

Projeto AGE – PETROBRAS

Sistema de Visão Computacional AGE – VisAGE

1. Introdução

Utilizando apenas uma câmera acoplada à Garra Magnética – GAMAG – do robô KUKA KR 500, o sistema VisAGE é capaz de identificar alvos do projeto AGE-PETROBRAS e posicionar o robô de modo que seu efetuador fique alinhado com o centro do alvo. Após o posicionamento e alinhamento da ferramenta, o programa através de um comando do usuário controla o robô para aproximar seu efetuador da chapa e pegá-la, ativando a GAMAG.

Feito isto, aliado ao Módulo de Visão Estéreo que possui uma visão geral do processo, é possível posicionar e alinhar a chapa recém pegada para sua posição definitiva na montagem da estrutura.

2. Especificações de Hardware e Software

O Sistema VisAGE está sendo desenvolvido em um computador com processador Intel(R) Core(TM) i7-3770 CPU @ 3.40GHz (8 CPUs), com 32768MB de memória RAM e Sistema Operacional GNU/Linux Debian Stable (Wheezy). A câmera acoplada ao efetuador do robô KUKA KR 500 é, no momento, uma câmera Microsoft HD 720p.

A biblioteca de Visão Computacional OpenCV 2.4.9 e a biblioteca PugiXML para interpretar e montar mensagens no formato XML estão sendo utilizadas no desenvolvimento do software, com as linguagens de programação C e C++, compilador GCC 4.7 e a ferramenta CMake para configuração de montagem do projeto.

3. Arquitetura e detalhes do Sistema

O Sistema de Visão Computacional VisAGE foi dividido em cinco módulos: comunicação com o robô KUKA através do protocolo RSI, calibração mono e estéreo das câmeras, visão da câmera acoplada ao efetuador robótico, visão estéreo das câmeras que possuem uma visão geral da montagem da estrutura e o módulo principal que integra e coordena os demais módulos.

Os módulos de calibração e visão estéreo estão fora do escopo deste relatório. O diagrama abaixo representa a arquitetura do sistema:

