

**UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
CENTRO TECNOLÓGICO
DEPARTAMENTO DE ENGENHARIA MECÂNICA**

**TÁSSIO CARIELO MACEDO DE OLIVEIRA PINTO
VINICIUS MIRANDA**

**ALGORÍTMO DE REALIMENTAÇÃO VISUAL PARA APLICAÇÃO EM
CONTROLE ROBÓTICO**

**VITÓRIA
2013**

TÁSSIO CARIELO MACEDO DE OLIVEIRA PINTO
VINICIUS MIRANDA

**ALGORÍTMO DE REALIMENTAÇÃO VISUAL PARA APLICAÇÃO EM
CONTROLE ROBÓTICO**

Trabalho de Conclusão de Curso
apresentado ao Departamento de
Engenharia Mecânica do Centro
Tecnológico da Universidade Federal do
Espírito Santo, como requisito parcial
para obtenção do grau de Engenheiro
Mecânico.

Orientador: Prof. Dr. Antônio Bento Filho

VITÓRIA
2013

TÁSSIO CARIELO MACEDO DE OLIVEIRA PINTO
VINICIUS MIRANDA

**ALGORÍTMO DE REALIMENTAÇÃO VISUAL PARA APLICAÇÃO EM
CONTROLE ROBÓTICO**

Trabalho de Conclusão de Curso apresentado ao Departamento de Engenharia Mecânica do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do grau de Engenheiro Mecânico.

Aprovado em 02 de Maio de 2013.

COMISSÃO EXAMINADORA:

Prof. Dr. Antônio Bento Filho
Universidade Federal do Espírito Santo
Orientador

Prof. Dr. Carlos Friedrich Loeffler Neto
Universidade Federal do Espírito Santo
Examinador

Prof. Dr. Fernando César Meira Menandro
Universidade Federal do Espírito Santo
Examinador

*“O gênio é um por cento de inspiração e
noventa e nove por cento de suor”*

Thomas A. Edison

AGRADECIMENTOS

Agradecemos aos nossos pais, irmãos, amigos e mestres por todo apoio nestes anos de faculdade.

RESUMO

Cada vez mais robôs são aplicados na indústria para executar diversos tipos de tarefas que envolvem, na maioria das vezes, a identificação de objetos e de suas posições e orientações em relação a um referencial de trabalho. Dentro desse escopo, os sistemas de controle servo visual tem mostrado excelentes resultados. Hoje, há a necessidade de se desenvolver algoritmos que façam melhor uso dos recursos computacionais disponíveis e forneçam resultados de maior qualidade. Este projeto visa desenvolver um algoritmo para a obtenção do ângulo de rotação de uma válvula, através de ferramentas disponíveis na plataforma MATLAB®, com o objetivo de realimentar visualmente um manipulador robótico. Além disso, avaliar parâmetros que influenciam no resultado do algoritmo.

Palavras Chave: Controle Servo Visual. Visão Artificial. Características de imagem. Válvula. Algoritmo

ABSTRACT

Robots are increasingly used in industry to execute different kinds of tasks which mostly involve the identification of objects and their positions and orientations relative to a working reference. Within this scope, the visual servo control systems have shown excellent results. Today, there is the need to develop algorithms that make better use of available computational resources and provide higher quality results. This project aims to develop an algorithm for obtaining the angle of rotation of a valve through tools available in MATLAB® platform, in order to visually feedback a robotic manipulator. Moreover, evaluate parameters that influence the results of the algorithm.

Keywords: Visual Servo Control. Artificial Vision. Image feature. Valve. Algorithm.

LISTA DE FIGURAS

Figura 1: Esquema básico para um Sistema de Visão Artificial	16
Figura 2: Imagem remapeada. Fonte	18
Figura 3: Aplicação de redução de ruído.....	18
Figura 4: Aumento de contraste	19
Figura 5: Imagem binarizada por <i>threshold</i>	20
Figura 6: Mean Shift. Imagem (a) original. Imagem (b) com efeito <i>mean-shift</i> aplicado	21
Figura 7: Exemplo de bordas	21
Figura 8: Crescimento de regiões. Em (a) pixels adjacentes com alguma característica comum. Em (b) início do crescimento. (c) Estado intermediário do crescimento (d) Estado completo	22
Figura 9: Estrutura de controle servo visual direta baseada em posição (PBVS).....	25
Figura 10: Estrutura de controle servo visual direta baseada em imagem (IBVS)	25
Figura 11: Estrutura dinâmica de controle servo visual indireta baseada em posição	25
Figura 12: Estrutura dinâmica de controle servo visual indireta baseada em imagem	26
Figura 13: Configuração quanto ao sistema de câmera	28
Figura 14: Resultado Edge Detection.....	30
Figura 15: Resultado Corner Detection Fonte	31
Figura 16: Resultado de detecção de regiões	32
Figura 17: Exemplo características do tipo Ridges	33
Figura 18: Sistema de visão artificial utilizado neste trabalho	35
Figura 19: Válvula do CT-3.....	36
Figura 20: Padrão de imagem para posicionamento de câmera	38
Figura 21: Pátio do CT-3.	39
Figura 22: Aquisição em (a) fundo complexo e em (b) fundo branco.	40
Figura 23: Pseudocódigo de extração e correlação de características em um quadro capturado.....	42
Figura 24 - Vetores u e v obtidos e ângulo entre eles.	42
Figura 25: Pseudocódigo de interpretação dos pontos características para obtenção do ângulo de rotação da válvula.	43

Figura 26: Pseudocódigo do rastreador de pontos característicos.....	45
Figura 27: Pontos correlacionados no teste 1.....	48
Figura 28: <i>Inliers</i> encontrados no teste 6.....	50

LISTA DE TABELAS

Tabela 1: Detectores de características e suas classificações	34
Tabela 2: Especificação da câmera Sony DSC-W510	37
Tabela 3: Especificação do computador XPS15 L502x utilizado no processamento das imagens	38
Tabela 4: Testes executados	46
Tabela 5: Quantidade de pontos encontrados na extração e correlação de características.....	47
Tabela 6: Ângulos finais calculados	49
Tabela 7: Resultados encontrados pelo algoritmo	49
Tabela 8: Ângulos de rotação da válvula aferidos automaticamente, manualmente e erro percentual	51
Tabela 9: Pontos correlacionados	59
Tabela 10: Pontos <i>Inliers</i>	62
Tabela 11: Pontos rastreados	65
Tabela 12: Pontos característicos aferidos manualmente no primeiro e último quadros capturados no teste	68

SUMÁRIO

1 INTRODUÇÃO	12
1.1 HISTÓRIA DO CONTROLE SERVO VISUAL	12
1.2 SISTEMAS DE CONTROLE SERVO VISUAL	13
1.3 TRABALHOS ANTERIORES	14
1.4 OBJETIVOS	15
2 CONCEITO DE CONTROLE SERVO VISUAL	16
2.1 SISTEMA DE VISÃO ARTIFICIAL	16
2.1.1 Aquisição	17
2.1.2 Pré-processamento	17
2.1.3 Segmentação	19
2.1.4 Extração de características	23
2.1.5 Reconhecimento e interpretação	23
2.1.6 Base de dados	23
2.2 TÉCNICAS DE CONTROLE DE SERVO VISUAL	24
2.2.1 Quanto ao arranjo hierárquico	26
2.2.2 Quanto à estratégia de controle	27
2.2.3 Quanto a configuração do sistema de câmeras	27
3 CARACTERISTICAS DE IMAGENS.....	29
3.1 CARACTERISTICA	29
3.2 TIPOS DE CARACTERÍSTICAS	30
3.2.1 Borda (Edges).....	30
3.2.2 Cantos (Corners) / Pontos de interesse	31
3.2.3 Regiões de interesse (Blobs)	32
3.2.4 Ridges	33
3.3 TIPOS DE DETECTORES DE CARACTERÍSTICAS.....	33
3.4 DESCRIÇÃO DE CARACTERISTICAS.....	34
4 SISTEMA DE VISÃO ARTIFICIAL.....	35
4.1 PROBLEMA E RESULTADOS.....	35

4.2 AQUISIÇÃO	36
4.2.1 Dispositivo e imagem.....	36
4.2.2 Posicionamento do dispositivo	38
4.2.3 Iluminação.....	39
4.2.4 Resolução	40
4.2.5 Fundo	40
4.2.6 Calibração de câmera	41
4.3 EXTRAÇÃO E CORRELAÇÃO DE CARACTERÍSTICAS.....	41
4.4 INTERPRETAÇÃO.....	42
4.5 BASE DE CONHECIMENTO	44
4.6 RASTREAMENTO	44
 5 RESULTADOS.....	 46
5.1 TESTES EXECUTADOS.....	46
5.2 EXTRAÇÃO E CORRELAÇÃO DE CARACTERÍSTICAS.....	47
5.3 INTERPRETAÇÃO.....	48
5.4 MEDIDA DE ERRO	50
 6 CONCLUSÃO	 52
 7 REFERENCIAS.....	 53
 APÊNDICE A - ALGORITMO	 57
 APÊNDICE B - PONTOS OBTIDOS NA EXTRAÇÃO E CORRELAÇÃO DE CARACTERISTICAS	 59
 APÊNDICE C - PONTOS CARACTERÍSTICOS AFERIDOS MANUALMENTE.....	 68

1 INTRODUÇÃO

Cada vez mais robôs são aplicados na indústria para executar diversos tipos de tarefas que envolvem, na maioria das vezes, a identificação de objetos e de suas posições e orientações em relação a um referencial de trabalho. Dentro desse escopo, os sistemas de controle servo visual tem mostrado excelentes resultados.

Com o aumento da velocidade de processamento e miniaturização de computadores nas últimas décadas viabilizou-se o uso desses sistemas para uma gama muito maior de aplicações. Porém, hoje, há a necessidade de se desenvolver algoritmos que façam melhor uso desses recursos computacionais disponíveis e forneçam resultados de maior qualidade.

Este capítulo fornece informações sobre as dificuldades enfrentadas pelos atuais robôs industriais e como o uso da realimentação através dos sensores visuais pode melhorar o desempenho desses robôs. Também discute trabalhos anteriores feitos por outros pesquisadores no campo da servo visão e define os objetivos e o escopo deste projeto.

1.1 HISTÓRIA DO CONTROLE SERVO VISUAL

O uso de Câmeras de vídeo para realimentação de sinais em sistemas de controle remonta aproximadamente da década de 70. O termo servo controle visual foi cunhado por Hill e Park (1979). Na ocasião, o conceito de robô começou a ser aceito como uma máquina suscetível de interagir com seu ambiente com uma certa autonomia. Entre os sensores disponíveis no momento, a câmera tem fascinado pesquisadores, devido à grande quantidade de informações que podem ser extraídas a partir das imagens. Prover um robô com informações visuais aproxima suas habilidades as humanas.

Devido à quantidade de informações fornecidas pelas câmeras, a ambição inicial na área de robótica foi em geral de entender o mundo, a fim de decidir, por meio de abordagens simbólicas, como agir. A ideia de utilizar câmeras dentro de uma malha de controle só veio mais tarde. Isto foi alcançado selecionando “sinais” específicos de uma imagem e utilizando como um sensor de realimentação associado a um sistema dinâmico.

Essas duas abordagens se desenvolveram separadamente nos anos 80. O primeiro ligou-se ao campo da visão computacional, liderado principalmente por cientistas da computação e matemáticos aplicados. O segundo deu origem à abordagem de controle servo visual, estudada por pesquisadores da área de automação e controle (CORKE, 1996).

1.2 SISTEMAS DE CONTROLE SERVO VISUAL

O termo Controle Servo Visual refere-se ao uso em malha fechada, de informações provenientes de visão computacional para determinar a ação de controle de um robô ou manipulador robótico para executar uma tarefa (BERNARDES, 2009). Tarefa esta que pode ser a aproximação em relação a um objeto alvo na cena em questão, ou, a aproximação do robô em relação a um conjunto de características visuais extraída da imagem. Esta Técnica de controle baseia-se em outras importantes áreas do conhecimento tais como: processamento de imagem, visão computacional, teoria de controle, cinemática, dinâmica e visão computacional (CHAUMETTE; HUTCHINSON, 2006).

Diversos tipos de sensores têm seu uso consagrado em aplicações de robótica, mas nenhum deles é capaz de proporcionar uma descrição tão rica do ambiente de trabalho quanto os sensores visuais. O rápido crescimento do poder computacional e a redução do custo de câmeras e placas de captura de quadros criaram um crescente interesse na aplicação desses sistemas na indústria.

1.3 TRABALHOS ANTERIORES

Um das primeiras referências é de Shirai e Inoue (1973), que descrevem a forma como um ciclo de realimentação visual pode ser usado para corrigir a posição de um robô afim de aumentar a precisão em uma tarefa. Em seu trabalho é descrito um sistema que permite ao robô agarrar um prisma quadrado e colocá-lo em uma caixa utilizando controle servo visual. Foram utilizadas extração de bordas e ajuste de linhas para determinar a posição e orientação da caixa, usava uma câmera fixa e o tempo do ciclo observado foi de 10 segundos.

Hill e Park (1979) descrevem o controle servo visual de um robô utilizando o processamento de imagens binárias para ganhos de velocidade e confiabilidade, provendo a posição planar, assim como uma estimativa simples de profundidade baseada na distância aparente entre características conhecidas. Experimentos também foram conduzidos usando uma faixa de luz projetada para determinação de profundidade mais robustamente, assim como orientação de superfície. Estes experimentos demonstraram movimentos visualmente orientados, tanto planares quanto 3D, assim como rastreamento e agarramento de objetos em movimento.

Máquinas visualmente guiadas tem sido construídas para emular as habilidades humanas no ping pong (ANDERSSON, 1987)(FÄSSLER; BEYER; WEN, 1990), balanceamento de pêndulo invertido (DICKMANNS; GRAEFE, 1988), e apanhamento de objetos (BUKOWSKI *et al.*, 1991)(BUTTAZZO; ALLOTTA; FANIZZA, 1993)(SAKAGUCHI *et al.*, 1993). Existem vários trabalhos no uso de servo visão para agarrar alvos móveis. O primeiro trabalho parece ter sido feito no *Stanford Research Institute* (SRI) na década de setenta (ROSEN, 1976). Zhang *et al.* (1990) apresentaram um controlador servo visual de rastreamento para que um robô pegasse itens de uma correia transportadora em movimento rápido (300mm/s). Harrel *et al.* (1989) descreve um robô hidráulico para colher frutas.

O uso de servo controle visual em um ambiente de tele robótica tem sido discutido por Yuan *et al.* (1990), Papanikolopoulos *et al.* (1992) e Tendick *et al.*(1991). Servo visão

permite que a tarefa a ser executada seja especificada por um operador humano em termos da seleção de características visuais e configurações desejadas.

Estudos baseados em redes neurais (HASHIMOTO *et al.*, 1989); (KUPERSTEIN, 1988); (MEL, 1990) e algoritmos de aprendizagem de máquinas (MILLER III, 1987) também tem sido utilizados para realizar a coordenação de robôs. Uma câmera fixa observa o objeto e o robô no espaço de trabalho e consegue fazer a relação entre os ângulos de junta e a pose (posição e orientação) 3D do efetuador. Tais sistemas requerem treinamento, mas eliminam a necessidade de relações analíticas complexas entre os recursos de imagem e ângulos de junta.

1.4 OBJETIVOS

Este trabalho tem por objetivo apresentar os princípios básicos de visão computacional que poderão ser utilizados para trabalhos futuros e desenvolver um algoritmo para a obtenção do ângulo de rotação de uma válvula, através de ferramentas disponíveis na plataforma MATLAB®, com o objetivo de realimentar visualmente um manipulador robótico.

2 CONCEITO DE CONTROLE SERVO VISUAL

A função de um sistema de posicionamento robótico baseado em visão é o controle da pose do efetuador usando informações visuais extraídas a partir da imagem captada pela câmera (CHONG, 2004).

2.1 SISTEMA DE VISÃO ARTIFICIAL

A organização de um sistema de visão robótica é dependente da aplicação. Um esquema básico de Sistema de Visão Artificial (SVA) é proposto por (MARQUES FILHO; VIEIRA NETO, 1999) e é apresentado na figura abaixo.

