

Um Sistema Genérico de Calibração de Câmera

Clarissa Marques
UFAL
Instituto de Matemática

Adelailson Peixoto (orientador)
UFAL
Instituto de Matemática

Thomas Lewiner
PUC-Rio
Departamento de Matemática

Luiz Velho
IMPA

Resumo

Um processo de calibração de câmera consiste no problema de determinar as características geométricas e ópticas da câmera a partir de um conjunto de dados iniciais. Este problema pode ser dividido em três etapas: aquisição de dados, o processo de calibração em si e otimização. Este trabalho propõe o desenvolvimento de uma ferramenta de calibração baseada em uma arquitetura genérica para qualquer processo de calibração. Para este propósito, o sistema apresentado neste trabalho permite a personalização de cada etapa da calibração. A inclusão de novos métodos de calibração é realizada de forma dinâmica, permitindo assim maior integração e flexibilidade entre os módulos do sistema.

1. Introdução

Os métodos de calibração de câmera consistem em determinar as características geométricas e ópticas internas da câmera assim como sua orientação e posição em relação a um certo sistema de coordenadas do mundo. Este processo geralmente é dado pela resolução de um problema de otimização não-linear supondo ser conhecido um conjunto de pares de pontos 3D, P_i , no sistema de coordenadas da câmera e sua projeção 2D, p_i , no sistema de coordenadas em pixels.

Existem diversos trabalhos e sistemas de calibração disponíveis que geram bons resultados para calibrações. Grande parte destes sistemas realizam a calibração utilizando padrões e suas respectivas imagens, com os métodos do Tsai [5] e Zhang [6]. Porém, nota-se a ausência de um sistema de calibração que permita calibrar vários equipamentos, com a possibilidade de utilizar métodos diferentes para cada um, e realizar uma análise comparativa entre os mesmos com o resultado de cada calibração. A não existência destes sistemas

de calibração adaptativos deu o impulso inicial para a proposta de uma estrutura generalizada e realização deste trabalho.

O trabalho encontra-se assim organizado: na Seção 2 descrevemos o problema de calibração de câmera e as transformações existentes em cada etapa; na Seção 3 enumeramos as etapas necessárias para uma calibração, e com base nessas etapas, propomos uma arquitetura para um sistema genérico de calibração. Procuramos também estender a arquitetura proposta para se adequar a problemas mais genéricos, que possam envolver não apenas elementos de calibração. Na Seção 4 mostramos a implementação do sistema de calibração e exibimos os testes realizados com o sistema e seus resultados. Na Seção 5 finalmente apresentamos as conclusões e recomendações para trabalhos futuros.

2. Calibração de Câmera

A calibração de câmera consiste no processo de determinar dados geométricos e ópticos da câmera, admitindo que sejam conhecidos pares de conjuntos de pontos bidimensionais em uma imagem e seus respectivos pontos tridimensionais. Estes parâmetros podem ser classificados em dois grupos: extrínsecos e intrínsecos. Os parâmetros extrínsecos fornecem informações da posição e orientação da câmera em relação a um certo sistema de coordenadas global (ou do mundo). Os parâmetros intrínsecos nos dão características ópticas e geométricas internas da câmera, que correspondem a distância focal, fatores de escala, posição em pixels da projeção ortogonal do centro óptico no plano de projeção e as distorções proporcionadas pelas lentes.

Desta forma, fica evidente a necessidade de definir relações entre as coordenadas de pontos 3D com as coordenadas 2D de imagens dos mesmos. Existe uma transformação que relaciona estes pontos tridimensionais com os respectivos bidimensionais, exceto por distorções e erros mínimos. Equacionando estas relações com o uso de equações lineares na variável de posição de um objeto, os coeficientes destas equações serão exatamente

funções dos dados que a calibração determina. O processo de calibração de câmera pode se resumir a encontrar tais valores a partir de um conjunto de pontos 3D e 2D correspondentes.

Podemos identificar 4 sistemas de coordenadas, de modo que a correspondência entre esses pontos possa ser expressa como uma composição de transformações que associe os pontos 3D aos 2D correspondentes. Os sistemas de coordenadas considerados são:

Sistema de Coordenadas do Mundo (SCM)

trata-se de um sistema de coordenadas tridimensionais escolhido de forma conveniente a definir as coordenadas de cada ponto da cena. As coordenadas de um ponto P_w neste sistema serão denotadas como (x_w, y_w, z_w) , como ilustra a Figura 1.

Sistema de Coordenadas da Câmera (SCC)

consiste em um sistema de coordenadas com origem no centro óptico da câmera. Os eixos x e y são escolhidos de forma que sejam paralelos ao plano da imagem, e o eixo z sendo o mesmo do eixo óptico. Denotaremos como f a distância entre a origem deste sistema, e o plano de projeção, usualmente denominada distância focal, e (x, y, z) como as coordenadas do ponto P neste sistema. Assim definindo, vemos que a equação do plano de projeção é dada por $z = f$.

