem correspondem a objetos de interesse. Assim, propriedades como área da região, perímetro, compactividade, centróide, distâncias mínima e máxima, entre outras, são importantes na identificação e na determinação da posição e orientação dos objetos.

A *área*  $A$  de uma região consiste na somatória dos pixels de um mesmo rótulo. Para comparar a área da região com a área do objeto imageado num determinado posicionamento, deve-se ter conhecimento da calibração da câmara, que dá a relação (unidade de medida da imagem)/(unidade de medida real), além de relacionar a posição da câmara com a posição do objeto na cena. O *perímetro*  $P$  pode ser calculado pela somatória dos pixels pertencentes à borda de interesse. A *compactividade*  $C$  de uma região é definida como sendo a relação:  $C = P^2/4pA$ .

O *centróide*, que no caso corresponde ao centro de área, é usado para determinar a posição dos objetos e é dado por:  $\bar{x} = (\sum x \sum y + f(x,y))/A$ ;  $\bar{y} = (\sum x \sum y \cdot x f(x,y))/A$ , sendo  $A$  a área, e as somatórias aplicadas à região de interesse. A Figura 4.28 ilustra este procedimento numa imagem

10x10, com quina superior esquerda sendo (0,0). A região de interesse encontra-se de (1,2) a (5,8).



Área = 35

$$x_c = (7.1+7.2+7.3+7.4+7.5)/35 = 3$$

$$y_c = (5.2+5.3+5.4+5.5+5.6+5.7+5.8)/35 = 5$$

Centróide  $(x_c, y_c) = (3,5)$

**Figura 4.28** – Cálculo do centro de área de uma região.

As maiores e menores distâncias do centróide às bordas são determinadas percorrendo o perímetro e calculando as distâncias euclidianas do ponto da borda ao centróide, armazenando a maior  $D_{max}$  e a menor  $D_{min}$  distância e as respectivas coordenadas dos pontos da borda. A *excentricidade* ou *alongamento* de uma região é definida por  $E_x = D_{max} / D_{min}$ . Muitos outros atributos podem ser extraídos das regiões, consulte (HORN, 1986; MARR, 1982; BALLARD, BROWN, 1982; GONZALEZ, WOODS, 1993) para uma revisão mais detalhada.

Após a extração dos atributos da imagem, o próximo passo consiste em corresponder os atributos e informações extraídos da imagem àqueles previamente armazenados como modelos, numa fase de treinamento do sistema de visão.

A correspondência entre dados extraídos e modelos pode se dar simplesmente através do uso de um classificador estatístico, permitindo certas variações estatísticas no processo de comparação. Neste caso, o conjunto de atributos (do modelo e o extraído da imagem) deve conter elementos em número e importância significativa para identificar e localizar corretamente cada objeto, evitando falsos positivos (classificando um objeto como sendo de uma classe incorreta - diz que é um parafuso quando na verdade trata-se de uma caneta, por exemplo) e falsos negativos (negando a classe correta na classificação do objeto - diz que não é um parafuso, quando na verdade trata-se de um parafuso). Pode-se também realizar um reconhecedor sintático de configuração, onde relações estruturais são utilizadas, tais como relações entre atributos e/ou segmentos de bordas.

A fase de treinamento dos sistemas de visão baseados em modelos normalmente se dá através de treinamento supervisionado, onde objetos conhecidos são apresentados ao sistema, que extrai automaticamente os atributos e relações e armazena-os em modelos de cada objeto. Estes modelos também podem ser fornecidos pelo usuário. No entanto, o melhor seria que o sistema os extraísse automaticamente, para que se tenha uma garantia maior de que os mesmos procedimentos serão executados para determinar dados dos modelos e da imagem da cena, durante a fase de execução. O treinamento do sistema de visão deve ser realizado sob condições o mais próximo possível das condições de operação.

#### 4.4.7 - Aplicações de Visão Robótica

As aplicações de visão em robótica incluem: inspeção, classificação, navegação, reconhecimento e manipulação (BALLARD, BROWN, 1982; GROOVER et al., 1989; HORN, 1986; NALWA, 1993; RUSSEL, NORVIG, 1995; SHIRAI, 1987).