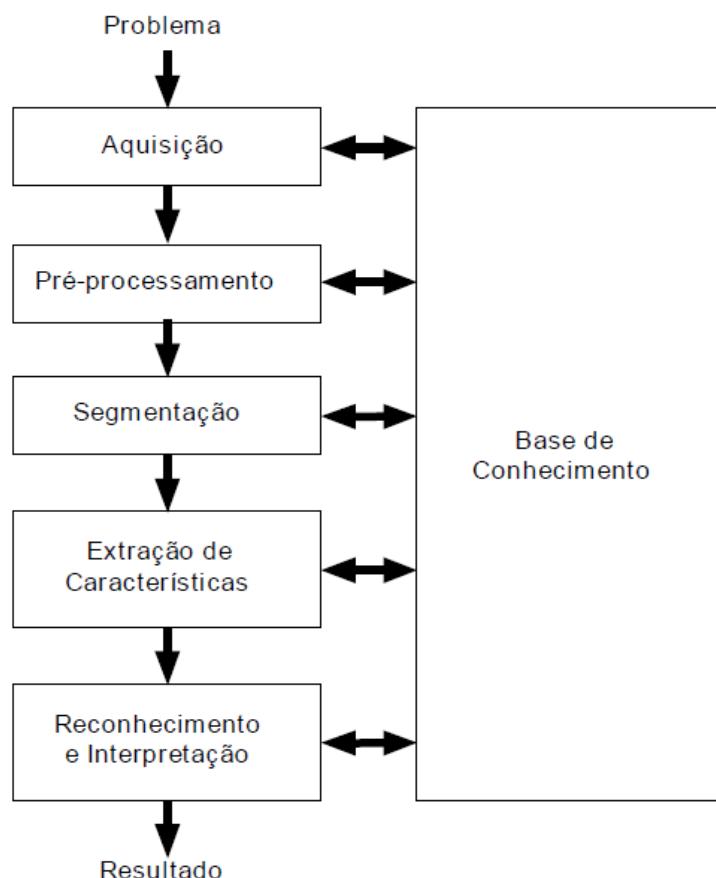


Figura 1: Esquema básico para um Sistema de Visão Artificial. Fonte: (MARQUES FILHO; VIEIRA NETO, 1999)

2.1.1 Aquisição

É a primeira etapa em um sistema de visão artificial e consiste na captação de uma imagem real e a transformação em uma imagem digital. Para tanto são necessários dois dispositivos básicos. O primeiro diz respeito ao dispositivo físico sensível a uma faixa do espectro eletromagnético luminoso como, por exemplo, raio X, ultravioleta, infravermelho ou a luz visível, capaz de converter os valores captados em sinais elétricos proporcionais. O segundo são os digitalizadores, que são dispositivos capazes de converter sinais analógicos de entrada como os gerados pelos sensores luminosos acima descritos, em sinais digitais reconhecidos pelo sistema computacional. Esta etapa produz na saída uma imagem digitalizada.

Dentre os aspectos de projeto envolvidos nesta etapa, pode-se mencionar: a escolha do tipo de sensor, o conjunto de lentes a utilizar, as condições de iluminação da cena, a calibração da câmera, os requisitos de velocidade de aquisição, a resolução e o número de níveis de cinza da imagem digitalizada, dentre outros (MARQUES FILHO; VIEIRA NETO, 1999).

2.1.2 Pré-processamento

A função da etapa de pré-processamento é aprimorar a qualidade da imagem para as etapas subsequentes (MARQUES FILHO; VIEIRA NETO, 1999). A imagem resultante desta etapa é uma imagem digitalizada de melhor qualidade que a original. Entre as tarefas mais comuns executadas nessa etapa podemos citar:

Remapeamento: técnica matemática utilizada para criar uma nova versão da imagem com largura e/ou altura diferentes em pixels. É utilizada para assegurar que o sistema de coordenadas está correto.

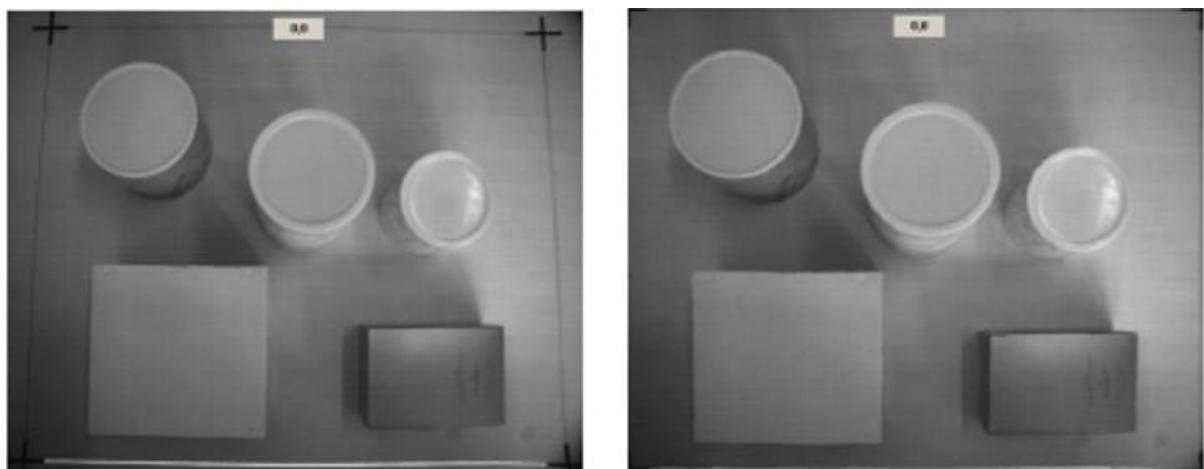


Figura 2: Imagem remapeada. Fonte: (ORLANDI, 2008).

Redução de ruídos: os ruídos podem aparecer de diversas fontes, como por exemplo, o tipo de sensor utilizado, a iluminação do ambiente, as condições climáticas no momento da aquisição da imagem ou a posição relativa entre o objeto de interesse e a câmera. A redução de ruídos é feita pela utilização de filtros, que podem ser espaciais (filtros que atuam diretamente na imagem) ou de frequência (onde a imagem é transformada para o domínio de frequência usando a transformada de Fourier, filtrada neste domínio e, em seguida transformada de volta para o domínio de espaço).

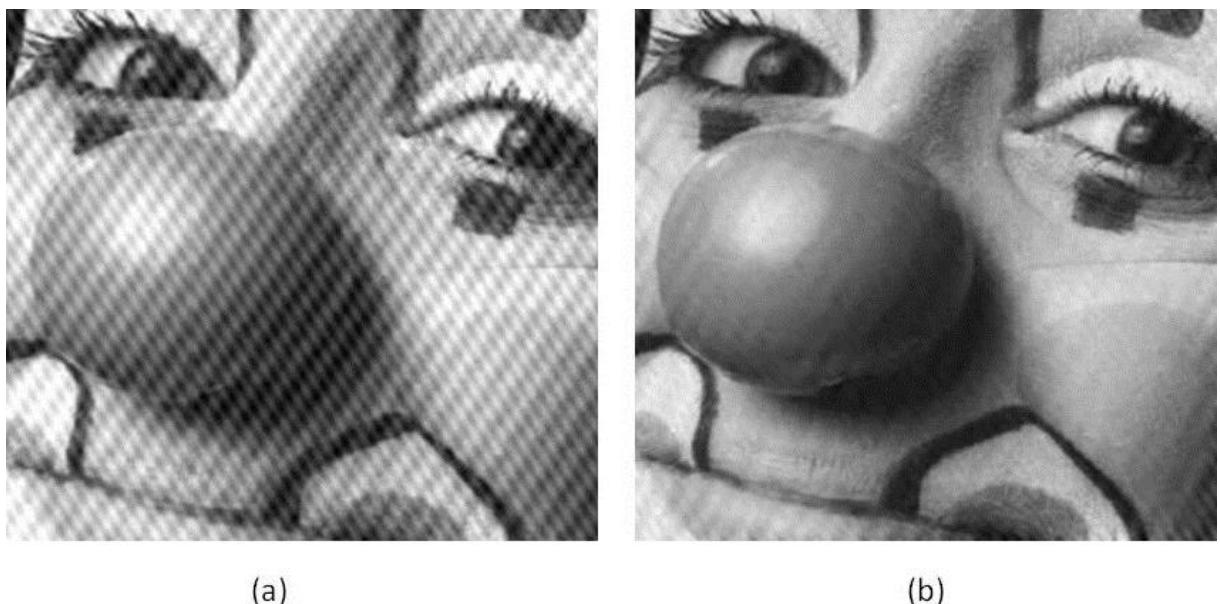


Figura 3: Aplicação de redução de ruído. Fonte: (MARENGONI; STRINGHINI, 2010)

Aumento de contraste: tem por objetivo melhorar a qualidade das imagens sob os critérios subjetivos do olho humano. Esta técnica consiste numa transformação do valor de cada pixel, em escala de cinza, com o objetivo de aumentar a discriminação visual entre os objetos presentes na imagem. Realiza-se a operação ponto a ponto, independentemente da vizinhança.

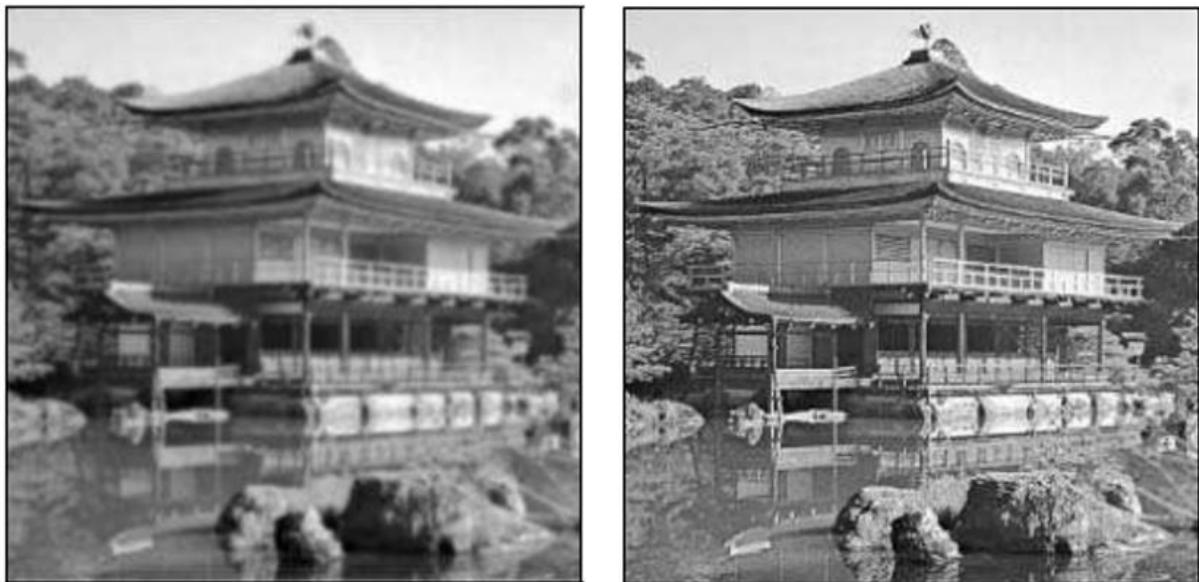


Figura 4: Aumento de contraste. Fonte: (MARQUES FILHO; VIEIRA NETO, 1999)

2.1.3 Segmentação

O processo de segmentação consiste em partitionar uma imagem em regiões, ou objetos distintos. Este processo é geralmente guiado por características do objeto ou região, como por exemplo, cor ou proximidade. O nível de detalhamento em um processo de segmentação depende da tarefa a ser executada e da resolução da imagem que se tem (MARENCONI; STRINGHINI, 2010). Nesta etapa são produzidas imagens digitais segmentadas.

A segmentação é uma das etapas mais difíceis na implantação de um sistema de visão artificial. Neste assunto não há unanimidade sobre o que é melhor ou pior. Algumas técnicas bastante utilizadas que obtém sucesso dependendo do tipo de imagem e de objeto são:

Técnicas baseadas em cores, ou intensidade do valor de pixel: São simples de serem implementadas, rápidas em termos computacionais e utilizam de propriedades intuitivas para criar a imagem segmentada. A ideia central deste tipo de operador é a de verificar no histograma da imagem quantas regiões existem (picos e vales) e segmentar a imagem baseado nesta informação. A técnica mais comum é o *Threshold*, Figura 5. Esta técnica divide o histograma da imagem em duas partes ou mais (chamado *threshold* múltiplo). Outra técnica utilizada é o *Mean-shift*, Figura 6, que usa toda a informação de cor numa imagem para obter grupos de pixels de cores semelhantes e separar regiões homogêneas, esperando que estes grupos formem objetos.

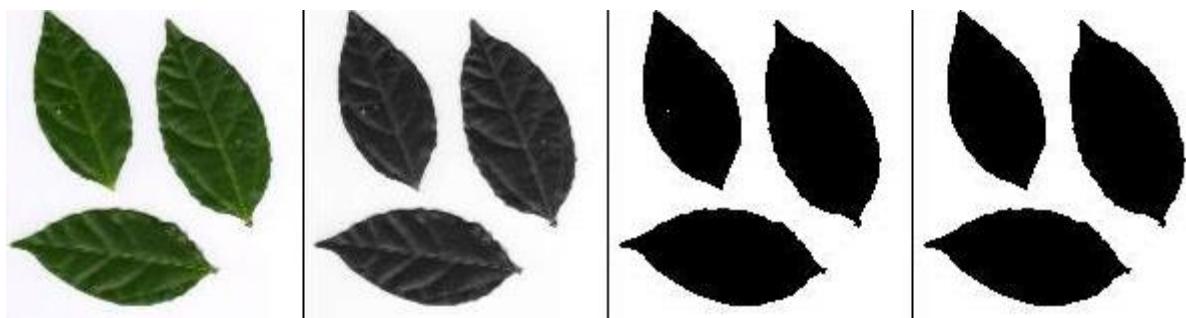


Figura 5: Imagem binarizada por *threshold* Fonte: (BAUERMANN, 2009)

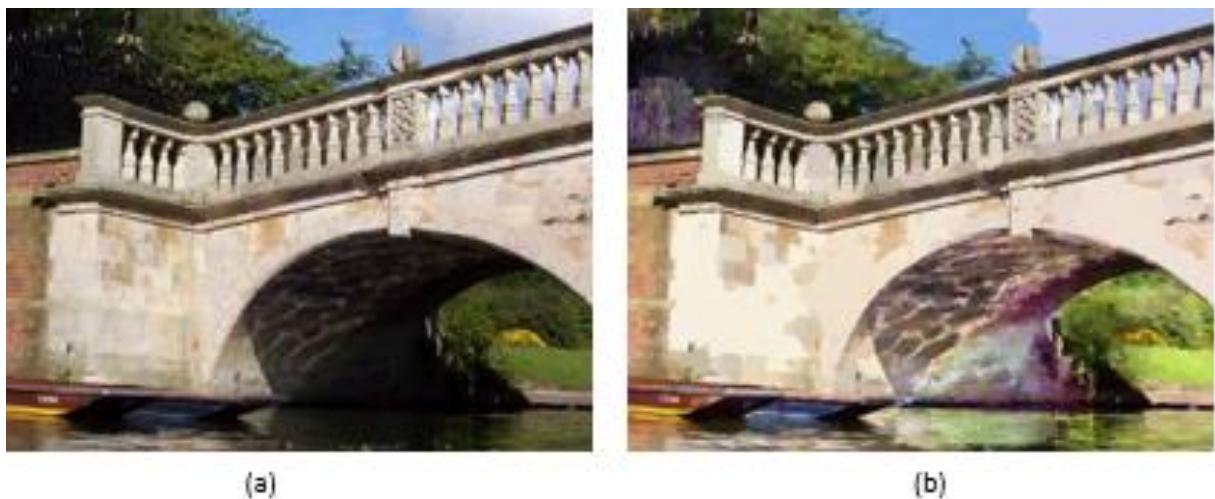


Figura 6: Mean Shift. Imagem (a) original. Imagem (b) com efeito *mean-shift* aplicado. Fonte: (BAUERMANN, 2009)

Reconhecimento de bordas: Uma borda em uma imagem é caracterizada por uma mudança, normalmente abrupta, no nível de intensidade dos pixels. Os detectores de borda são definidos para encontrar este tipo de variação nos pixels e quando estes pixels estão próximos eles podem ser conectados formando uma borda ou um contorno e assim definindo uma região ou objeto. Para encontrar as bordas são utilizados diversos tipos de filtros. Alguns dos filtros mais usados são *Sobel*, *Prewit* e *Canny*.

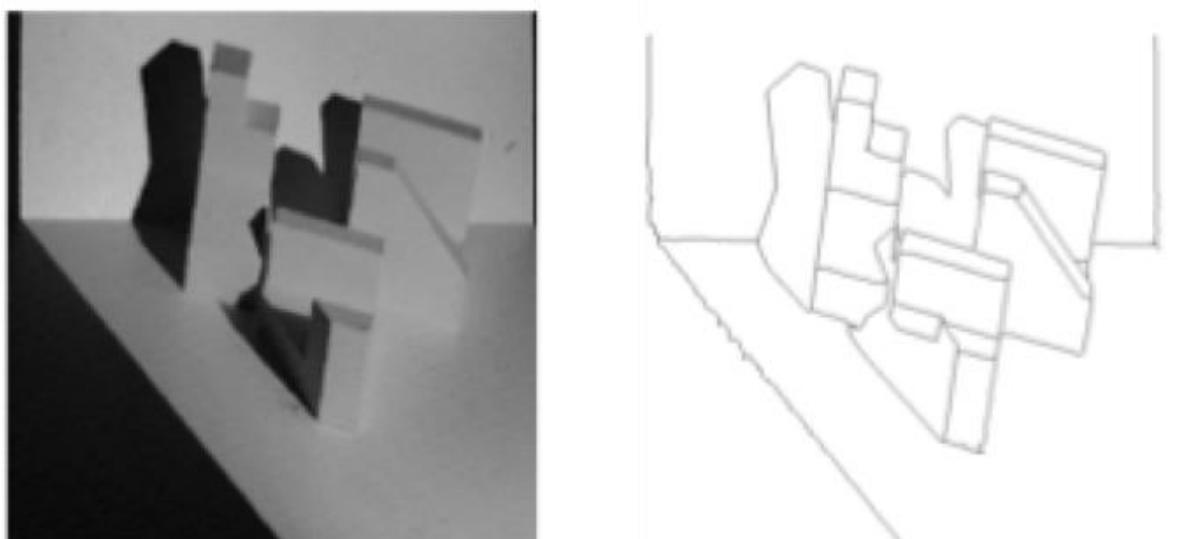


Figura 7: Exemplo de bordas Fonte: (MELO, 2011)

Crescimento de regiões: Este método assume que uma região é formada por pixels adjacentes e com algumas características estatísticas em comum, como média e desvio-padrão das intensidades (cores) dos pixels. O procedimento parte de um conjunto de pontos, chamados de sementes, e, a partir destes pontos vai agrupando outros pontos utilizando uma vizinhança de influência, formando as regiões Figura 8. Detalhes importantes desta técnica são a definição das sementes, que são pontos por onde o crescimento deve começar, e a definição de um critério de parada para o crescimento de regiões.