Sistema de Coordenadas da Imagem (SCI)

é um sistema de coordenadas bidimensional situado no plano de projeção. A origem deste sistema corresponde à projeção ortogonal do centro óptico no plano de projeção. Iremos denotar as coordenadas de um ponto neste referencial por (X, Y) . Denotaremos também (X_u, Y_u) como as coordenadas da imagem, quando usamos uma câmera do tipo pinhole, isto é, sem as distorções da lente, e (X_d, Y_d) como as coordenadas da imagem com as distorções existentes.

Sistema de Coordenadas em Pixels (SCP)

também consiste de um sistema de coordenadas bidimensional. As coordenadas de um ponto da imagem neste sistema são expressas em pixels. Usualmente, toma-se como centro deste sistema o canto superior esquerdo da imagem. As coordenadas de um ponto neste referencial será denotada por (X_f, Y_f) .

A transformação que associa um ponto do sistema de coordenadas do mundo ao sistema de coordenadas em pixels pode ser representada fazendo uma composição das transformações descritas entre os sistemas de coordenadas citados acima. O resultado final obtido encontra-se na equação 1.

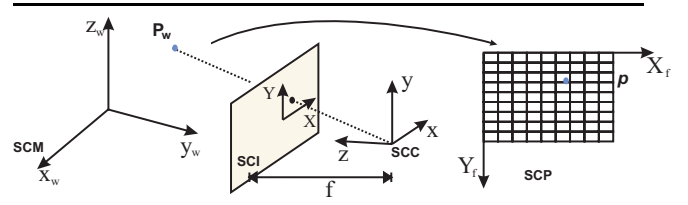


Figura 1. Sistemas de Coordenadas na calibração.

$$[p] \simeq \begin{bmatrix} f s_x & f \tau & C_x \\ 0 & f s_y & C_y \\ 0 & 0 & 1 \end{bmatrix} [R \ T][P]. \quad (1)$$

A matriz $K = \begin{bmatrix} f s_x & f \tau & C_x \\ 0 & f s_y & C_y \\ 0 & 0 & 1 \end{bmatrix}$ é chamada de

matriz intrínseca de calibração e agrupa todos os parâmetros intrínsecos da câmera.

f - distância focal;

s_x e s_y - correspondem ao número de pixels por unidade de comprimento em ambas direções, horizontal e vertical, respectivamente;

τ - dada pela tangente do ângulo que as colunas da matriz de pixels formam com a perpendicular com as linhas;

C_x e C_y - representam as coordenadas da projeção ortogonal do centro óptico no plano de projeção;

R e T - a matriz de rotação e o vetor de translação, respectivamente, que compõe o movimento rígido que leva o SCM (Sistema de Coordenadas do Mundo) no SCC (Sistema de Coordenadas da Câmera).

Note que a equação 1 não é inversível. Ora,

$$\bar{p} = K^{-1}[p] \simeq [R \ T][P].$$

O ponto $\bar{p} = (\bar{x}, \bar{y}, 1)$ corresponde à imagem do ponto P tomando a matriz de calibração como a identidade. Os pontos que possuem p como projeção seriam da forma $\lambda(\bar{x}, \bar{y}, 1)$, onde $\lambda \in \mathbb{R}$. As coordenadas de \bar{p} são conhecidas como *coordenadas normalizadas do ponto p* da imagem.

A resolução desse sistema de equações, conhecendo os conjuntos de pontos p_i e P_i , é o problema central da calibração, e por isso existem diversos métodos para encontrar tais soluções, tais como Tsai [5], Zhang [6], entre outros. De um modo geral o processo de calibração pode ser descrito da seguinte forma:

1. Determinar um conjunto de pontos no sistema de coordenadas do mundo e suas respectivas projeções nas imagens;
2. Resolver as equações lineares;
3. Otimizar os dados obtidos, utilizando o resultado da etapa anterior como solução inicial.

3. Arquitetura Proposta

Uma das dificuldades de se projetar um sistema genérico de calibração de câmera está na variedade dos dados de entrada que cada método requer, principalmente na especificação e na correspondência entre estes dados. Outro obstáculo reside na forma como estes dados são transformados durante o processo de calibração. Esta seção apresenta a modelagem de um sistema genérico de calibração. Para tal é necessário analisar o problema de uma forma global e identificar quais etapas estão envolvidas num processo de calibração, identificando os elementos necessários em cada etapa.

3.1. Etapas Envolvidas

Um processo padrão de calibração de câmera, como vimos anteriormente, pode ser estruturado da seguinte forma:

1. Aquisição dos dados (especificação e correspondência);
2. Calibração (resolução de equações lineares);
3. Otimização dos dados obtidos, utilizando o resultado da etapa anterior como solução inicial.

Pelas etapas acima, podemos dizer que a calibração de uma câmera se resume no processo de obter os seus parâmetros, a partir de um conjunto de dados iniciais, conforme mostra a Figura 2.

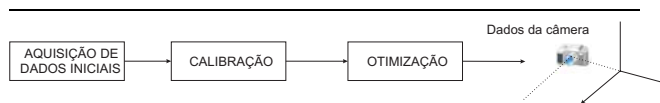


Figura 2. Etapas da Calibração de Câmera.

3.1.1. Etapa de Aquisição de Dados Iniciais Esta etapa está altamente relacionada com a especificação do usuário, a partir da qual o conjunto de dados de entrada será definido. Esses dados, em geral, são composto por um conjunto de pares de pontos 2D e 3D.