Inspeção constitui uma das aplicações mais comuns de visão computacional na indústria, sendo que muitas delas não envolvem robôs. Uma aplicação típica consiste na seleção (aceite ou rejeição) de peças produzidas na linha (GROOVER et al., 1989). O uso de visão computacional permite inspeção automática de toda as peças da linha (100% das peças), em alta velocidade, garantindo qualidade constante. Assim, diversos tipos de inspeções são realizadas, tais como inspeção de placas de circuito impresso (conferindo se furos, filetes, etc estão dentro das especificações), produtos agrícolas (inspeção de frutas, cereais, vegetais), peças (verificando se

estão dentro das especificações, com a furação correta), indústria farmacêutica (conferindo embalagens e rótulos, se pílulas estão nos lugares corretos das cartelas, etc.) e uma infinidade de outras aplicações.

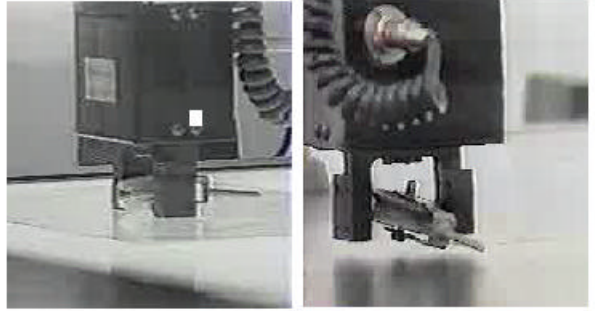
Tarefas de classificação têm por objetivo identificar e classificar objetos, e não inspecioná-los, isto é, os objetos não estão sendo aceitos ou recusados, mas sim associados a uma determinada classe ou rótulo. Assim, peças são separadas em categorias pré-determinadas (classes). A separação é realizada em função de sua identificação, baseada principalmente em sua forma ou em atributos específicos dos objetos de cada classe. Esta classificação normalmente é seguida por algum processo decisório e ações realizadas por robôs.

Navegação de robôs é uma aplicação que pode ser muito auxiliada por visão computacional. Os Veículos Autonomamente Guiados - AGVs (sigla do inglês "Autonomously Guided Vehicles") movem-se em um ambiente e precisam evitar colisões, descobrir onde estão os obstáculos, onde estão os caminhos livres disponíveis, desviar de obstáculos, localizar-se, calcular velocidade e orientação (sua própria ou de possíveis obstáculos móveis), entre outras. Normalmente, visão estereoscópica é utilizada nesta aplicação para fornecer informações de distância dos objetos ao AGV.

Em tarefas de montagem ou manufatura, peças a serem manipuladas podem chegar em posições pré-definidas em uma esteira transportadora em movimento ou numa bandeja. A visão seria usada para determinar a chegada da peça e, eventualmente, identificá-la, visando dirigir a ação de um manipulador que irá agarrar a peça e colocá-la na posição adequada. No entanto, para uma maior flexibilidade na aplicação, peças diferentes podem ser fornecidas às tarefas de montagem ou manufatura, sem restrição alguma quanto a seus posicionamentos. Assim, torna-se necessária a execução de um reconhecimento de objetos pelo sistema de visão.

O reconhecimento envolve não somente a identificação dos objetos na cena, mas também a determinação de suas respectivas posições e orientações. Tarefas de montagem e manufatura por robôs envolve manipulação de peças, incluindo ações tais como: segurar peças, inserir, encaixar, etc. Assim, o reconhecimento de objetos pode ser usado para adquirir informações necessárias ao robô, referentes à forma e posição das peças a serem manipuladas. Além disso, o sistema de visão pode fornecer informações de realimentação à tarefa de manipulação, tais como: está se aproximando, está desencaixando, etc.