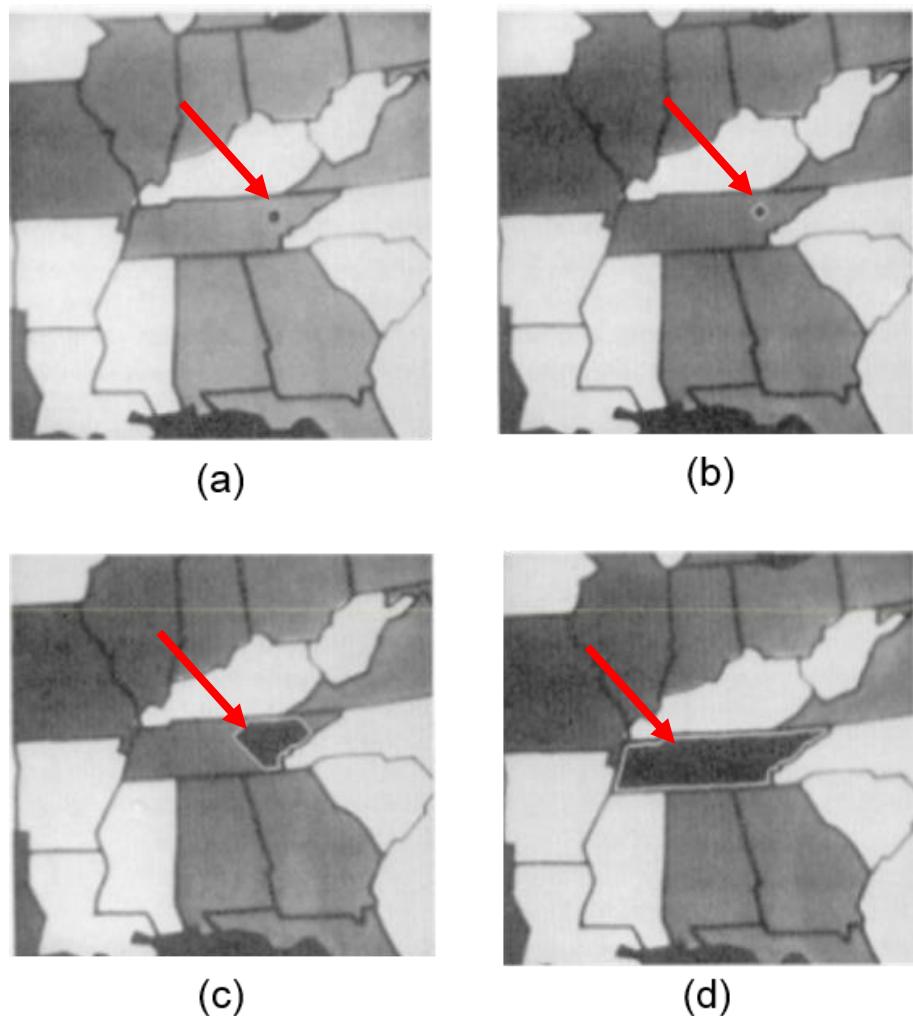


Figura 8: Crescimento de regiões. Em (a) pixels adjacentes com alguma característica comum. Em (b) início do crescimento. (c) Estado intermediário do crescimento (d) Estado completo. Fonte:(MELO, 2011)

2.1.4 Extração de características

As características são extraídas da imagem através de algoritmos. Dependendo da característica selecionada, o algoritmo pode detectar o centroide, área, perímetro, etc. É importante observar que nesta etapa a entrada ainda é uma imagem, mas a saída é um conjunto de dados correspondentes àquela imagem.

2.1.5 Reconhecimento e interpretação

Denominamos reconhecimento o processo de atribuição de um rótulo a um objeto baseado em suas características, traduzidas por seus descritores. A tarefa de interpretação, por outro lado, consiste em atribuir significado a um conjunto de objetos reconhecidos.

Da extração de características a pose relativa do objeto pode ser computada. Esta pose deve ser computada no plano da imagem e mais tarde transformada para um sistema global usando uma matriz de transformação adequada.

2.1.6 Base de dados

Todas as tarefas das etapas descritas acima pressupõem a existência de um conhecimento prévio sobre o problema a ser resolvido armazenado em uma base de conhecimento. Idealmente, esta base de conhecimento deveria não somente guiar o funcionamento de cada etapa, mas também permitir a realimentação entre elas.

Esta integração entre as várias etapas através da base de conhecimento ainda é um objetivo difícil de alcançar e não está presente na maioria dos Sistemas de Visão Artificial (SVAs) existentes atualmente.

2.2 TÉCNICAS DE CONTROLE SERVO VISUAL

O objetivo do controlador servo visual é minimizar um erro $e(t)$ tipicamente definido como:

$$e(t) = s(m(t), a) - s^*(t) \quad (1)$$

Onde $s^*(t)$ é o vetor de características da imagem desejado e $s(m(t), a)$ é o vetor de características da imagem atual, computadas a partir de um conjunto $m(t)$ de pixels extraídos da imagem e de a , que representa as informações conhecidas do sistema como, por exemplo, parâmetros intrínsecos da câmera ou o modelo 3D do objeto (CHAUMETTE; HUTCHINSON, 2006). Os vetores s^* e s são designados de acordo com a estratégia de controle adotada, podendo ser parâmetros 3D estimados a partir de modelos computacionais de objetos, ou 2D extraídos diretamente da imagem.

Sanderson e Weiss (1980) introduziram uma importante classificação das estruturas servo visuais, apresentada nas figuras Figura 9 à Figura 12.

As técnicas de controle servo visual podem ser classificadas quanto ao arranjo hierárquico do controlador, quanto à estratégia de controle e quanto à configuração do sistema de câmeras (BERNARDES, 2009).

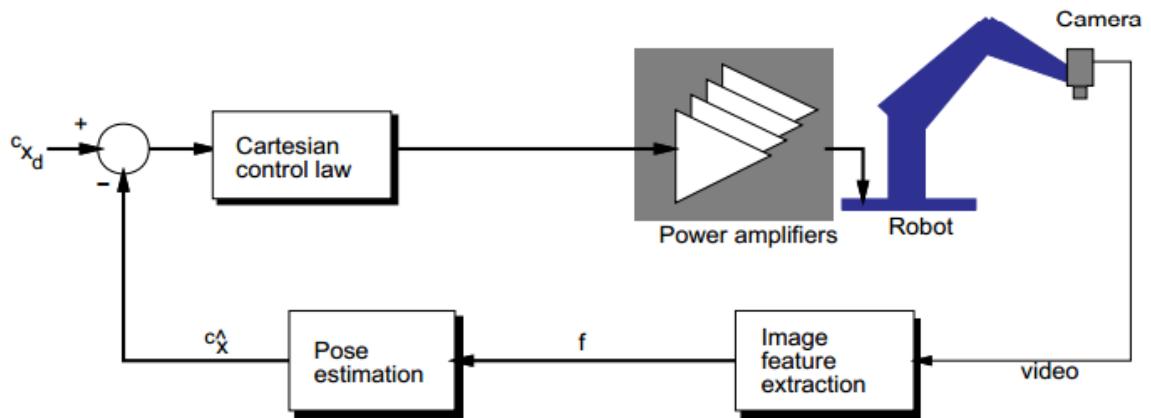


Figura 9: Estrutura de controle servo visual direta baseada em posição (PBVS). Fonte: (CORKE, 1996)

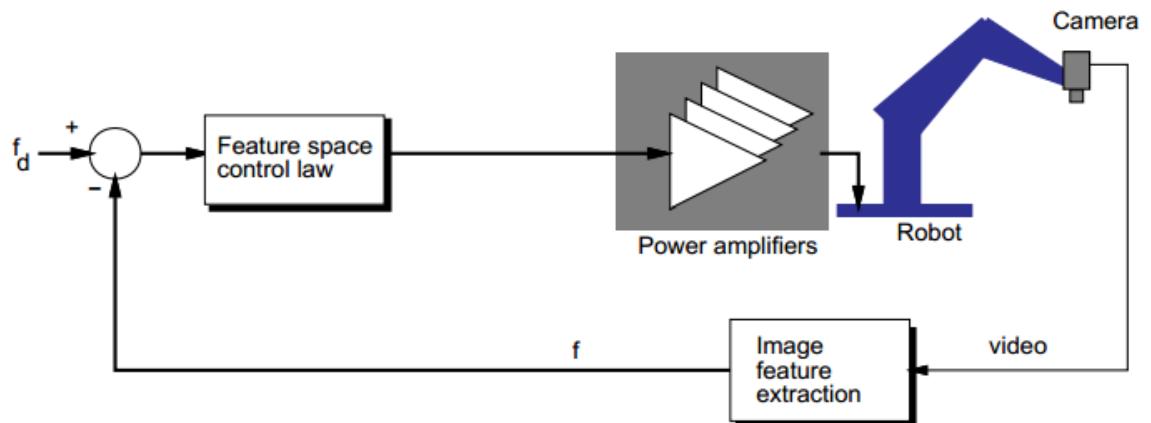


Figura 10: Estrutura de controle servo visual direta baseada em imagem (IBVS). Fonte(CORKE, 1996)

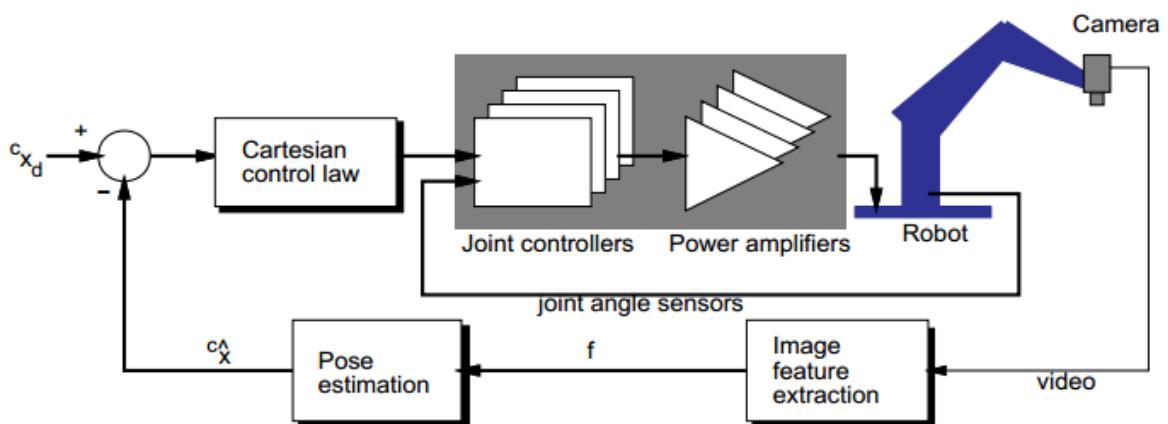


Figura 11: Estrutura dinâmica de controle servo visual indireta baseada em posição. Fonte: (CORKE, 1996).

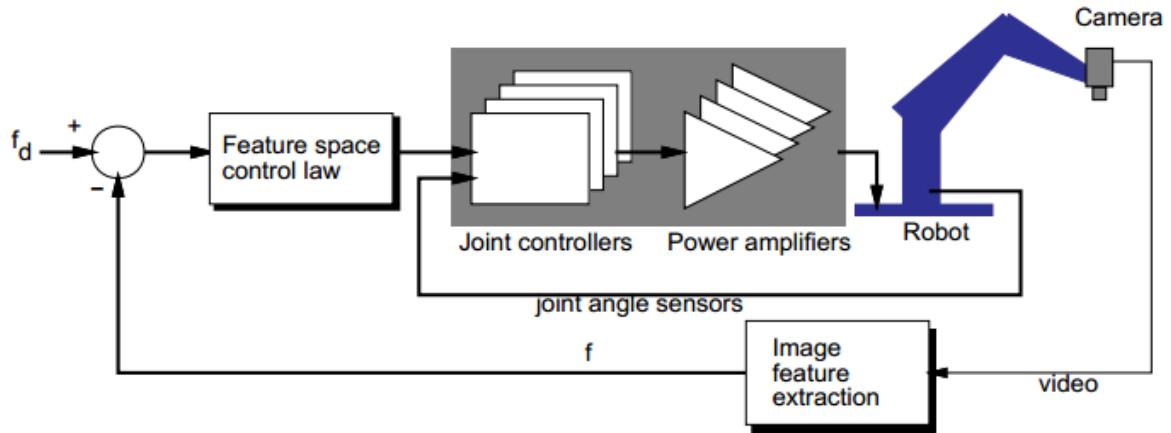


Figura 12: Estrutura dinâmica de controle servo visual indireta baseada em imagem. Fonte: (CORKE, 1996).

2.2.1 Quanto ao arranjo hierárquico

As duas possíveis configurações hierárquicas são direta e indireta (*Look and Move*).

Controle servo visual direto: usa as informações do sistema visual para controlar diretamente a estabilidade dos robôs. Esta configuração direta apresenta um enorme inconveniente em sistemas com dinâmica não linear devido à baixa taxa de amostragem dos sistemas de visão computacional, o que pode gerar grande instabilidade no sistema. Figura 9 e Figura 10.

Controle servo visual indireto: usa uma realimentação de junta. Esta realimentação adicional garante uma maior precisão do posicionamento do robô, entretanto, ocorre uma sobrecarga de processamento. Figura 11 e Figura 12.

2.2.2 Quanto à estratégia de controle

Basicamente se destacam duas metodologias:

Baseado em posição: as características extraídas da imagem são usadas em conjunto com um modelo geométrico do objeto a fim de determinar a pose do objeto em relação à câmera. Assim, essa estratégia toma como referência a ser seguida a pose ou modelo 3D final desejado. É importante que os parâmetros do modelo sejam bem definidos, pois um pequeno erro na mensuração da imagem pode implicar em grandes diferenças na posição estimada (CHAUMETTE; HUTCHINSON, 2006). Figura 9 e Figura 11.

Baseado em imagem: a determinação da pose é eliminada e o controle é realizado diretamente a partir das características extraídas das imagens. Isto é feito minimizando a diferença entre os parâmetros extraídos da imagem atual e de uma imagem final de referência. Essa estratégia apresenta menor custo computacional, além de ser robusta aos erros da modelagem e calibração da câmera. Por outro lado, pode implicar em maior dificuldade de estabilização em um problema de controle potencialmente complexo e não linear devido à ausência de um modelo tridimensional do objeto (CHAUMETTE; HUTCHINSON, 2006)(BERNARDES, 2009) Figura 10 e Figura 12.

2.2.3 Quanto à configuração do sistema de câmeras

O sistema pode ser Monocular (uma câmera), Binocular ou Estéreo (duas Câmeras) ou Redundante (mais de duas câmeras). Além disso a câmera pode estar fixa em um ponto ou embarcada no robô, não sendo os dois sistemas excludentes. Quando a câmera está embarcada é preciso descrever como o movimento do robô afeta a posição da câmera em relação ao alvo. Este sistema possui uma visão mais precisa da cena, principalmente para situações de aproximação, como *docking*, por exemplo.

O sistema com câmera fixa possui uma visão mais global, porém, com menor precisão. Na figura abaixo tem-se um resumo dessas configurações.

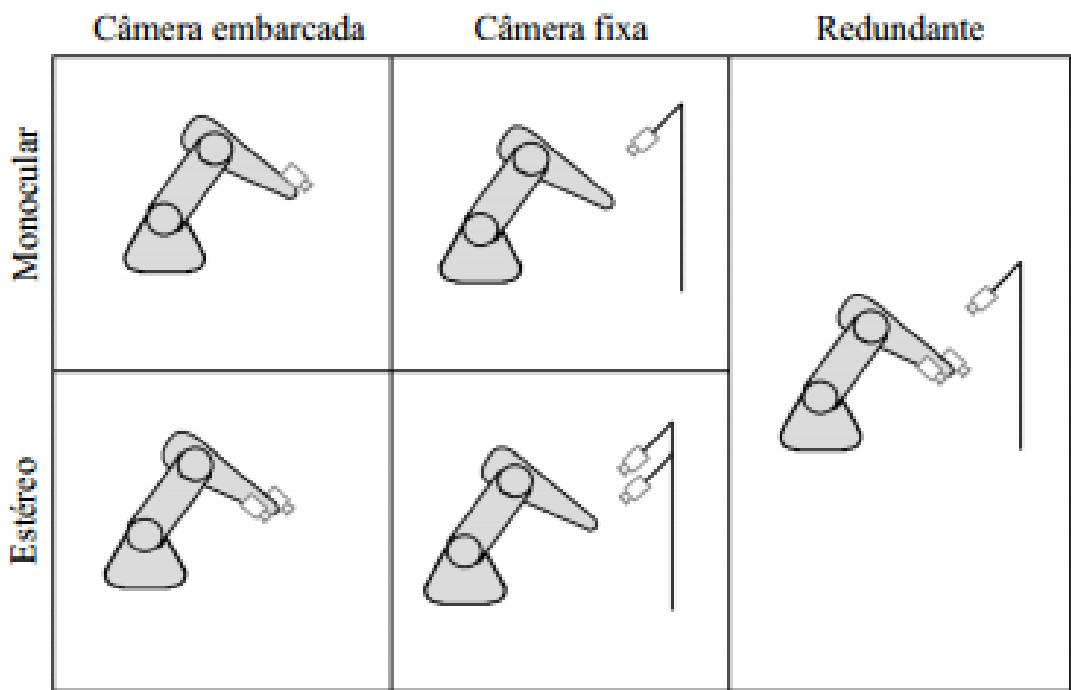


Figura 13: Configuração quanto ao sistema de câmera. Fonte: (MUÑOZ, 2011)

3 CARACTERISTICAS DE IMAGENS

Após a aquisição da imagem é necessário utilizar técnicas para identificar e descrever as características contidas no quadro.