Usualmente o conjunto de pontos 3D correspondem a pontos de um padrão de calibração. Diversos métodos utilizam padrões cujas coordenadas de seus pontos sejam relativamente fáceis de se calcular. São vários os padrões utilizados para calibração de câmera, por exemplo o padrão

xadrez Figura 3(a), o mais popularmente conhecido e utilizado, devido à simplicidade de obter as coordenadas de seus pontos. Outro exemplo clássico é o de retângulos encaixados [3], [4] que se interceptam como mostra a Figura 3(b). A Figura 3(c) mostra um padrão tridimensional.

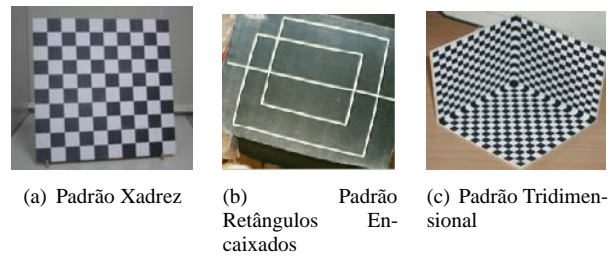


Figura 3. Padrões.

O conjunto de pontos 2D é extraído de uma imagem do padrão adquirida pelo equipamento a ser calibrado. As imagens são utilizadas para fazer a correspondência entre os pares de pontos 2D e 3D, e dependendo do método podem ser utilizadas várias imagens tiradas de ângulos diferentes ou mesmo apenas uma. A partir do padrão e das imagens precisamos de algum método para obter a correspondência entre os pontos adquiridos que serão utilizados na calibração. Nesta etapa de aquisição de dados iniciais definimos os seguintes elementos:

- equipamento: corresponde ao equipamento que será calibrado (câmera ou projetor);
- padrão: corresponde ao padrão a partir do qual os pontos 3D serão obtidos;
- imagem: corresponde às imagens do padrão a partir das quais os pontos 2D serão obtidos;
- métodos de correspondência: indicam como os pontos 2D e 3D serão especificados e como eles se relacionam.

3.1.2. Etapa de Calibração Dados um conjunto de pontos 2D e o correspondente conjunto de pontos 3D esta etapa tem como finalidade resolver um sistema de equações lineares que apontam uma solução inicial para os parâmetros intrínsecos e extrínsecos do equipamento a ser calibrado.

Existem várias heurísticas para resolver esta etapa, que define diferentes métodos como o Tsai [5], Zhang [6], etc. Assim o elemento a ser definido é o método de calibração.

3.1.3. Etapa de Otimização Todos os elementos citados anteriormente possuem alta influência na precisão do resultado obtido pelo método de calibração. Isto implica que nem sempre os resultados são satisfatórios. Uma solução para este problema consiste em utilizar uma otimização para melhorar o resultado obtido. O método de otimização

mais utilizado é o algoritmo de Levenberg-Marquadt [2], [1], que provê uma solução numérica para o problema de minimização de uma função de reprojeção dos pontos, esta geralmente não linear. Com isto, definimos o último elemento que pode fazer parte de um sistema de calibração.

A Figura 4 mostra as etapas de calibração detalhando os elementos envolvidos.

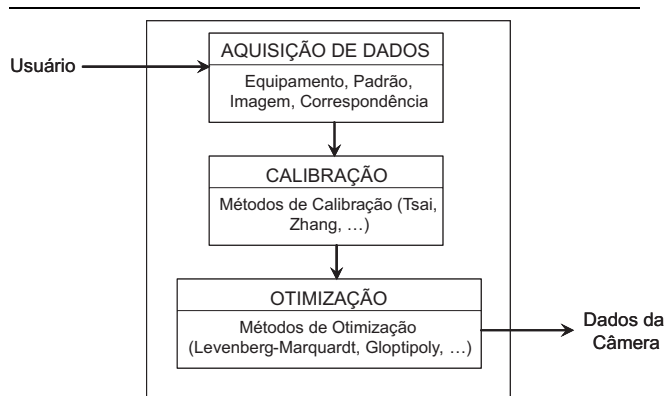


Figura 4. Processo de Calibração de Câmera.

3.2. Arquitetura

Para desenvolver um sistema adaptável a diferentes métodos de calibração, foi proposta uma arquitetura com todos os elementos possivelmente necessários a um pipeline padrão de calibração. A modelagem desta arquitetura encontra-se na Figura 5.

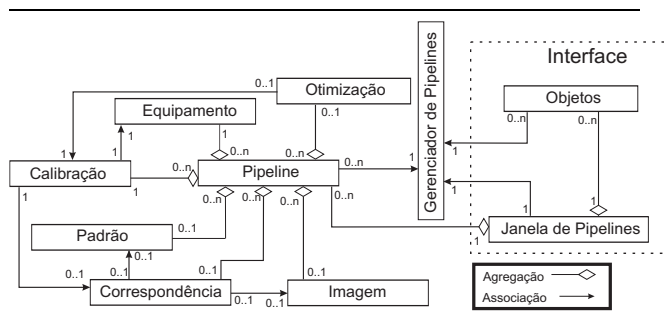


Figura 5. Modelagem do Sistema.