A Figura 4.29 mostra a execução de uma tarefa de manipulação na célula de montagem da Escola Politécnica da Universidade de São Paulo. Esta célula é composta por dois robôs Mitsubishi Melfa ME-RD1, duas micro-câmaras SONY (colocadas nas garras dos manipuladores), uma câmara colorida e placas digitalizadoras de imagens, além de computadores e estações de trabalho. Para tarefas de manipulação, o sistema total (visão e controle do manipulador) são treinados para a geração de modelos internos das diferentes peças, em diferentes posições (para o reconhecimento visual) e também para a especificação da melhor posição de agarramento de cada peça em particular. Na fase de execução do sistema, a visão computacional reconhece cada uma das peças depositadas na área de trabalho, fornecendo as informações necessárias para direcionar a ação do manipulador (RILLO et al., 1992; REALI, 1994).



**Figura 4.29** – Montagem feita por manipuladores guiados por visão computacional, na célula de montagem da Escola Politécnica da Universidade de São Paulo.

## 4.5 - CONCLUSÃO

Este capítulo mostrou que os sensores possibilitam a extração de informações perceptuais relevantes para a execução de tarefas, permitindo a interação dos robôs no ambiente.

Existem diversos tipos de sensores, tanto internos quanto externos, incluindo sensores de posição, velocidade, aceleração, força, presença, distância, tato, visão. O uso de visão robótica tem sido muito difundido nos últimos anos, acrescentando autonomia aos robôs, além de flexibilização na sua atuação e nas tarefas a serem executadas, nas mais diversas aplicações.

A inserção de uma maior autonomia nos robôs permite que novas aplicações possam ser vislumbradas, mostrando grande potencial de exploração do uso de robôs em ambientes diferentes, desafiadores, distantes e hostis.

## 4.6 - REFERÊNCIAS

- BALLARD, D. H., BROWN, C. M., *Computer Vision*, Prentice-Hall, Inc., 1 ed., New York ?, 1982.
- DU BUF, J. M. H., KARDAN, M., SPANN, M., "Texture Feature Performance for Image Segmentation", *Pattern Recognition*; v. 23, editora ?, local ???, 1990.
- GONZALEZ, R. C., WOODS, R. E., *Digital Image Processing*, Addison Wesley, New York ?, 1992.
- GROOVER, M. P., WEISS, M., NAGEL, R.N., ODREY, N. G., *Robótica: tecnologia e programação*, Tradução de D. M. Savatovsky, Editora McGraw-Hill Ltda, São Paulo, 1989.
- HORN, B. K. P., *Robot Vision*, MIT Press, Massachusetts, 1986.
- MARR, D., *Vision*, Freeman, San Francisco, USA, 1982.
- NALWA, V. S. A., *Guided Tour of Computer Vision*, AT&T, 1993.
- REALI C. Rillo, Anna H., "RECTRI: um sistema de reconhecimento 3D baseado em uma hierarquia de estratégias", In: *Anais do 2º Simpósio Brasileiro de Automação Inteligente - 2o. SBAI*, Curitiba, pp.1-6, 1995.
- RILLO, M., RILLO, A. H. R. C., COSTA, L. A. R., "LSI Assembly cell", In: *Proceedings of 7<sup>th</sup> IFAC/IFIP/IFORS/IMACS/ISPE Symposium on Information Control Problems in Manufacturing Technology - INCOM 92*, May 25-28, 1992, Toronto, Canada, 1992.
- RUSSEL, S., NORVIG, P., *Artificial Intelligence: a modern approach*, Prentice-Hall Inc., London, 1995.
- SHIRAI, Y., *Three-Dimensional Computer Vision*, Springer-Verlag, Heidelberg, ed?, 1987.

TSAL, R. Y., "An efficient and accurate camera calibration technique for 3D machine vision", In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition - CVPR'86*, Florida, pp. 364-374, 1986.

WILKINSON, S., GUANIO, D., "Development of a Spectrometric Citrus Odor Sensor for Frugivorous Robot", In: *Proceedings of the IASTED International Conference on Robotics and Manufacturing*, p. 252-255, Banff, Canadá; 1998.