3.1 CARACTERISTICA

Uma característica de imagem é definida geralmente como qualquer relação mensurável em uma imagem (CORKE, 1996). São exemplos de características: furos, bordas, cantos, contornos e áreas.

As características locais são padrões da imagem que se diferenciam da região imediatamente vizinha. Elas são associadas a mudanças de uma ou mais propriedades da imagem, como intensidade, cor e textura (TUYTELAARS; MIKOLAJCZYK, 2008). As informações são retiradas de uma região em torno de uma característica local e convertidas em descritores. O conjunto de descritores forma uma representação da imagem que permite a identificação de objetos nela contidos.

Porém, na fase de interpretação das características podem haver variações, como por exemplo no ponto de vista, escala ou rotação. Portanto, as regiões descritas pelos pontos devem ser covariantes.

Regiões covariantes, segundo Sivic e Zisserman (2003), são regiões que devem corresponder às mesmas características da imagem original sob pontos de vista diferentes, ou seja, sua forma não é fixa, mas adapta-se automaticamente, de modo que as proporções aproximadas da imagem original sejam mantidas.

Muitos dos algoritmos de visão computacional utilizam a detecção de uma característica como um passo inicial, desta forma, um grande número de detectores de características tem sido desenvolvidos. Estes variam muito em função do tipo de

característica, complexidade computacional e a repetitividade. A seguir são explicados cada tipo de característica.

3.2 TIPOS DE CARACTERÍSTICAS

Alguns tipos mais comuns de características covariantes são sumarizadas neste tópico.

3.2.1 Borda (*Edges*)

Bordas são pontos onde se encontra uma fronteira entre duas regiões na imagem, Figura 14. Em geral uma borda pode ser de uma forma arbitrária e talvez inclua junções. Na prática, uma borda é normalmente definida como um conjunto de pontos na imagem. Normalmente os algoritmos costumam colocar alguma restrição nas propriedades da borda, como uma forma, suavidade e gradiente de valor afim de aprimorar o resultado final. Localmente bordas possuem uma estrutura unidimensional.

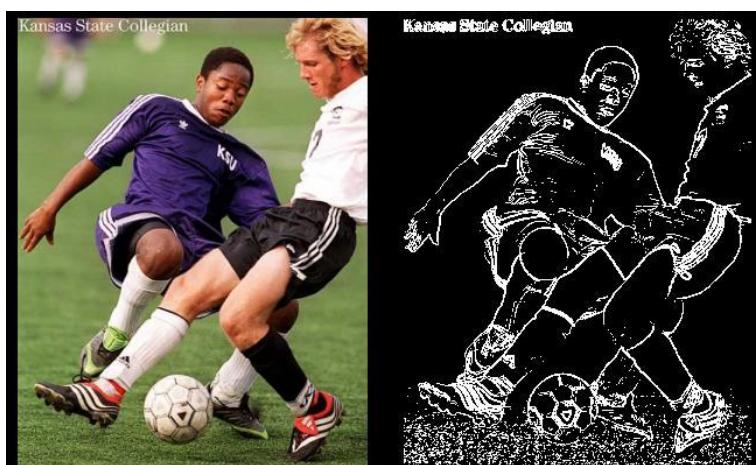


Figura 14: Resultado Edge Detection Fonte: (PATIN, 2003)

3.2.2 Cantos (Corners) / Pontos de interesse

Os termos cantos e pontos de interesse são utilizados como sinônimos e referem-se tanto a pontos como características de uma imagem, Figura 15. O nome "canto" surgiu desde algoritmos iniciais utilizados na detecção de bordas, e depois se analisou as bordas para encontrar mudanças rápidas de direção (cantos). Estes algoritmos foram desenvolvidos também para procurar níveis elevados de curvatura no gradiente de imagem. Em seguida, foram notados que os cantos assim chamados também formam algumas partes da imagem detectadas que não são cantos, no sentido tradicional (por exemplo, uma pequena mancha brilhante sobre um fundo escuro pode ser detectado). Esses pontos são frequentemente conhecidos como pontos de interesse, mas o "canto" é utilizado pela tradição.

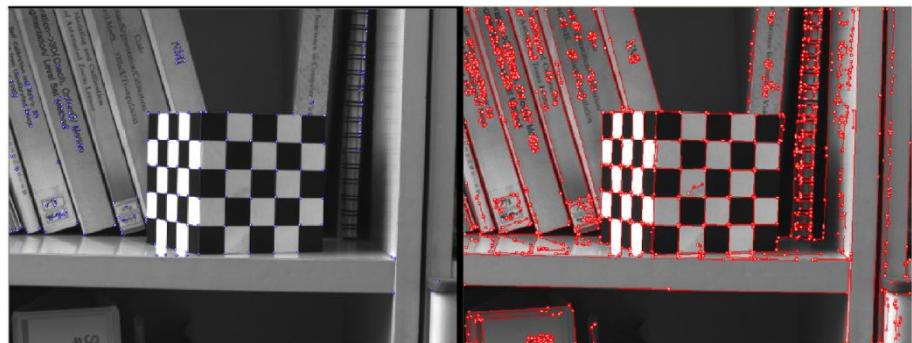


Figura 15: Resultado *Corner Detection* Fonte: (HA, 2007)

3.2.3 Regiões de interesse (*Blobs*)

Blobs fornecem uma descrição complementar das estruturas da imagem em termos de regiões, em oposição aos cantos que são pontos característicos, Figura 16. No entanto, descritores *blobs* geralmente contêm um ponto preferido (um máximo local de uma resposta do operador ou de um centro de gravidade), o que significa que os detectores de *blobs* podem também ser considerados como operadores de pontos de interesse. Os detectores de *blobs* podem detectar áreas de uma imagem, que são muito suaves para serem detectadas por um detector de cantos.

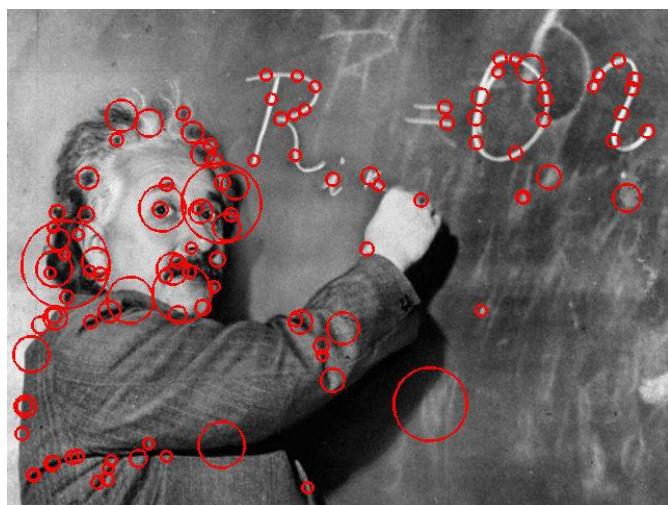


Figura 16: Resultado de detecção de regiões. Fonte:(KIXOR, 2008)

3.2.4 Ridges

Para objetos alongados, *Ridges* são uma ferramenta natural. Um descritor do tipo *ridges* calcula, a partir de uma imagem de níveis de cinza, uma generalização de um eixo mediano, Figura 17. Do ponto de vista prático, pode ser pensado como uma curva unidimensional, que representa um eixo de simetria na imagem. Infelizmente, no entanto, é mais difícil de extrair essas características a partir de imagens gerais de níveis de cinza do que os detectores de Bordas, Cantos ou *Blobs*. Mesmo assim, descritores *ridges* são frequentemente usados para a extração de rodovias em imagens aéreas e para a extração de vasos sanguíneos em imagens médicas.

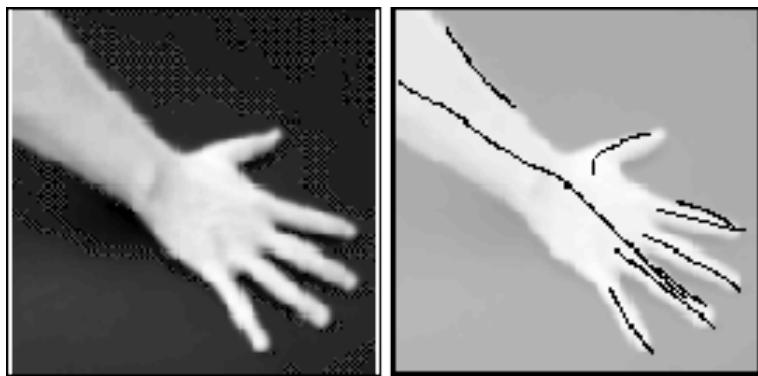


Figura 17: Exemplo características do tipo Ridges. Fonte: (LINDEBERG, 1996)

3.3 TIPOS DE DETECTORES DE CARACTERÍSTICAS

Uma vez que as características principais são usadas como ponto de partida nos algoritmos, estes, em geral, oferecem um bom resultado na detecção. A repetibilidade é o fator principal em um detector de características, ou seja, se a característica desejada será detectada, ou não, em duas ou mais imagens diferentes na mesma cena.

Ocasionalmente, quando um detector de características tem custo computacional elevado e não há restrições de tempo (o que não se aplica no escopo deste trabalho, uma vez que o tempo de processamento é fundamental para o sucesso), um algoritmo de nível mais elevado pode usado para guiar na identificação da cena.

A seguir a Tabela 1 mostra alguns dos detectores usados com a característica que é capaz de capturarem de uma cena:

Tabela 1: Detectores de características e suas classificações

DETECTOR	CLASSE		
	Bordas	Cantos	Regiões
Canny (CANNY, 1986)	X		
Sobel (CHIVAPREECHA; DEJHAN; PIENVIJARNPONG, 2004)	X		
Harris & Stephens (HARRIS; STEPHENS, 1988)	X	X	
SUSAN (MUYUN; MINGYI, 2006)	X	X	
Shi & Tomasi (SHI; TOMASI, 1994)		X	
FAST (ROSTEN, E.; DRUMMOND, 2005)(2006)		X	X
MSER (MATAS <i>et al.</i> , 2004)			X
SURF(BAY <i>et al.</i> , 2008)			X

3.4 DESCRIÇÃO DE CARACTERISTICAS

Após a detecção dos pontos de interesse em uma imagem o procedimento habitual é extrair um descritor de cada um desses pontos, ou seja, descrevê-los de forma padronizada para serem utilizados no processo de correlação.

O descritor é um vetor de características extraídas para um ponto e sua localização correspondente. O vetor de características deriva dos pixels ao redor do ponto de interesse. Esses pixel descrevem e relacionam características específicas para um único ponto. O método utilizado por um descritor depende do tipo de característica extraída.

4 SISTEMA DE VISÃO ARTIFICIAL

A Figura 18 apresenta a estrutura do SVA desenvolvido neste trabalho. A seguir são debatidos cada elemento desta estrutura.

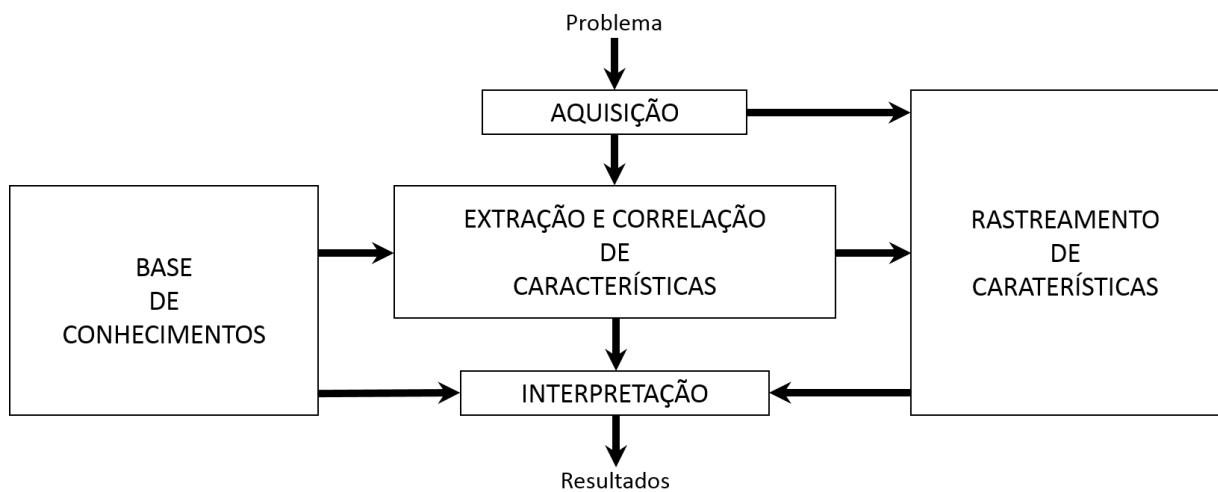


Figura 18: Sistema de visão artificial utilizado neste trabalho.

4.1 PROBLEMA E RESULTADOS

O problema consiste em elaborar um algoritmo capaz de realimentar visualmente um controlador em um robô que manipula uma válvula, apresentada na Figura 19, permitindo que este tenha informações acerca da rotação da válvula. O resultado esperado do algoritmo é, então, o ângulo de rotação da válvula em relação à sua posição inicial.



Figura 19: Válvula do CT-3.

4.2 AQUISIÇÃO

A primeira etapa no SVA proposto é a aquisição de imagens. Nesta etapa se definem as condições de captura dos quadros. Os parâmetros considerados nesta etapa são descritos a seguir.

4.2.1 Dispositivo e imagem

Como dispositivo de captura utilizou-se uma câmera DSC-W510 da Sony. Este modelo de câmera utiliza um sensor Charge-Coupled Device (CCD) e estabilizador de imagem. A especificação da câmera é apresentada na Tabela 2.

Tabela 2: Especificação da câmera Sony DSC-W510; Fonte: Manual do fabricante.

CARACTERÍSTICA	ESPECIFICAÇÃO
Dispositivo de imagem	CCD colorido de 7,79 mm (tipo 1/2,3) com filtro de cor primária
Número total de pixels	12,1 Megapixels efetivo
Zoom Óptico	4x; F = 4,7mm
Lente	Lentes Sony
Estabilizador de Imagem	Steady Shot
Tipo de Sensor	Super HAD CCD
Distância Focal	Imagens 16 por 9 - 28-114mm
	Imagens 4 por 3 - 26-105mm
	Vídeo 4 por 3 - 26-105mm
Dimensões	9,6 x 5,4 x 1,99 cm (LxAxP).
Temperaturas de operação	0°C a +40°C
Filmação	640 x 480(VGA) / 320 x 240 (QVGA)

As aquisições foram realizadas a taxas de 29 quadros por segundo e então processadas em um computador XPS15 L502X, cujas configurações se encontram na Tabela 3.

Tabela 3: Especificação do computador XPS15 L502x utilizado no processamento das imagens;
Fonte: Manual do fabricante.

CARACTERÍSTICA	ESPECIFICAÇÃO
Processador	Intel® Core™ i5-2450M CPU @ 2.50GHz 2.50GHz
Sistema operacional	Windows 7 Ultimate 64-bit, x64-based processor.
Memória	6,0GB DDR3 1333MHz
Placa de vídeo	Intel® HD Graphics 3000 (Onboard)/ NVIDIA GeForce GT 525M (Offboard)

4.2.2 Posicionamento do dispositivo

O algoritmo parte da suposição de que a câmera está posicionada conforme um padrão fixo estabelecido para a aplicação. Ou seja, dada uma imagem padrão (Figura 20) a câmera deve ser posicionada de forma capturar esta mesma imagem, em escala e posição. As técnicas de controle para o posicionamento satisfatório da câmera, por um manipulador robótico, são descritas em (CHAUMETTE; HUTCHINSON, 2006)(2007).



Figura 20: Padrão de imagem para posicionamento de câmera.

No caso deste trabalho, como não se dispunha de um sistema de manipulação e controle para o posicionamento automático da câmera, a mesma foi manualmente posicionada sobre um suporte. Desta forma, não se pode garantir a precisão com respeito ao padrão, assim, foi necessário alimentar manualmente a base de conhecimento a cada teste.

4.2.3 Iluminação

A válvula utilizada se encontra no pátio do CT III a céu aberto (Figura 21) e, portanto, sujeita a diferentes intensidades de iluminação ao longo do dia. De forma a analisar a robustez do algoritmo em relação a esta condição, realizou-se a aquisição de imagens em 3 horários ao longo do dia (9h, 12h e 16h)



Figura 21: Pátio do CT-3.

4.2.4 Resolução

As imagens foram adquiridas em duas resoluções típicas da câmera.

640x480 pixels: imagens de melhor qualidade e que exigem maior processamento;

320x240 pixels: imagens processadas mais rapidamente, porém com menos informações em cada quadro.