Cada elemento das etapas envolvidas corresponde a um módulo do sistema. Os relacionamentos envolvidos são basicamente de *associação*, que mostra o fluxo de dados entre

os elementos, e a *agregação*, que mostra a relação de pertinência entre os elementos.

O elemento *padrão* e o elemento *imagem*, por exemplo, estão associados ao elemento *correspondência*, uma vez que os métodos de correspondência utiliza os dados do padrão e da imagem para obter os pontos 2D e 3D. O elemento *correspondência*, por sua vez, fornece os pontos 2D e 3D ao elemento *calibração*. Todos estes elementos estão agregados ao elemento *pipeline*.

Portanto, a relação entre o elemento pipeline e os elementos que determinam um pipeline, conforme Figura 5, pode ser definido como uma relação de agregação. Ou seja, todos estes elementos fazem parte do elemento pipeline.

O Gerenciador de Pipelines têm como principal finalidade adicionar (ou editar) a um pipeline os elementos que serão agregados a ele. A interface gráfica do sistema pode ser definida a partir dos elementos *objeto* e *janela de pipelines*. O elemento *objeto* são retângulos que serão exibidos na janela de pipelines e representam quaisquer elementos (equipamento, padrão, imagem, etc.). O elemento *janela de pipelines* é a janela principal onde os pipelines serão exibidos.

3.3. Extensão da Arquitetura

A arquitetura proposta na seção anterior foi modelada para as etapas básicas de calibração. Atualmente, esta arquitetura foi estendida para integrar ao sistema pipelines mais complexos, onde talvez contenham operações que não sejam necessariamente de calibração. Esta nova modelagem permite então a execução de pipelines muito mais genéricos, com as mais diversas funcionalidades possíveis.

Para isto a arquitetura foi alterada de forma que cada elemento que compõe o pipeline seja uma subclasse da classe abstrata *Objetos*. Para integrar os métodos correspondentes a cada objeto esta subclasse deverá ser implementada em uma biblioteca dinâmica, a qual deverá ser chamada ao pipeline ser executado.

A classe *Objetos*, já existente na arquitetura anterior foi modificada de forma que armazene todas as informações necessárias do objeto, ou seja as informações necessárias para executar o método, e os dados para a sua visualização na janela. Os dados necessários para a execução do objeto são por exemplo os tipos e dados de entrada e saída, e sua quantidade, e o diretório da dll correspondente. E os dados para sua visualização na janela corresponde as dimensões e posição do objeto na janela.

Conhecendo os tipos de dados de entrada e saída, o sistema poderá fazer restrições impossibilitando a conexão entre elementos com dados incompatíveis.

A Figura 6 abaixo ilustra a nova arquitetura implementada.

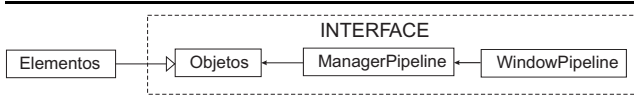


Figura 6. Arquitetura estendida.

Os *Elementos* correspondem aos novos objetos que serão integrados ao sistema. Como mostra a Figura 6 todos esses objetos devem ser subclasses da classe mãe *Objetos*. A classe *ManagerPipeline* tem por finalidade administrar a inclusão, edição e exclusão de objetos no pipeline. Devido a isto ela se relaciona com a classe *WindowPipeline* que representa a janela de visualização dos pipelines. Para um processo usual de calibração, a arquitetura anterior pode se adaptar a esta como ilustra na Figura 7 abaixo.

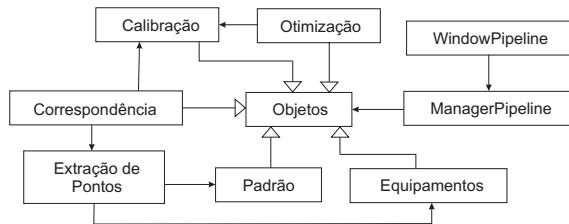


Figura 7. Arquitetura estendida para a calibração.

A classe *Imagem* existente na arquitetura deixou de existir e tornou-se um dado de entrada da classe *Equipamentos*. A execução dos pipelines é realizada de forma sequencial, e implementada no formato “data-pull”. Isto é, os dados de entrada de um objeto são “solicitados” para o seu antecessor, e o objeto apenas será executado se tiver recebido todos os seus dados de entrada.

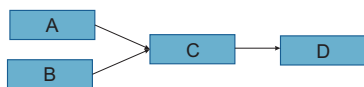


Figura 8. Processo Data-Pull.

Ao executar o pipeline ilustrado na Figura 8, o objeto D deverá pedir seus dados de entrada para o objeto C. Porém, o C só poderá ser executado com os dados de entrada recebidos do objeto A e B. Então, C deverá chamar o objeto A que por não ter anterior deverá ser executado e passará seu resultado ao objeto C. Em seguida, o mesmo acontecerá com o objeto B. Por não ter antecessor, deverá ser executado e retornará seu resultado para o objeto C. Como o objeto C já contém todos seus dados de entrada, poderá ser

executado e por sua vez, dará os resultados para o objeto D que enfim será executado.