4.2.5 Fundo

Para avaliar a influência do fundo nos resultados do algoritmo foram realizados testes com o fundo normal (complexo) e utilizando um anteparo como fundo branco.



Figura 22: Aquisição em (a) fundo complexo e em (b) fundo branco.

4.2.6 Calibração de câmera

Para estratégias de controle baseadas em posição (PBVS), a determinação da pose do objeto em relação a câmera é requerida para o cálculo da cinemática inversa do robô. Por isso, a etapa de calibração é muito importante na implementação destas técnicas. Entretanto, no caso deste trabalho, em que a estratégia de controle é baseada em imagem (IBVS), a determinação da pose não é relevante para o controle, dispensando a necessidade da realização da calibração (CHONG, 2004).

4.3 EXTRAÇÃO E CORRELAÇÃO DE CARACTERÍSTICAS

A Figura 23 apresenta o pseudocódigo utilizado nesta etapa. Primeiramente são detectados pontos característicos do quadro capturado através de detectores de características. Em seguida são extraídas as características que descrevem esses pontos utilizando-se o extrator de descritores padrão do MATLAB® 2013 através da função `extractFeatures`. Estas são então correlacionadas com características presentes na base de conhecimento e, que foram extraídas do padrão na Figura 20. Separam-se os pontos característicos que melhor descrevem este padrão (*inliers*) e contém informações que os distinguem melhor dos demais. Por fim, selecionam-se dez destes pontos para serem utilizados na próxima etapa. Caso não tenham sido encontrados pontos *inliers* suficientes o algoritmo utiliza os pontos correlacionados para fazer o rastreamento.

```
% EXTRAÇÃO E CORRELAÇÃO DE CARACTERÍSTICAS

% Detecta e extraí características do quadro.
>> pCaracteristicos_Quadro = detectaCaracteristicas(quadro);
>> caracteristicas_Quadro = extractFeatures(quadro, pCaracteristicos_Quadro);

% Correlaciona as características encontradas com uma base de dados.
>> pCorrelacionados = correlacionaPontosCaracteristicos(pCaracteristicos_Quadro, , pCaracteristicos_Padrao);

% Separa pontos que mais se assemelham ao modelo e seleciona os 10 primeiros.
>> pCorrelacionados_Padrao = pCorrelacionados(:, 1);
>> pCorrelacionados_Quadro = pCorrelacionados(:, 2);
>> [matriz_de_transformação, inliers, outliers] = estimaTransformacaoGeometrica(pCorrelacionados_Padrao, ...
>> pCorrelacionados_Quadro, 'semelhança');
>> p10 = inliers(10)
```

Figura 23: Pseudocódigo de extração e correlação de características em um quadro capturado.

Os testes foram realizados com o detector de características SURF ([Speeded Up Robust Features](#)). Ferramenta da *toolbox* de visão computacional do MATLAB® 2013, *detectSURFFeatures*.

4.4 INTERPRETAÇÃO

O ângulo de rotação da válvula é interpretado como o ângulo entre dois vetores com origem no centroide da válvula (informação na base de dados), direção, sentido e módulo com relação a um mesmo ponto característico em quadros de imagem diferentes. O primeiro vetor (\vec{u}) é obtido no primeiro quadro capturado e o segundo (\vec{v}) é encontrado nos quadros subsequentes. O ângulo entre os vetores é dado pela expressão:

$$\cos\theta = \frac{u \cdot v}{\|u\| \cdot \|v\|} \quad (2)$$

Figura 24 - Vetores u e v obtidos e ângulo entre eles.

Os ângulos são calculados para todos os pontos característicos a cada quadro. Considera-se como ângulo de rotação da válvula o maior dos ângulos encontrados, pois com o passar dos quadros algumas características se perdem, principalmente por variações de iluminação sobre o objeto, deixando de incrementar o valor dos ângulos nestes pontos.

Abaixo é apresentado o pseudocódigo de interpretação.

```
% INTERPRETAÇÃO
```

```
% Cria vetor no primeiro quadro capturado.  
>> para i = 1:10, faça  
>>     u(i, :) = [p10(i, 1) - centroide_x, p10(i, 2) - centroide_y];  
>> fim  
  
% Cria vetor a cada quadro e calcula o ângulo.  
>> enquanto captura_de_quadro = verdade, faça  
>>     para i = 1:10, faça  
>>         v(i, :) = [p_Quadro(i, 1) - centroide_x, p_Quadro(i, 2) - centroide_y];  
>>         cos_Theta = multiplicação_escalar(u(i,:),v(i,:))/(unitário(u(i,:))*unitário(v(i,:)));  
>>         theta_Ponto(i) = acos(cos_Theta)*180/pi;  
>>     fim  
>>     Theta = máximo(theta_Ponto);  
>>fim
```

Figura 25: Pseudocódigo de interpretação dos pontos característicos para obtenção do ângulo de rotação da válvula.

4.5 BASE DE CONHECIMENTO

É uma base de dados que contém informações preliminares necessárias ao funcionamento do algoritmo. Estas informações podem ter sido adquiridas através de outros algoritmos ou manualmente. Neste trabalho foram consideradas duas informações:

1. Pontos característicos do objeto e características extraídas destes pontos; e
2. Posição do centroide nas coordenadas cartesianas de aquisição das imagens.

4.6 RASTREAMENTO

A etapa de extração e correlação de características executa inúmero cálculos iterativos em cada pixel da imagem. A quantidade de cálculos varia com o tipo de detector de características e com o tamanho da imagem. A extração e correlação de características, caso fossem realizadas quadro a quadro, demandariam um tempo computacional inviável para aplicações em robótica. Para solucionar este problema utilizamos um rastreador de pontos característicos (KANADE; LUCAS, 1981) (KANADE; TOMASI, 1991), parte da toolbox de visão computacional do MATLAB® 2013. O rastreador funciona limitando a busca pelo ponto característico, em um quadro, a pixels próximos à posição em que ele se encontrava no quadro anterior. Abaixo tem-se o pseudocódigo do rastreador:

```
% RASTREAMENTO

% Cria um rastreador.

>> rastreador = rastreadorPontos('Erro máximo', 1);

>> inicializa(rastreador, p10, Quadro);

% Inicia o rastreamento dos pontos

>> enquanto captura_de_quadro = verdade, faça
    >>     p_quadro = step(rastreador, Quadro_atual);
    >> fim
```

Figura 26: Pseudocódigo do rastreador de pontos característicos.

A etapa de extração e correlação de característica é realizada apenas no primeiro quadro capturado para obtenção dos pontos característicos que o rastreador deve seguir. O rastreador é criado a partir da função *vision.PointTracker*.

5 RESULTADOS

O sucesso das técnicas de controle visual está definido pela sua capacidade de cumprir integralmente a tarefa para a qual foi planejado. Neste contexto, cada etapa do processo deve desempenhar seu papel com maior precisão possível. Para identificar as fraquezas e virtudes dos sistemas de controle implementados durante esta pesquisa foram feitos testes de cada uma das etapas envolvida (MUÑOZ, 2011).

5.1 TESTES EXECUTADOS

Foram executados ao todo nove testes, com o objetivo de avaliar o algoritmo em diferentes condições de iluminação (4.2.3), resolução (4.2.4) e fundo (4.2.5). A Tabela 4 apresenta os parâmetros de cada teste executado.

Tabela 4: Testes executados.

TESTE	HORA	RESOLUÇÃO	FUNDO
1	9h	640x480	Complexo
2	12h	640x480	Complexo
3	16h	640x480	Complexo
4	9h	640x480	Branco
5	12h	640x480	Branco
6	16h	640x480	Branco
7	9h	320x240	Branco
8	12h	320x240	Branco
9	16h	320x240	Branco

5.2 EXTRAÇÃO E CORRELAÇÃO DE CARACTERÍSTICAS

A tabela abaixo apresenta a quantidade de pontos característicos encontrados em cada etapa da extração e correlação de características (4.3). Imagens com as informações contidas na Tabela 4 encontram-se no Apêndice bB.

Tabela 5: Quantidade de pontos encontrados na extração e correlação de características (4.3); coluna “TESTE” conforme Tabela 4.

TESTE	QUANTIDADE DE PONTOS		
	Característicos	Correlacionados	<i>Inliers</i>
1	623	8	3
2	438	13	4
3	347	9	4
4	391	13	3
5	241	45	6
6	271	50	34
7	222	6	4
8	151	22	6
9	164	24	16

Percebe-se que em fundo complexo, testes 1 a 3, foram detectados maior quantidade de pontos característicos. Isso ocorre pois o detector de características não distingue pontos do objeto e do fundo. No caso do fundo branco nota-se a incidência maior de pontos sob a condição de iluminação direta devido a presença de sobras.

Nota-se uma baixa quantidade de pontos correlacionados no caso de fundo complexo e/ou sob condições de iluminação direta (9h). A baixa quantidade de pontos correlacionados reflete-se na quantidade de *inliers*.

No caso de fundo complexo, a baixa quantidade de pontos correlacionados ocorre pois o algoritmo relaciona características do padrão a características do fundo, como pode-se notar na Figura 27. Já sob iluminação direta, acontece uma suavização da textura da válvula, minimizando as diferenças entre os pontos característicos.

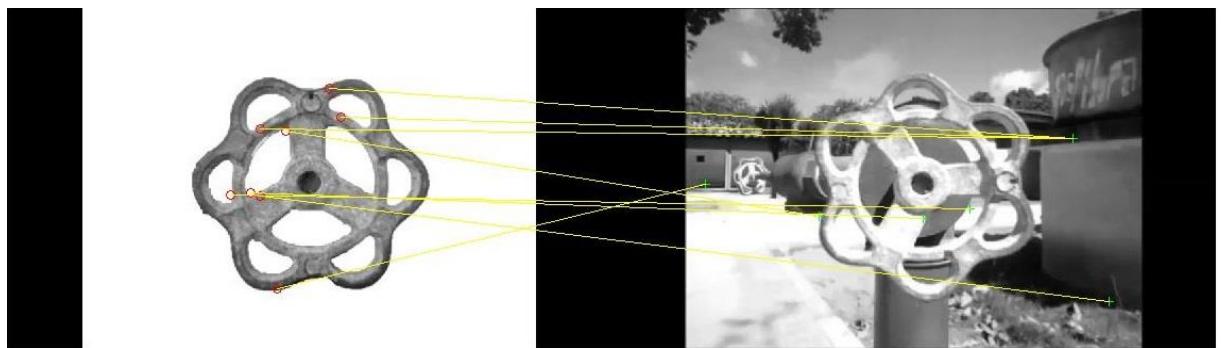


Figura 27: Pontos correlacionados no teste 1.

A redução da quantidade de pixels na imagem, testes 7 a 9, também reduz a quantidade de informação contida na mesma, ocorrendo não só a detecção de menos pontos característicos, como a qualidade destes pontos é reduzida levando a uma menor quantidade de pontos correlacionados.

Ao fim desta etapa o algoritmo seleciona pontos para serem rastreados e interpretados na próxima etapa a fim de se obter o ângulo da válvula. Estes pontos são apresentados na Tabela 11 no Apêndice bB.

5.3 INTERPRETAÇÃO

Esta etapa visa interpretar a posição dos pontos característicos no primeiro quadro capturado, selecionados na etapa de extração e correlação de características (4.3) e, rastreados (4.6) nos quadros subsequentes. A Tabela 6 apresenta os ângulos encontrados, ao final da aquisição, para cada ponto rastreado.

Tabela 6: Ângulos finais calculados conforme 4.4 em cada ponto rastreado; coluna “TESTE” conforme Tabela 4.

TESTE	PONTOS									
	1	2	3	4	5	6	7	8	9	10
1	40,63	101,75	0,88	23,12	0,04	0,04	0,04	0,07		
2	0,06	0,07	51,69	26,02	39,80	39,80	39,80	39,80	21,44	21,44
3	11,47	10,77	39,03	75,65	75,65	75,65	75,65	31,99	31,99	
4	8,22	4,81	25,81	25,81	81,00	68,44	94,45	94,45	0,58	0,58
5	64,88	91,95	78,28	78,28	78,28	78,28	83,43	65,70	58,23	27,92
6	90,24	88,61	93,97	87,19	79,03	92,19	93,76	75,25	78,42	92,61
7	25,70	36,28	36,28	36,28	89,50	89,50				
8	74,31	74,31	77,40	77,40	78,87	35,04	60,06	67,08	45,27	94,17
9	87,05	92,83	80,52	66,80	77,86	76,73	96,85	86,04	78,43	94,31

Na tabela acima nota-se que os ângulos calculados para cada ponto em um mesmo teste são discrepantes, isto pode ser verificado na Tabela 7, pelo desvio padrão.

Tabela 7: Resultados encontrados pelo algoritmo; coluna “TESTE” conforme Tabela 4.

TESTE	MÉDIA	DESVIO PADRÃO	MÁXIMO
1	20.82	36.00	101.75
2	27.99	17.54	51.69
3	47.54	28.22	75.65
4	40.42	39.69	94.45
5	70.53	17.91	91.95
6	87.13	7.00	93.97
7	52.26	29.14	89.50
8	68.39	17.40	94.17
9	83.74	9.38	96.85

Se as características de um ponto rastreado não são suficientemente invariantes com a iluminação ou não o distinguem bem dos demais, o rastreador pode não conseguir encontrá-lo no quadro subsequente, ou ainda confundi-lo com outro pixel, do próprio objeto ou do fundo. O ângulo calculado a partir destes pontos não varia, necessariamente, com o ângulo de rotação da válvula.

Os testes com maior ocorrência de pontos *inliers*, 6 e 9, apresentaram desvio padrão menor, pois as características destes pontos possuem informações que os definem melhor.

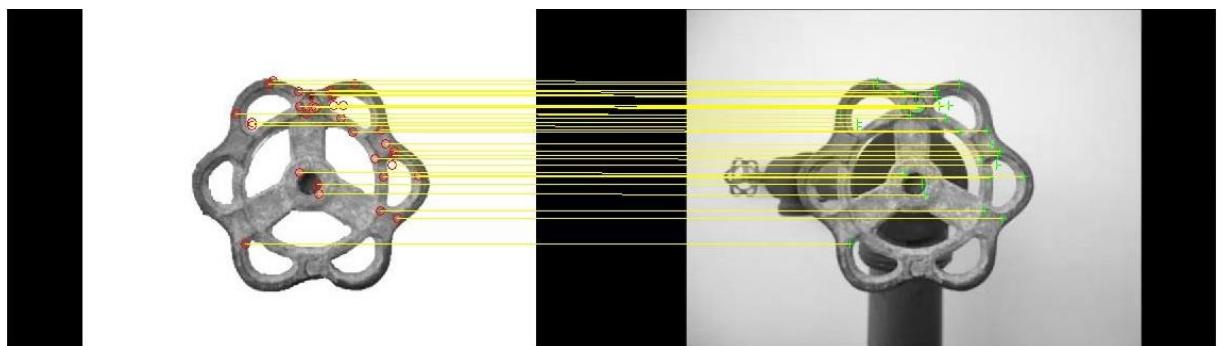


Figura 28: *Inliers* encontrados no teste 6.

5.4 MEDIDA DE ERRO

A Tabela 8 apresenta os ângulos calculados pelo algoritmo, aferidos manualmente a partir do primeiro e último quadros capturados (Apêndice C - POnotos característicos aferidos manualmente) e o erro percentual dos valores encontrados pelo algoritmo. O maior entre os ângulos calculados para os pontos rastreados (Máximo, Tabela 7) é considerado como o ângulo de rotação da válvula encontrado pelo algoritmo

Tabela 8: Ângulos de rotação da válvula aferidos automaticamente, manualmente e erro percentual; coluna “TESTE” conforme Tabela 4.

TESTE	ÂNGULO CALCULADO (º)		ERRO (%)
	Algoritmo	Manualmente	
1	101.75	85.04	19.65
2	51.69	95.33	45.78
3	75.65	85.83	11.86
4	94.45	91.76	2.93
5	91.95	88.22	4.23
6	93.97	90.43	3.91
7	89.50	93.47	4.25
8	94.17	94.72	0.58
9	96.85	89.96	7.66

Nota-se, da Tabela 8, que os testes realizados com fundo complexo, 1 a 3, não obtiveram bons resultados. Nos testes com fundo branco, 4 a 9, os resultados foram razoáveis, atendendo ao que propunha.

Não se percebe influência da variação de iluminação e da resolução nos resultados finais encontrados.

6 CONCLUSÃO

Na etapa de extração a redução da resolução da imagem capturada tem como consequência a redução da quantidade de pontos extraídos da cena. Já na correlação a iluminação é um parâmetro importante, pois sob luz direta são correlacionados menos pontos. Muitos pontos não foram bem rastreados, gerando discrepâncias entre os ângulos calculados a partir deles.

Não se nota influência da variação de iluminação e da redução de resolução nos resultados finais obtidos. O fundo é o principal parâmetro a ser considerado, pois implica em erros que não permitem a aplicação do algoritmo.

Sugere-se como trabalhos futuros:

1. Criação de um banco de banco dados com padrões de referência sujeitos a iluminações diferentes;
2. Utilização de um rastreador de pontos característicos mais robusto;
3. Utilização de técnicas de remoção de fundo;
Simulação de um sistema de controle robótico no Simulink®;
4. Implementação em outras linguagens computacionais.

7 REFERENCIAS

1. ANDERSSON, R. L. Real time expert system to control a robot ping-pong player. 1987. University of Pennsylvania, Philadelphia, PA, USA, 1987.
2. BAUERMANN, G. Medição de área foliar. Disponível em: <<http://www.imagesurvey.com.br/2009/02/tutorial-imagej-medicao-de-area-foliar/>>. Acesso em: 28 abr. 2013.
3. BAY, H.; ESS, A.; TUYTELAARS, T.; GOLL, L. V. Speeded-Up Robust Features (SURF). Computer Vision and Image Understanding, v. 110, n. 3, p. 346–359, jun. 2008.
4. BERNARDES, M. C. Controle servo-visual para aproximação de portas por robôs móveis equipados com duas câmeras. 2009. Dissertação de Mestrado – Universidade de Brasília, Brasília, 2009.
5. BUKOWSKI, R. et al. Robot hand-eye coordination rapid prototyping environment. 1991, [S.l: s.n.], 1991.
6. BUTTAZZO, G. C.; ALLOTTA, B.; FANIZZA, F. P. Mousebuster: a robot system for catching fast moving objects by vision. 1993, [S.l: s.n.], 1993. p. 932–937.
7. CANNY, John. A computational approach to edge detection. Pattern Analysis and Machine Intelligence, IEEE Transactions on, n. 6, p. 679–698, 1986.
8. CHAUMETTE, F.; HUTCHINSON, S. Visual servo control Part I: Basic approaches. IEEE Robotics Automation Magazine, v. 13, n. 4, p. 82 –90, dez. 2006.
9. CHAUMETTE, F.; HUTCHINSON, S. Visual servo control Part II. Advanced approaches [Tutorial]. Robotics & Automation Magazine, IEEE, v. 14, n. 1, p. 109–118, 2007.
10. CHIVAPREECHA, S.; DEJHAN, K.; PIENVIJARNPONG, C. Hardware Implementation of Sobel-Edge Detection Distributed Arithmetic Digital Filter. 2004, [S.l: s.n.], 2004. p. 284–289.
11. CHONG, M. H. Visual servoing. 2004.
12. CHUM O.; MATAS, J.; PAJDLA, T.; URBAN, M. Robust wide-baseline stereo from maximally stable extremal regions. Image and vision computing, v. 22, n. 10, p. 761–767, 2004.
13. CORKE, Peter I. Visual Control of Robots: high-performance visual servoing. [S.I.]: Research Studies Press Taunton, Somerset, England, 1996.

14. DICKMANNS, Ernst Dieter; GRAEFE, Volker. Applications of dynamic monocular machine vision. *Machine vision and Applications*, v. 1, n. 4, p. 241–261, 1988.
15. FÄSSLER, H.; BEYER, H. A.; WEN, J. A robot ping pong player: optimized mechanics, high performance 3D vision, and intelligent sensor control. *Robotersysteme*, v. 6, n. 3, p. 161–170, 1990.
16. HA, J. Automatic detection of calibration markers on a chessboard. 2007
17. HARRELL, R. C.; SLAUGHTER, D. C.; ADSIT, P. D. A fruit-tracking system for robotic harvesting. *Machine Vision and Applications*, v. 2, n. 2, p. 69–80, 1989.
18. HARRIS, Chris; STEPHENS, Mike. A combined corner and edge detector. 1988, [S.l.: s.n.], 1988. p. 50.
19. HASHIMOTO, Hideki et al. A control scheme of visual servo control of robotic manipulators using artificial neural network. 1989, [S.l.: s.n.], 1989. p. 68–69.
20. HILL, J.; PARK, W. T. Real time control of a robot with a mobile camera. In: *INTERNATIONAL SYMPOSIUM ON INDUSTRIAL ROBOTS*, Washington: [s.n.], 1979.
21. KANADE, T.; LUCAS, B. An Iterative Image Registration Technique with An Application to Stereo Vision. In: *INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE*, p. 674–679, 1981.
22. KANADE, T.; TOMASI, C. Detection and Tracking of Point Features. Carnegie Mellon University Technical Report, 1991.
23. KIXOR, T. Blob detection. Disponível em: <<http://www.kixor.net/school/2008spring/comp776/assn1/>>. Acesso em: 28 abr. 2013.
24. KUPERSTEIN, M. Generalized neural model for adaptive sensory-motor control of single postures. 1988, [S.l.: s.n.], 1988. p. 140–144.
25. LINDEBERG, T. Scale-space: A framework for handling image structures at multiple scales. Disponível em: <<http://www.nada.kth.se/~tony/cern-review/cern-html/node17.html>>. Acesso em: 28 abr. 2013.
26. MARENCONI, M.; STRINGHINI, S. Tutorial: Introdução à Visão Computacional usando OpenCV. *Revista de Informática Teórica e Aplicada*, v. 16, n. 1, p. 125–160, 8 mar. 2010.
27. MARQUES FILHO, O.; VIEIRA NETO, H. Processamento digital de imagens. Rio de Janeiro: BRASPORT, 1999.
28. MEL, B. W. Connectionist robot motion planning: a neurally-inspired approach to visually-guided reaching. [S.l.]: Academic Press Professional, Inc., 1990.

29. MELO, N. Abordagens do processo de Segmentação: Limiarização, Orientada a Regiões e Baseada em Bordas. Disponível em: <<http://www.dsc.ufcg.edu.br/~pet/jornal/setembro2011/materias/recapitulando.html>>. Acesso em: 29 abr. 2013.
30. MILLER III, W. Sensor-based control of robotic manipulators using a general learning algorithm. *Robotics and Automation, IEEE Journal of*, v. 3, n. 2, p. 157–165, 1987.
31. MUÑOZ, G. L. L. Análise comparativa das técnicas de controle servo-visual de manipuladores robóticos baseadas em posição e em imagem. 2011.
32. MUYUN, Weng; MINGYI, He. Image feature detection and matching based on SUSAN method. 2006, [S.I.: s.n.], 2006. p. 322–325.
33. ORLANDI, J. G. N. Sistema de visão a laser para mapeamento de superfície de robos quadrúpedes. 2008. UFES, Vitória, 2008.
34. PAPANIKOLOPOULOS, N. P.; KHOSLA, P. K. Shared and traded telerobotic visual control. 1992, [S.I.: s.n.], 1992. p. 878–885.
35. PATIN, F. An introduction to Digital Image Processing - Graphics programming and theory. Disponível em: <http://www.gamedev.net/page/resources/_/technical/graphics-programming-and-theory/an-introduction-to-digital-image-processing-r2007>. Acesso em: 28 abr. 2013.
36. ROSEN, C. Machine Intelligence Research Applied to Industrial Automation: Sixth Report. [S.I.]: Standard Research Institute, 1976.
37. ROSTEN, E.; DRUMMOND, T. Fusing points and lines for high performance tracking. 2005, [S.I.]: IEEE, 2005. p. 1508–1515 Vol. 2.
38. ROSTEN, E.; DRUMMOND, T. Machine Learning for High-Speed Corner Detection. In: LEONARDIS, ALEŠ; BISCHOF, HORST; PINZ, AXEL (Org.). *Computer Vision – ECCV 2006*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006. v. 3951. p. 430–443.
39. SAKAGUCHI, T. et al. Motion planning and control for a robot performer. 1993, [S.I.: s.n.], 1993. p. 925–931.
40. SANDERSON, A. C.; WEISS, L. E. Image-based visual servo control using relational graph error signals. *Proc. ieee*, v. 1074, 1980.
41. SHI, J.; TOMASI, C. Good features to track. 1994, [S.I.: s.n.], 1994. p. 593–600.
42. SHIRAI, Y.; INOUE, H. Guiding a robot by visual feedback in assembling tasks. *Pattern Recognition*, v. 5, n. 2, p. 99–108, jun. 1973.

43. SIVIC, J.; ZISSERMAN, A. Video Google: A text retrieval approach to object matching in videos. 2003, [S.l.: s.n.], 2003. p. 1470–1477.
44. TENDICK, F.; STARK, L; THARP, G.; VOICHICK, J. A supervisory telerobotic control system using model-based vision feedback. 1991, [S.l.: s.n.], 1991. p. 2280–2285.
45. TUYTELAARS, T; MIKOLAJCZYK, K. Local invariant feature detectors: a survey. *Found. Trends. Comput. Graph. Vis.*, v. 3, n. 3, p. 177–280, jul. 2008.
46. YUAN, J. S.; MACDONALD, R. A; KEUNG, Felix HN. Telerobotic tracker. . [S.l.: s.n.]. , jul. 1990
47. ZHANG, D. B.; VAN GOOL, L.; OOSTERLINCK, A. Stochastic predictive control of robot tracking systems with dynamic visual feedback. 1990, [S.l.: s.n.], 1990. p. 610–615.

APÊNDICE A - ALGORITMO

```

%%% BASE DE DADOS
C = [430,248]; % Centroide nas coordenadas da aquisição de imagens
Template = rgb2gray(imread('template.jpg'));% Carrega o padrão
templatePoints = detectSURFFeatures(Template); % Encontra pontos
características do padrão
[templateFeatures, templateValid_points] = extractFeatures(Template,
templatePoints);% Extrai características do padrão
% FIM DA BASE DE DADOS

%%% AQUISIÇÃO
videoFReader = vision.VideoFileReader('video.mp4');%%% Carrega video/câmera
firstFrame = rgb2gray(step(videoFReader));%%% Captura o primeiro frame
%(CONTINUA)

%%% EXTRAÇÃO E CORRELAÇÃO
firstPoints = detectSURFFeatures(firstFrame); % Encontra pontos
características do quadro
[firstFeatures, firstValid_points] = extractFeatures(firstFrame,
firstPoints); % Extrai as características dos pontos encontrados
indexPairs = matchFeatures(templateFeatures, firstFeatures); % Correlaciona
os pontos característicos
templateMatched_points = templateValid_points(indexPairs(:, 1), :);%
Localização dos pontos correlacionados, no primeiro quadro
firstMatched_points = firstValid_points(indexPairs(:, 2), :); % Localização
dos pontos correlacionados, no padrão
[tform,template_inliers,first_inliers]=
estimateGeometricTransform(templateMatched_points,firstMatched_points,'affi
ne'); % Encontra os inliers

[row,col] = size(first_inliers.Location);
if row>10
    first2follow = first_inliers.selectStrongest(10); % Seleciona os 10
    pontos para serem rastreados na próxima etapa
    c =10;
else
    first2follow = firstMatched_points.selectStrongest(10); % Seleciona os
    10 pontos para serem rastreados na próxima etapa
    c = row;
end
%FIM DA ETAPA DE EXTRAÇÃO E CORRELAÇÃO

%%% INTERPRETAÇÃO
for i = 1:c
u(i,:) = [first2follow.Location(i,1)-C(1), first2follow.Location(i,2)-
C(2)]; % Cria vetor u para o ponto no primeiro quadro
end
%(CONTINUA)

%%% RASTREAMENTO
markerInserter =
vision.MarkerInserter('Shape','Plus','BorderColor','White'); % Cria uma
marca para acompanhas os pontos na imagem (DESNECESSÁRIO)
tracker = vision.PointTracker('MaxBidirectionalError', 1); % Cria um
rastreador

```

```

initialize(tracker, first2follow.Location, firstFrame); % Inicializa o
rastreador
% CONTINUA

%%% AQUISIÇÃO (CONTINUAÇÃO)
while ~isDone(videoFReader) % Inicia a captura de frames
    videoFrame = rgb2gray(step(videoFReader)); % Captura um frame e
transforma o quadro para escala de cinza
    % CONTINUA

%%% RASTREAMENTO (CONTINUAÇÃO)
[points, validity] = step(tracker, videoFrame); % Rastreia os pontos
no frame
% FIM DO RASTREAMENTO

%%% INTERPRETAÇÃO (CONTINUAÇÃO)
for i = 1:c
    v(i,:) = [points(i,1)-C(1), points(i,2)-C(2)]; % Cria vetor v para o
ponto no quadro atual
    CosTheta = dot(u(i,:),v(i,:))/(norm(u(i,:))*norm(v(i,:))); % Calcula
o coseno do ângulo entre o vetor u e v
    ThetaInDegrees(i) = acos(CosTheta)*180/pi; % Calcula o ângulo em
graus
end
Ang = max(ThetaInDegrees)
% FIM DA INTERPRETAÇÃO

%%% AQUISIÇÃO (CONTINUAÇÃO)
end
release(videoPlayer); % Libera o video
release(videoFReader); % Libera o player
% FIM DA AQUISIÇÃO

```

APÊNDICE B - PONTOS OBTIDOS NA EXTRAÇÃO E CORRELAÇÃO DE CARACTERÍSTICAS

Tabela 9: Pontos correlacionados; coluna “TESTE” conforme Tabela 4. (PARTE 1 – 1/3)

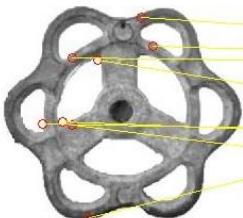
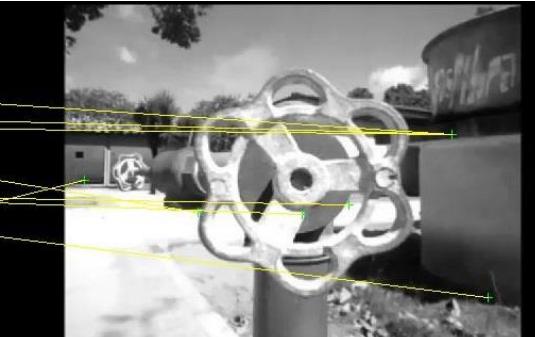
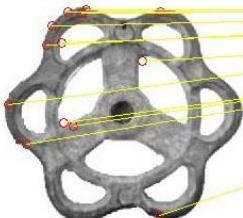
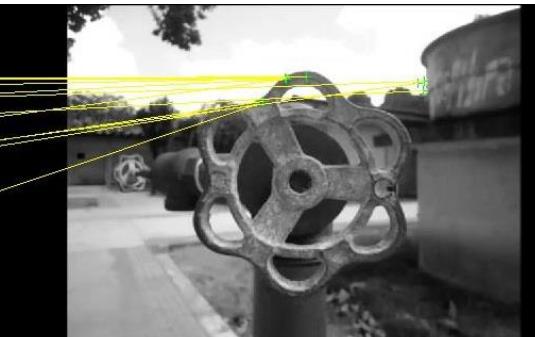
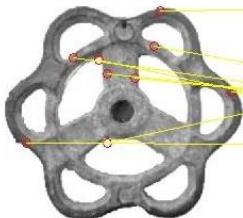
TESTE	CORRELAÇÃO
1	 
2	 
3	 

Tabela 9: Pontos correlacionados; coluna “TESTE” conforme Tabela 4. (CONTINUAÇÃO – 2/3)

TESTE	CORRELAÇÃO		
4			
5			
6			

Tabela 9: Pontos correlacionados; coluna “TESTE” conforme Tabela 4. (CONTINUAÇÃO – 3/3)

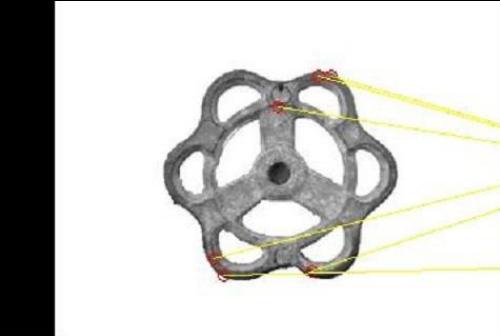
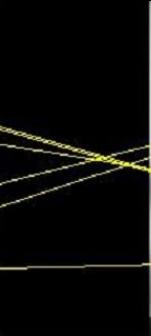
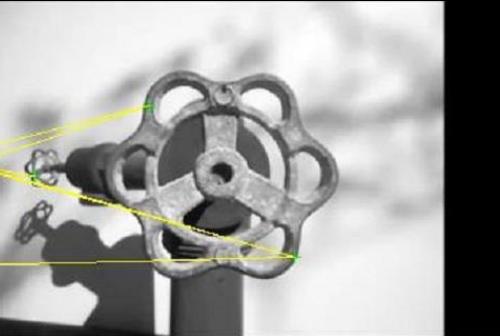
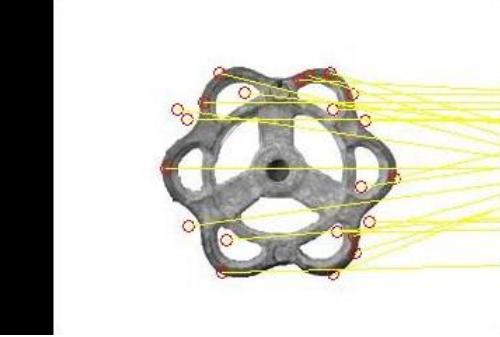
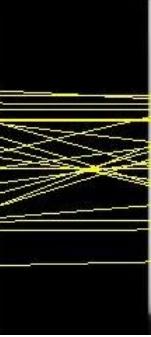
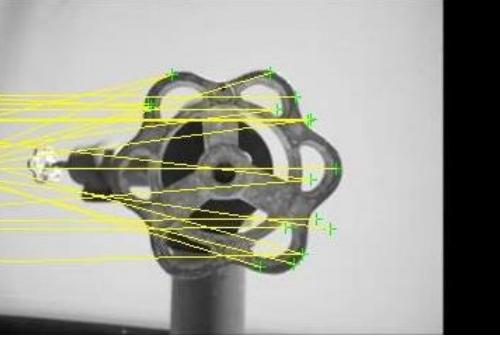
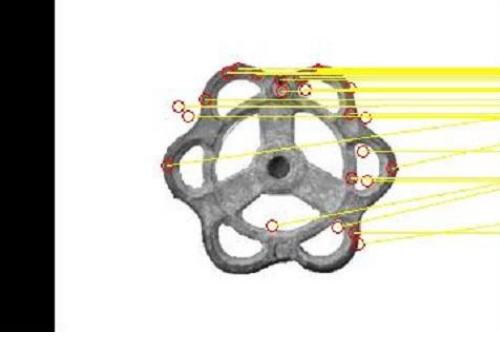
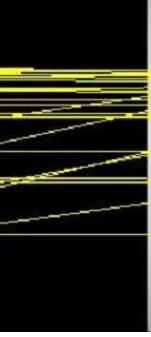
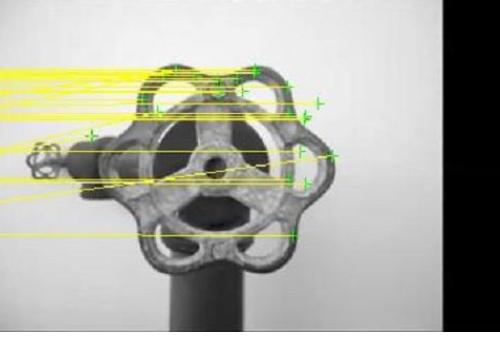
TESTE	CORRELAÇÃO		
7			
8			
9			