4. Resultados

Para a implementação desta ferramenta para a arquitetura básica, foram desenvolvidos dois módulos denominados *EditCalib* e *ExecCalib*. O módulo *EditCalib* é responsável pela edição dos pipelines de calibração. Este módulo permite a personalização de cada etapa do processo de calibração, com a possibilidade de incluir novos métodos de correspondência, calibração e otimização. A Figura 9 ilustra este módulo.

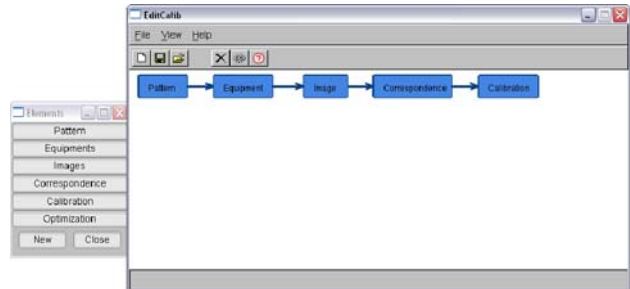


Figura 9. EditCalib.

O módulo *ExecCalib* tem como funcionalidade executar os pipelines gerados no módulo anterior, com a possibilidade de alterar os dados de cada elemento do pipeline. A sua interface encontra-se ilustrada na Figura 10.

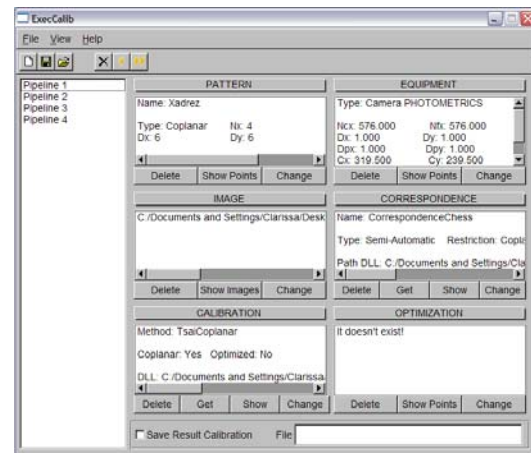


Figura 10. ExecCalib.

4.1. Arquitetura

A arquitetura básica proposta permite que quaisquer elementos sejam combinados em um pipeline e seu resultado seja os parâmetros extrínsecos e intrínsecos do equipamento calibrado. A coerência entre os elementos combinados deve ser mantida pelo usuário de forma que o resultado seja o esperado.

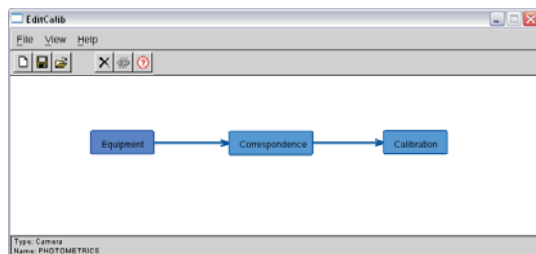


Figura 11. Calibração não-convencional.

O pipeline construído na Figura 11 pode obter os resultados desejados, apesar de se tratar um pipeline de calibração não-convencional, ou seja, não segue a seqüência usual de um pipeline de calibração. Desde que ou o elemento de calibração ou o de correspondência tome seus dados de entrada em um arquivo, o resultado final da execução deste pipeline pode ser o esperado. Por exemplo, podemos tomar o método de calibração definido no pipeline acima como o Tsai Coplanar, que apenas precisa do conjunto de pares de pontos 2D e 3D, e algumas especificações do equipamento. Assim, a semântica do pipeline da Figura 11 ficará correta.

É possível ainda a combinação entre elementos de pipelines diferentes. Neste caso, os pipelines antigos deverão ser reformulados de forma que a coerência nos mesmos permaneça. Para exemplificar considere os pipelines da Figura 12. Os pipelines válidos para o sistema serão os identificados na ilustração.

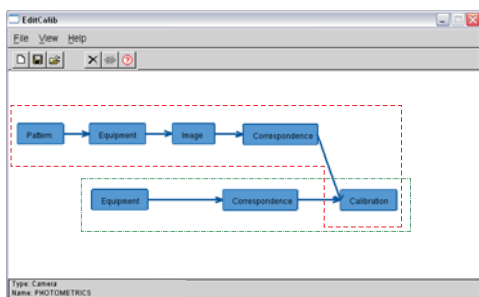


Figura 12. Interpretando pipelines.

Ao seleccionar um elemento suas informações serão ex-

ibidas na barra da janela como mostra a Figura 12.

4.2. Módulo EditCalib

O foco deste módulo consiste na personalização de cada elemento presente em um pipeline.

Inclusão de Elementos

A inclusão de novos padrões, equipamentos e imagens é feita naturalmente apenas informando dados necessários para a identificação dos mesmos. A Figura 13 ilustra a inclusão de um novo tipo de padrão.

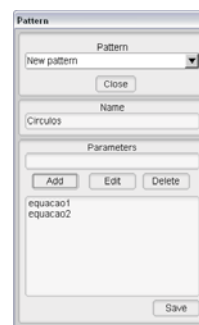


Figura 13. Inclusão de elementos estáticos.

A integração de novos métodos de correspondência, calibração e otimização não ocorre de forma tão natural, visto que seria necessária a inclusão do código do método no sistema. Este problema foi solucionado com a implementação de subclasses da classe relativa ao método, por exemplo métodos de correspondência, calibração ou otimização, através do uso de bibliotecas dinâmicas (dll's), visando proteger o sistema e torná-lo mais fácil e flexível para o usuário. Um exemplo de inclusão de um novo método de calibração encontra-se ilustrado na Figura 14.

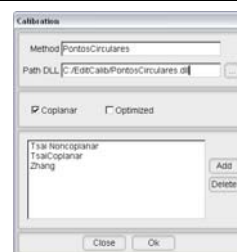


Figura 14. Inclusão de um novo método de calibração.

Edição de Elementos

Outra funcionalidade disponível é a alteração de elementos contidos em um pipeline. Para alterar as informações de um elemento, basta dar um clique duplo sobre o elemento para que seja exibida a janela com as informações do mesmo para que possam ser editadas. Um exemplo desta operação encontra-se na Figura 15.

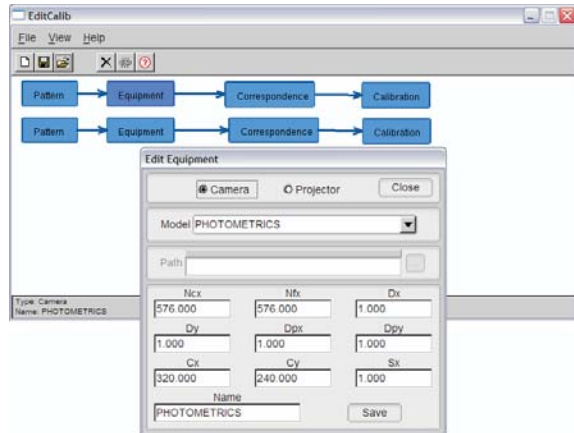


Figura 15. Alterar um elemento.

4.3. Módulo ExecCalib

Um dos testes realizados no sistema foi a calibração de uma câmera Cyber-Shot modelo DSC-P93A, utilizando o método de Tsai-Coplanar e um padrão Xadrez. As Figuras 16(a) e 16(b) mostram a etapa do método de correspondência durante a execução do pipeline no sistema.

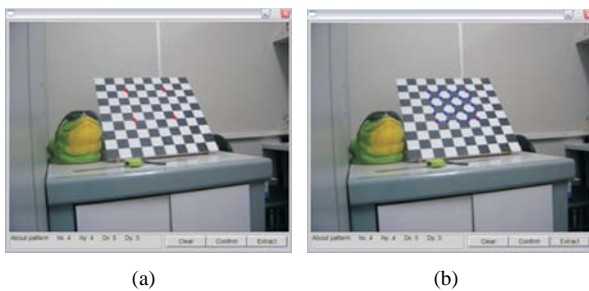


Figura 16. Correspondência na Calibração da Câmera.

Com os pontos obtidos na correspondência, é realizado a calibração e seu resultado pode ser visto na Figura 17.

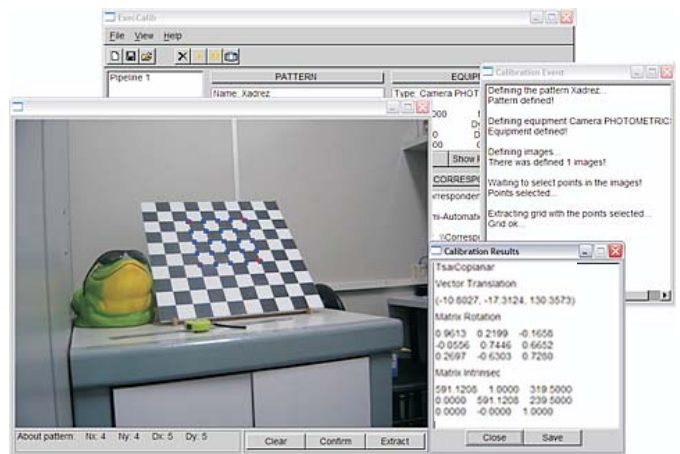


Figura 17. Calibração da Câmera.

Também calibramos um projetor utilizando os dados da calibração da câmera citada acima. As imagens 18(a) e 18(b) abaixo ilustram a correspondência entre os pontos durante a calibração do projetor.

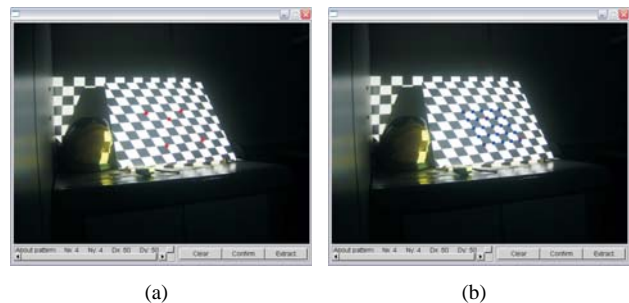


Figura 18. Correspondência na Calibração do Projetor.