Tabela 10: Pontos *Inliers*; coluna “TESTE” conforme Tabela 4. (PARTE 1 – 1/3)

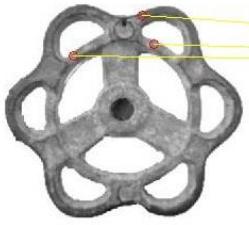
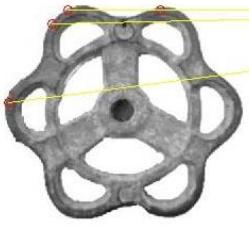
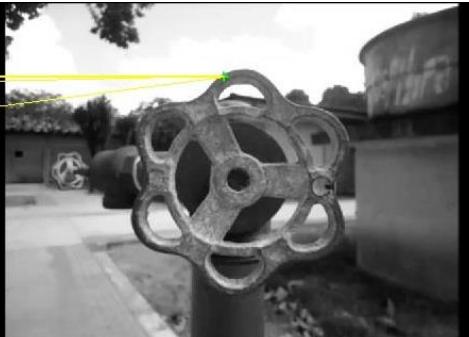
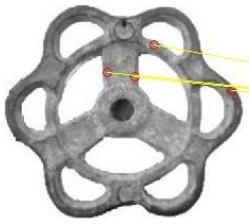
TESTE		INLIERS
1		 
2		 
3		 

Tabela 10: Pontos Inliers; coluna “TESTE” conforme Tabela 4. (PARTE 1 – 1/3)
(CONTINUAÇÃO – 2/3)

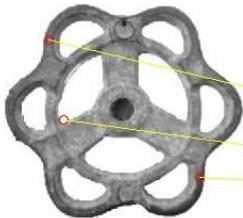
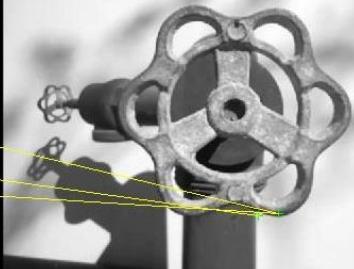
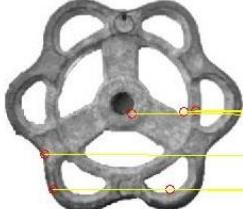
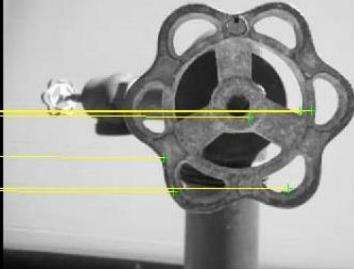
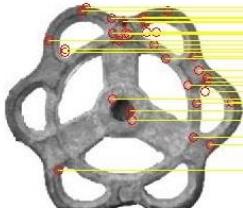
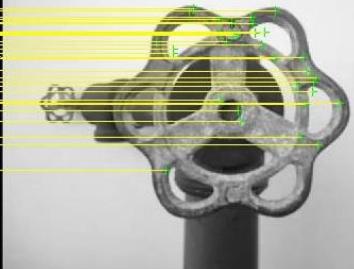
TESTE	INLIERS		
4			
5			
6			

Tabela 10: Pontos *Inliers*; coluna “TESTE” conforme Tabela 4. (PARTE 1 – 1/3) (CONTINUAÇÃO – 3/3)

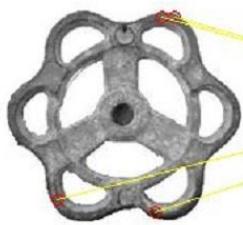
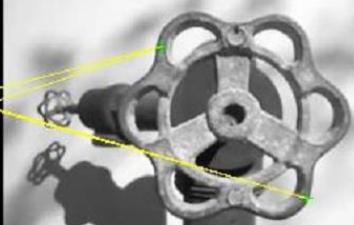
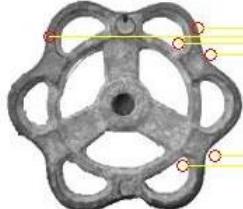
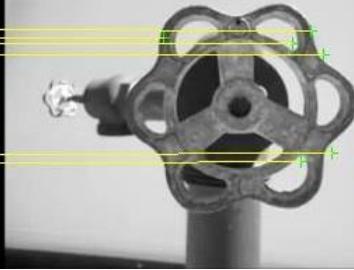
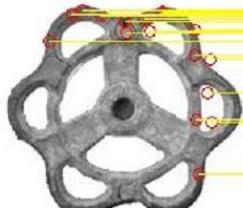
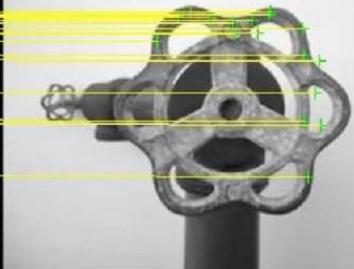
TESTE	INLIERS		
7			
8			
9			

Tabela 11: Pontos rastreados; coluna “TESTE” conforme Tabela 4. (Parte 1 – 1/3)

TESTE	PONTOS RASTREADOS
1	A black and white photograph of a metal valve handle. Six green circles with a white cross inside are overlaid on the image, tracking specific points on the handle's surface.
2	A black and white photograph of a metal valve handle. Five green circles with a white cross inside are overlaid on the image, tracking specific points on the handle's surface.
3	A black and white photograph of a metal valve handle. Seven green circles with a white cross inside are overlaid on the image, tracking specific points on the handle's surface.

Tabela 11: Pontos rastreados; coluna “TESTE” conforme Tabela 4. (Parte 1 – 1/3) (CONTINUAÇÃO – 2/3)

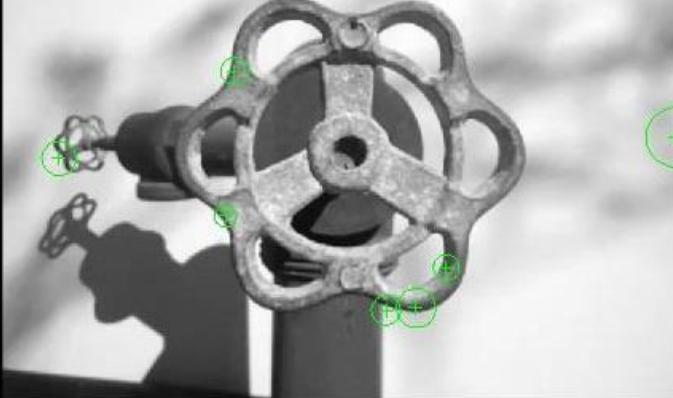
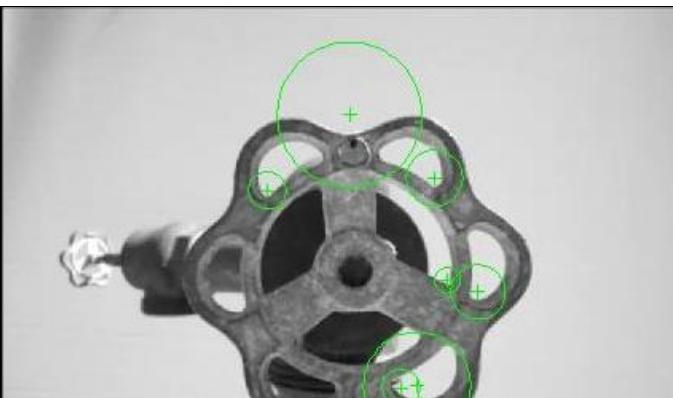
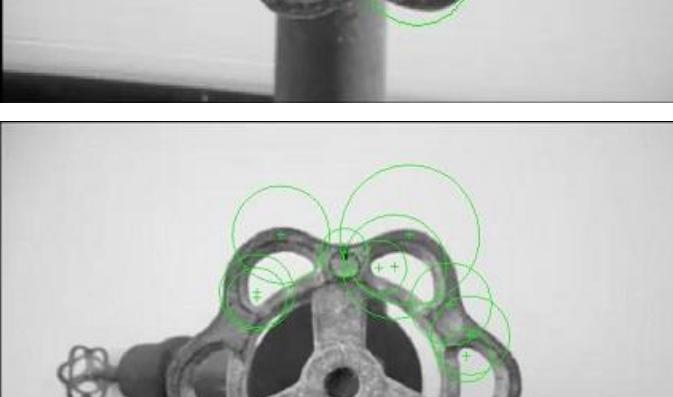
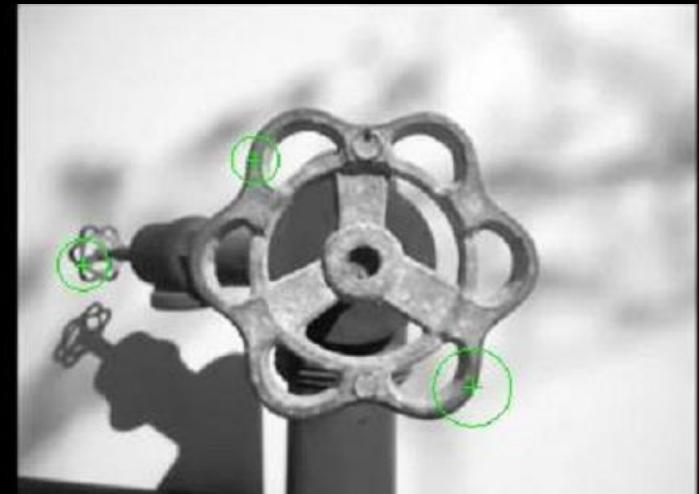
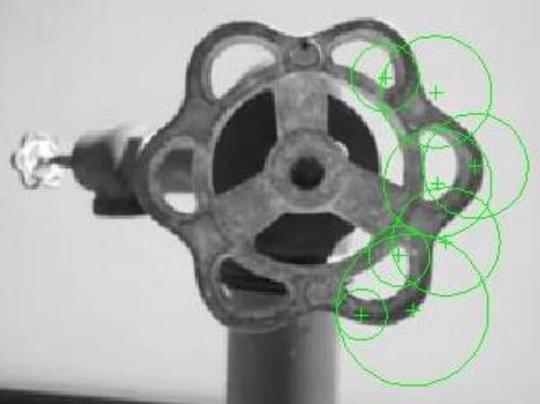
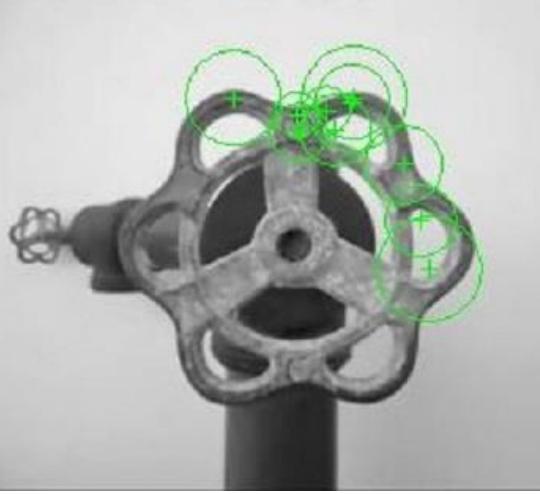
TESTE	PONTOS RASTREADOS
4	 A black and white photograph of a valve handle. Several green circles with a plus sign inside are overlaid on the image, tracking specific points of interest on the handle's body and the stem.
5	 A black and white photograph of a valve handle. Multiple green circles with a plus sign inside are overlaid on the image, tracking points on the handle's body and the stem.
6	 A black and white photograph of a valve handle. A large number of green circles with a plus sign inside are overlaid on the image, tracking points across the entire surface of the handle's body and stem.

Tabela 11: Pontos rastreados; coluna “TESTE” conforme Tabela 4. (Parte 1 – 1/3) (CONTINUAÇÃO – 3/3)

TESTE	PONTOS RASTREADOS
7	
8	
9	

APÊNDICE C - PONTOS CARACTERÍSTICOS AFERIDOS MANUALMENTE

Tabela 12: Pontos característicos aferidos manualmente no primeiro e último quadros capturados no teste; coluna “TESTE” conforme Tabela 4. (PARTE 1 - 1/9)

TESTE	PRIMEIRO QUADRO	ÚLTIMO QUADRO
1	 <p>X: 440 Y: 248 Index: 0.2549 RGB: 0.255, 0.255, 0.255</p> <p>X: 572 Y: 245 Index: 0.3103 RGB: 0.31, 0.31, 0.31</p>	 <p>X: 448 Y: 122 Index: 0.352 RGB: 0.353, 0.353, 0.353</p> <p>X: 440 Y: 247 Index: 0.2419 RGB: 0.239, 0.239, 0.239</p>

Tabela 12: Pontos característicos aferidos manualmente no primeiro e último quadros capturados no teste; coluna “TESTE” conforme Tabela 4. (CONTINUAÇÃO - 2/9)

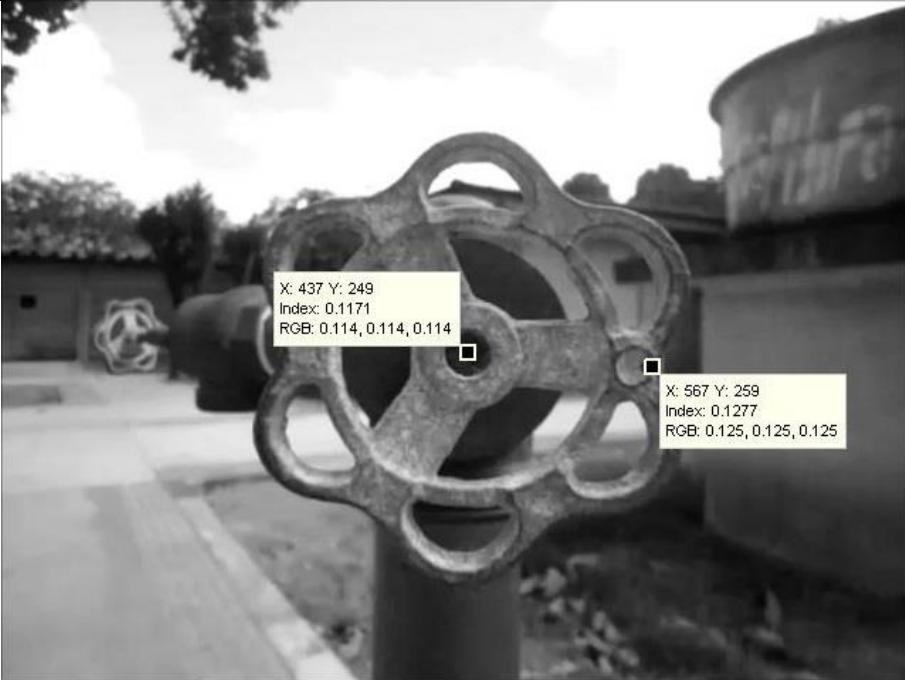
TESTE	PRIMEIRO QUADRO	ÚLTIMO QUADRO
2	 <p>X: 437 Y: 249 Index: 0.1171 RGB: 0.114, 0.114, 0.114</p> <p>X: 567 Y: 259 Index: 0.1277 RGB: 0.125, 0.125, 0.125</p>	 <p>X: 436 Y: 125 Index: 0.1884 RGB: 0.188, 0.188, 0.188</p> <p>X: 438 Y: 249 Index: 0.1292 RGB: 0.129, 0.129, 0.129</p> <p>X: 567 Y: 259 Index: 0.1277 RGB: 0.125, 0.125, 0.125</p>

Tabela 12: Pontos característicos aferidos manualmente no primeiro e último quadros capturados no teste; coluna “TESTE” conforme Tabela 4. (CONTINUAÇÃO - 3/9)

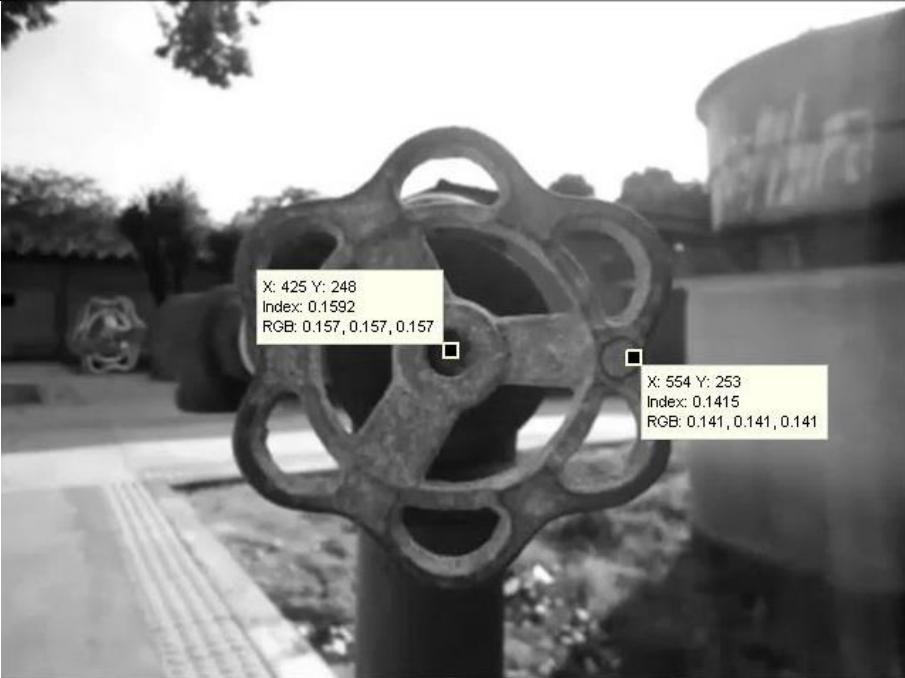
TESTE	PRIMEIRO QUADRO	ÚLTIMO QUADRO
3	 <p>X: 425 Y: 248 Index: 0.1592 RGB: 0.157, 0.157, 0.157</p> <p>X: 554 Y: 253 Index: 0.1415 RGB: 0.141, 0.141, 0.141</p>	 <p>X: 420 Y: 123 Index: 0.2833 RGB: 0.282, 0.282, 0.282</p> <p>X: 426 Y: 248 Index: 0.1883 RGB: 0.188, 0.188, 0.188</p>

Tabela 12: Pontos característicos aferidos manualmente no primeiro e último quadros capturados no teste; coluna “TESTE” conforme Tabela 4. (CONTINUAÇÃO - 4/9)

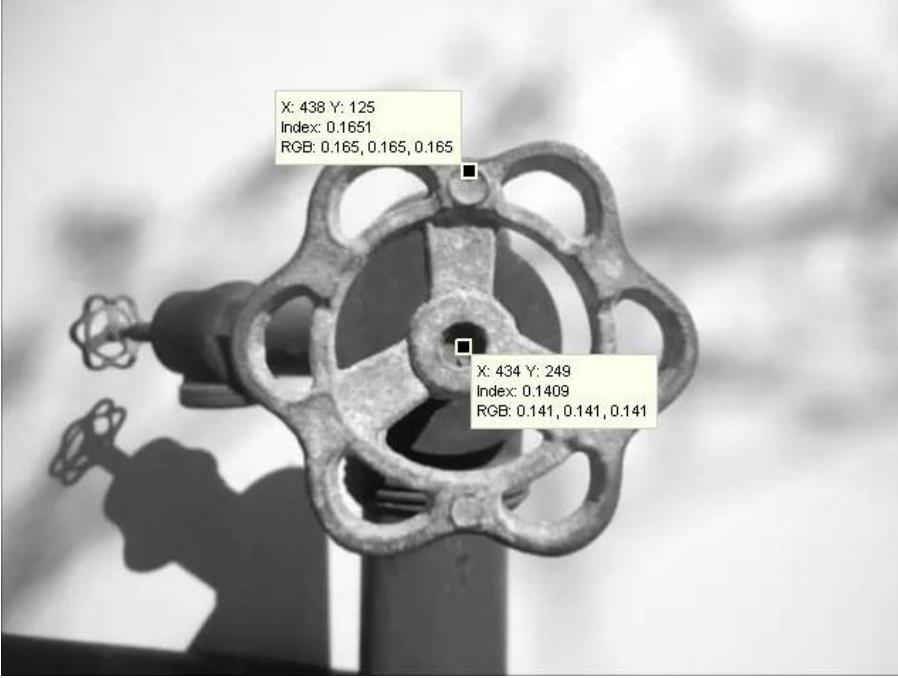
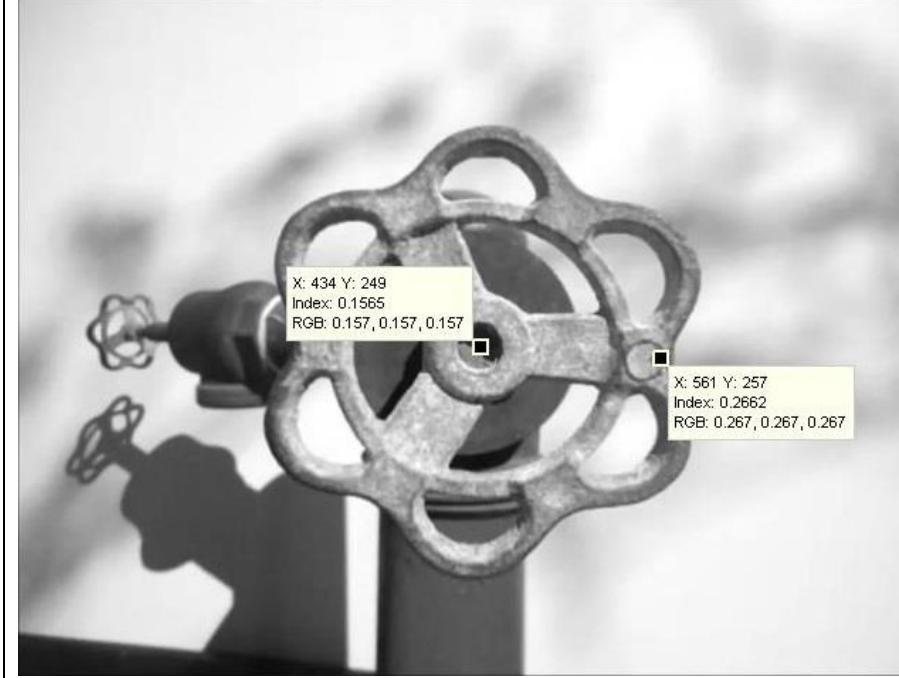
TESTE	PRIMEIRO QUADRO	ÚLTIMO QUADRO
4		

Tabela 12: Pontos característicos aferidos manualmente no primeiro e último quadros capturados no teste; coluna “TESTE” conforme Tabela 4. (CONTINUAÇÃO - 5/9)

TESTE	PRIMEIRO QUADRO	ÚLTIMO QUADRO
5	<p>X: 436 Y: 128 Index: 0.1283 RGB: 0.125, 0.125, 0.125</p> <p>X: 440 Y: 250 Index: 0.1595 RGB: 0.157, 0.157, 0.157</p>	<p>X: 441 Y: 250 Index: 0.16 RGB: 0.157, 0.157, 0.157</p> <p>X: 566 Y: 242 Index: 0.109 RGB: 0.106, 0.106, 0.106</p>

Tabela 12: Pontos característicos aferidos manualmente no primeiro e último quadros capturados no teste; coluna “TESTE” conforme Tabela 4. (CONTINUAÇÃO - 6/9)

TESTE	PRIMEIRO QUADRO	ÚLTIMO QUADRO
6	<p>X: 431 Y: 124 Index: 0.1686 RGB: 0.176, 0.176, 0.176</p> <p>X: 430 Y: 248 Index: 0.26 RGB: 0.271, 0.271, 0.271</p>	<p>X: 428 Y: 248 Index: 0.2424 RGB: 0.255, 0.255, 0.255</p> <p>X: 557 Y: 250 Index: 0.1906 RGB: 0.2, 0.2, 0.2</p>

Tabela 12: Pontos característicos aferidos manualmente no primeiro e último quadros capturados no teste; coluna “TESTE” conforme Tabela 4. (CONTINUAÇÃO -7/9)

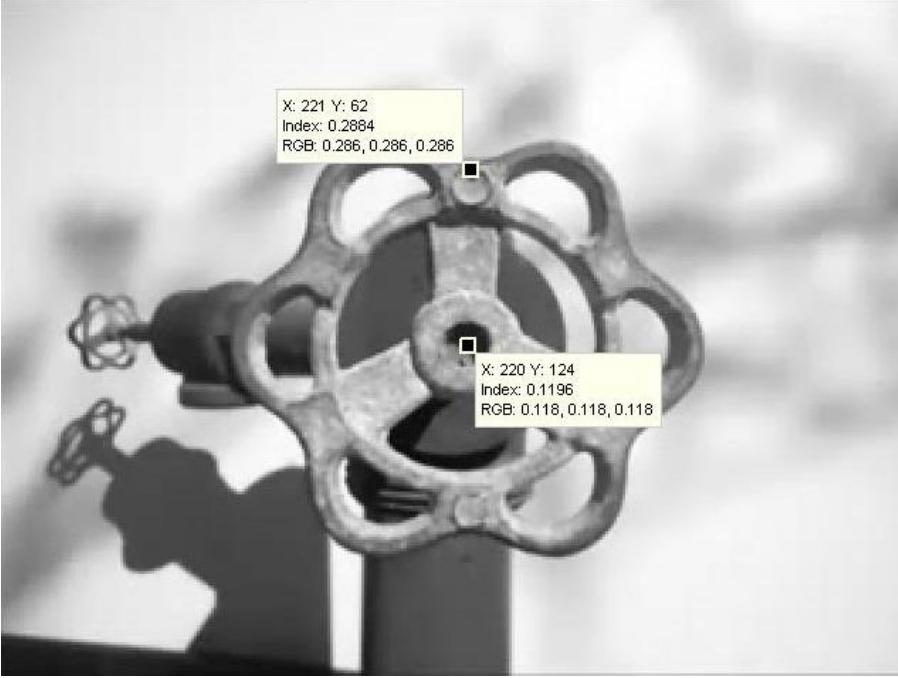
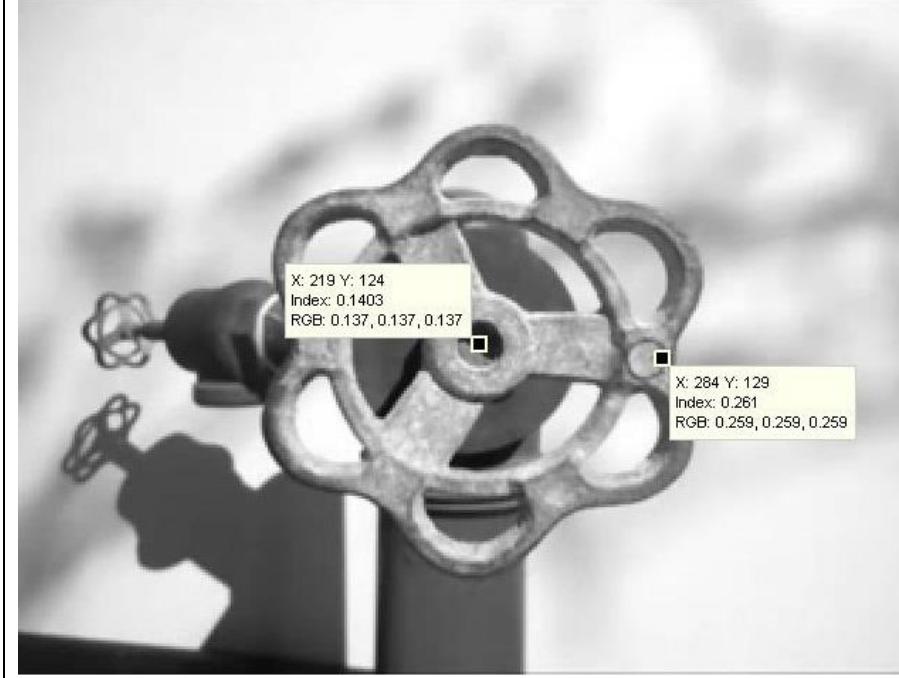
TESTE	PRIMEIRO QUADRO	ÚLTIMO QUADRO
7		

Tabela 12: Pontos característicos aferidos manualmente no primeiro e último quadros capturados no teste; coluna “TESTE” conforme Tabela 4. (CONTINUAÇÃO - 8/9)

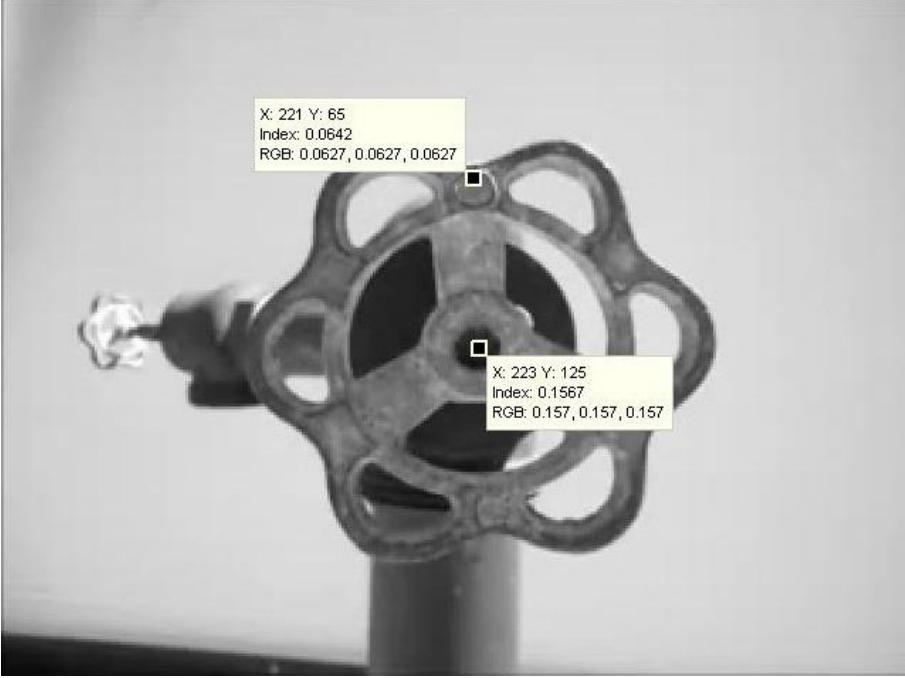
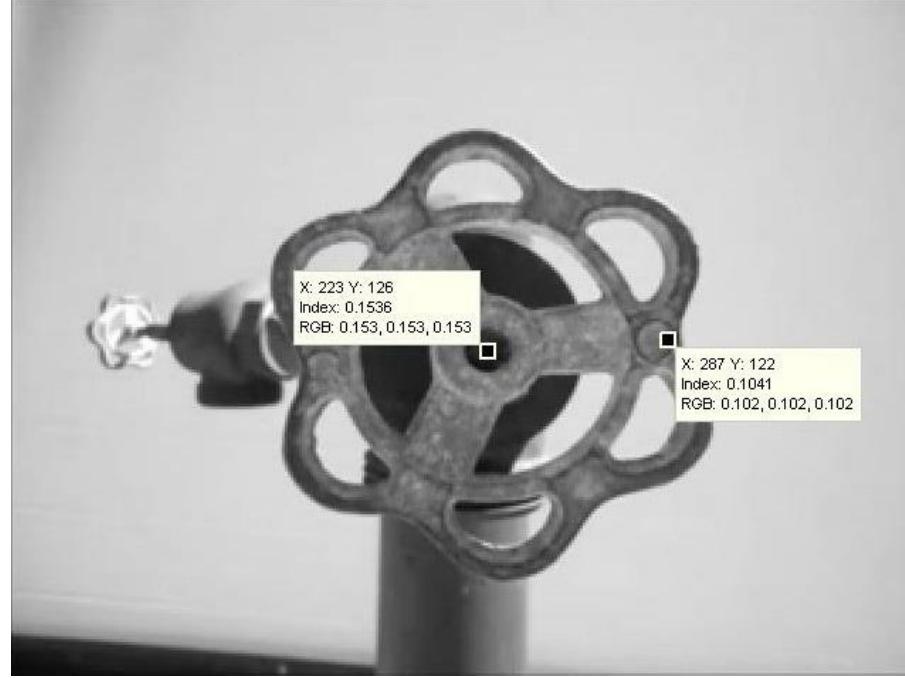
TESTE	PRIMEIRO QUADRO	ÚLTIMO QUADRO
8	 <p>X: 221 Y: 65 Index: 0.0642 RGB: 0.0627, 0.0627, 0.0627</p> <p>X: 223 Y: 125 Index: 0.1567 RGB: 0.157, 0.157, 0.157</p>	 <p>X: 223 Y: 126 Index: 0.1536 RGB: 0.153, 0.153, 0.153</p> <p>X: 287 Y: 122 Index: 0.1041 RGB: 0.102, 0.102, 0.102</p>

Tabela 12: Pontos característicos aferidos manualmente no primeiro e último quadros capturados no teste; coluna “TESTE” conforme Tabela 4. (CONTINUAÇÃO - 9/9)

TESTE	PRIMEIRO QUADRO	ÚLTIMO QUADRO
9	<p>X: 218 Y: 62 Index: 0.1742 RGB: 0.18, 0.18, 0.18</p> <p>X: 217 Y: 124 Index: 0.242 RGB: 0.251, 0.251, 0.251</p>	<p>X: 217 Y: 124 Index: 0.2432 RGB: 0.255, 0.255, 0.255</p> <p>X: 282 Y: 125 Index: 0.2725 RGB: 0.286, 0.286, 0.286</p>