O resultado da calibração do projetor encontra-se na imagem 19 abaixo.

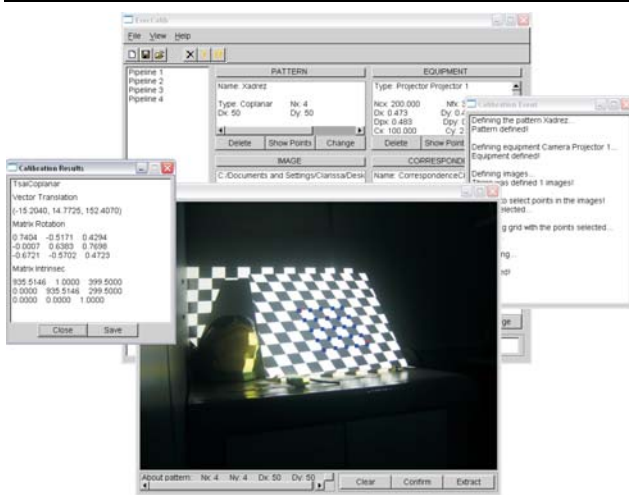


Figura 19. Calibração do Projetor.

Para testar a escalabilidade da ferramenta, construímos pipelines não-usuais para um processo de calibração. A Figura 20 mostra testes realizados com combinações entre elementos de diferentes pipelines.

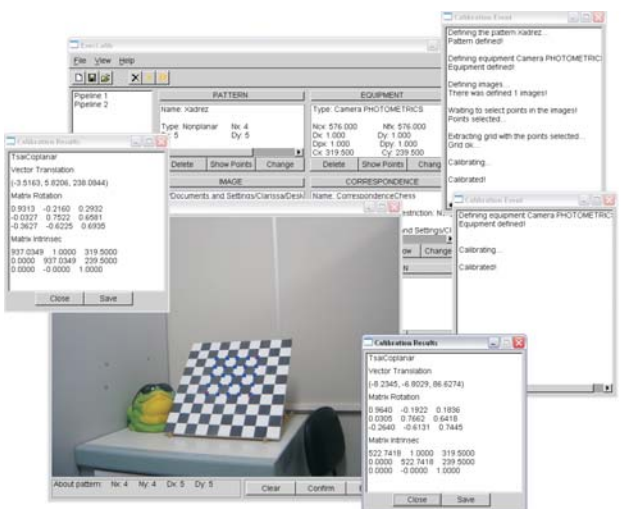


Figura 20. Execução de pipelines não-convencionais.

A Figura 21 mostra o resultado da execução de dois pipelines que tomam os dados de entrada da etapa de calibração em um arquivo. Neste último o método de calibração usado no pipeline 1 foi o Tsai Coplanar, e no pipeline 2 o Tsai Não-Coplanar.

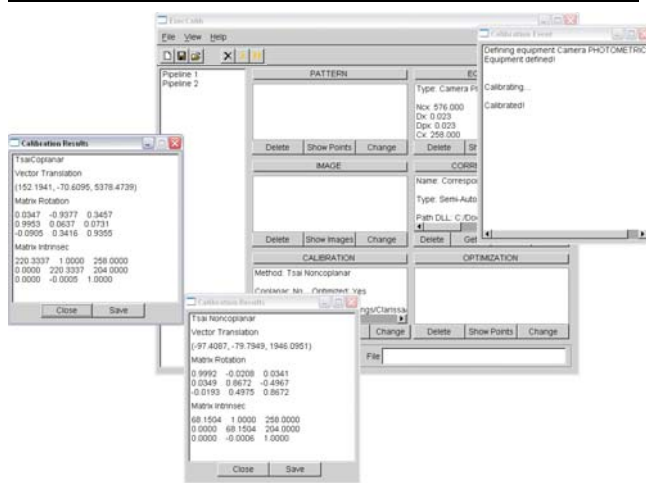


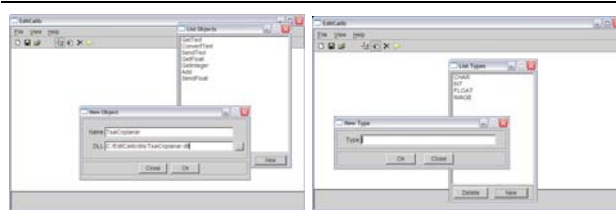
Figura 21. Execução de vários pipelines.

4.4. Arquitetura Extendida

Modificamos o módulo EditCalib para que se torna-se responsável também pela execução dos pipelines, além da criação e edição dos mesmos. O método de incorporar novos objetos ao sistema é idêntica aos módulos implementados para a arquitetura básica.

Inclusão de Elementos e Tipos de Dados

A cada subclasse a ser integrada ao sistema deve-se informar o nome e o diretório da dll em que foi implementada. A imagem 22(a) mostra a inserção de novos objetos como por exemplo o método de calibração de Tsai. Além de novos objetos, também podem ser adicionados novos tipos de dados de entrada e saída permitindo a adaptação do sistema a diferentes problemas. A Figura 22(b) ilustra como adicionar novos tipos.



(a) Inclusão de novo objeto.

(b) Inclusão de novos tipos.

Restrições

Conhecendo as entradas e saídas de cada objeto, o sistema faz restrições de forma a permitir apenas que se conectem

objetos que tenha pelo menos um dado compatível. Isto é, um dos tipos de saída de um objeto seja um dos tipos de entrada do objeto da outra extremidade.

A Figura 22 abaixo ilustra dois objetos *GetWord*, cujo objetivo apenas é obter um texto dado pelo usuário, e o objeto *AddNumbers*, que realiza a soma entre dois números reais. Logo, a conexão entre eles não é válida, visto que o objeto *AddNumbers* não possui nenhuma entrada do tipo “CHAR”.

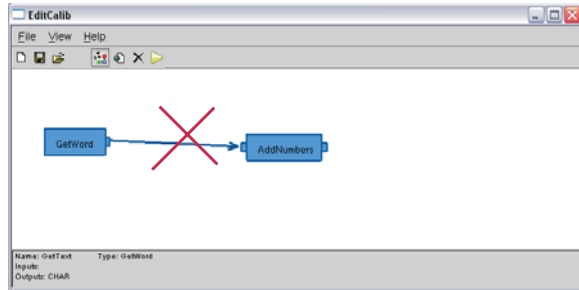
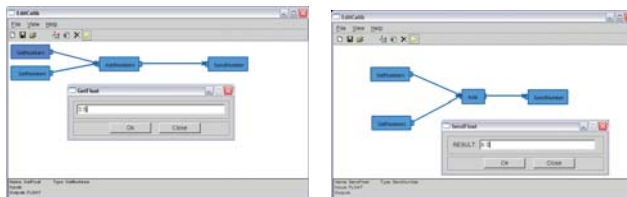


Figura 22. Conexão não permitida.

Testes

As Figuras 23(a) e 23(b) mostram um simples exemplo da execução com a nova arquitetura. Temos dois objetos do tipo *GetNumber*, que deverá tomar cada um números reais, um objeto do tipo *Add*, que irá somar tais número e retornar o resultado para o último objeto, do tipo *SendNumber*, que irá exibir o resultado para o usuário.



(a) Número real

(b) Resultado

Figura 23. Exemplos.

Um exemplo de pipeline de calibração diferente a ser elaborado consiste em antes de realizar a calibração em si, fazer filtragens sobre imagens que venham a ser utilizadas na calibração. Um simples pipeline para filtrar imagens pode ser visto na Figura 24 abaixo.

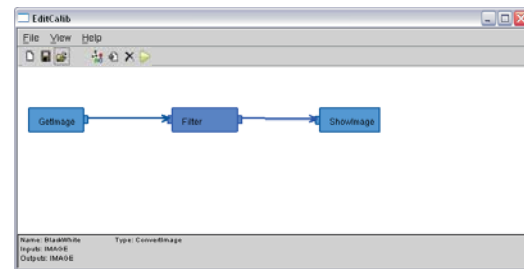


Figura 24. Filtragem de imagens.

5. Conclusões e Trabalhos Futuros

Este trabalho propôs uma arquitetura genérica onde fosse possível implementar diversos métodos de calibração de câmera. Baseado na arquitetura proposta o desenvolvimento do sistema teve como principal objetivo dar maior flexibilidade ao usuário para criar diferentes formas de calibração e assim analisar e comparar os resultados. Foram desenvolvidos dois módulos para a edição e execução dos pipelines de calibração na arquitetura básica.

A adição de novos métodos de calibração, correspondência e otimização através de bibliotecas dinâmicas (Dynamically Linked Library) de suas subclasses tornou o sistema mais compacto, simples e flexível. Posteriormente, a arquitetura foi estendida de forma que abrangesse operações e elementos fora do contexto da calibração. A diversidade de elementos e de métodos que podem ser aplicados à ferramenta tornam o sistema mais versátil e útil para as mais diversas aplicações.

Como trabalhos futuros propostos para este trabalho podemos citar: a inclusão de novos métodos de calibração e de otimização, permitindo assim uma análise numérica e a comparação da convergência com outros métodos, a implementação de um ambiente 3D que simule a reconstrução dos equipamentos calibrados, a implementação de testes de projeção dos pontos 3D do padrão virtual, utilizando uma câmera virtual com os mesmos dados da câmera real calibrada. Em seguida, comparar estas projeções com as fotografias originais dos padrões que foram tiradas com a câmera real.

Referências

- [1] M. I. A. Lourakis. A brief description of the levenberg-marquardt algorithm implemented by levmar. 2005.
- [2] A. Ranganathan. The levenberg-marquardt algorithm. 2004.
- [3] F. Szenberg. *Acompanhamento de Cenas com Calibração Automática de Câmeras*. PhD thesis, Pontifícia Universidade Católica do Rio de Janeiro, 2001.
- [4] F. Szenberg, P. Carvalho, and M. Gattass. Automatic camera calibration for image sequences of a football match, advances in pattern recognition. pages 301–311, 2001.

- [5] R. Y. Tsai. A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE Journal Of Robotics And Automation*, RA-3(4):323–344, 1987.
- [6] Z. Zhang. A flexible new technique for camera calibration. Technical report, Microsoft Corporation, 1998.