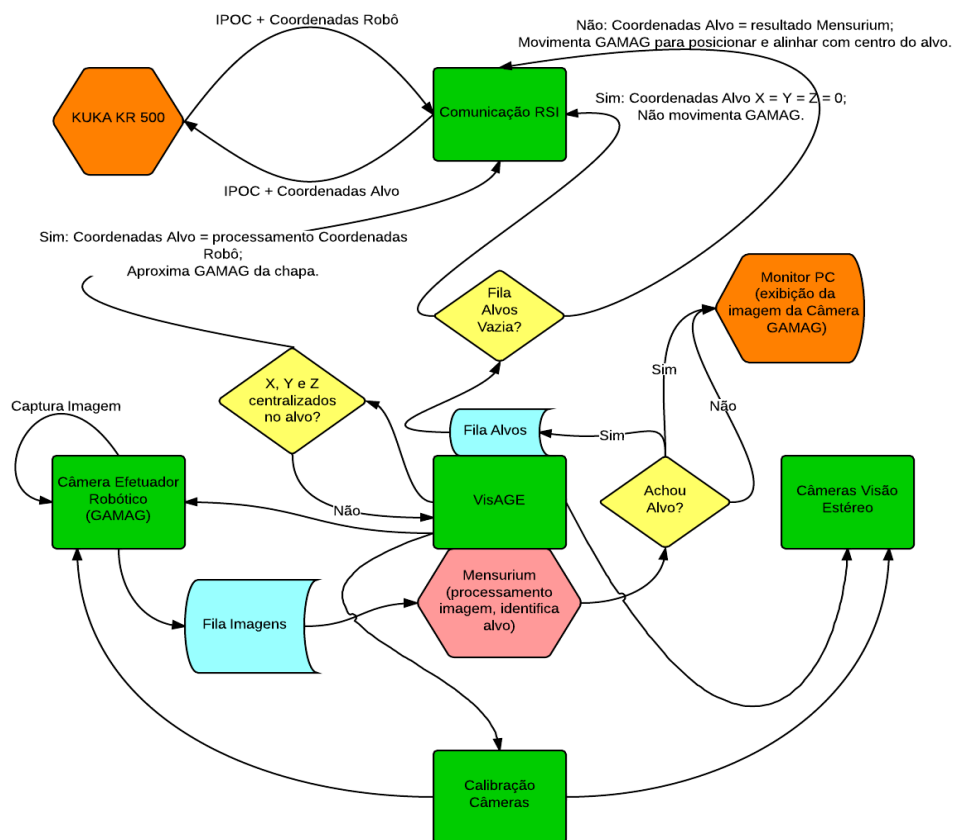


Figura 1: Diagrama Arquitetura Sistema VisAGE

A comunicação com o robô KUKA através do protocolo RSI é de natureza de tempo real. Nesta comunicação, o robô faz papel de cliente e o módulo de comunicação RSI do sistema VisAGE faz papel de servidor. O cliente envia um código denominado IPOC, que é um *timestamp*, além de enviar outras informações do robô em uma mensagem no formato XML via UDP. O manual do fabricante contém detalhes sobre esta mensagem e seus campos, como mostra a Figura 2:

```
<Rob TYPE="KUKA">
  <RIst X="1620.0008" Y="0.0000" Z="1910.0000" A="0.0000" _
    B="90.0000" C="0.0000"/>
  <RSol X="1620.0000" Y="0.0000" Z="1910.0000" A="0.0000" _
    B="90.0000" C="0.0000"/>
  <AIPos A1="0.0000" A2="-90.0000" A3="90.0000" A4="0.0000" _
    A5="0.0000" A6="0.0000"/>
  <AIPos A1="0.0000" A2="-90.0000" A3="90.0000" A4="0.0000" _
    A5="0.0000" A6="0.0000"/>
  <ASPos A1="0.0000" A2="-90.0000" A3="90.0000" A4="0.0000" _
    A5="0.0000" A6="0.0000"/>
  <EIPos E1="0.0000" E2="0.0000" E3="0.0000" E4="0.0000" _
    E5="0.0000" E6="0.0000"/>
  <ESPos E1="0.0000" E2="0.0000" E3="0.0000" E4="0.0000" _
    E5="0.0000" E6="0.0000"/>
  <MACur A1="0.0000" A2="0.0000" A3="0.0000" A4="0.0000" _
    A5="0.0000" A6="0.0000"/>
  <MECur E1="0.0000" E2="0.0000" E3="0.0000" E4="0.0000" _
    E5="0.0000" E6="0.0000"/>
  <Delay D="0" />
  <Tech C11="0.000000" C12="0.000000" C13="0.000000" C14="0.000000" _
    C15="0.000000" C16="0.000000" C17="0.000000" C18="0.000000" _
    C19="1.000000" C110="0.000000" />
  <DiL>0</DiL>
  <Digout o1="0" o2="0" o3="0" />
  <ST_SOURCE>17.232147</ST_SOURCE>
  <IPOC>4208163634</IPOC>
</Rob>
```

Figura 2: Protocolo RSI - Mensagem Cliente

O servidor ao receber esta mensagem tem 12 milissegundos para enviar a resposta com o IPOC, caso contrário a comunicação com o robô é perdida. O módulo de visão da câmera do efetuador robótico ao detectar o alvo, processa a imagem e insere as novas coordenadas do robô que irão posicionar e alinhar o efetuador de acordo com o centro do alvo numa fila. O servidor verifica esta fila imediatamente após receber um IPOC, e caso não esteja vazia, envia as novas coordenadas junto com o último IPOC recebido, movimentando o efetuador robótico em direção ao centro do alvo. Caso a fila esteja vazia, pode ser que a imagem ainda não tenha sido processada (este processamento é superior a 12ms) ou nenhum alvo tenha sido detectado. Neste caso, os parâmetros X, Y e Z da *tag Rkorr* (Figura) são enviados com valor 0 e o último IPOC recebido também é reenviado. Desta forma, a comunicação com o robô fica assíncrona e independente da captura e processamento de imagens provenientes das câmeras.

O manual do fabricante contém detalhes sobre esta mensagem de resposta e seus campos, como mostra a Figura:

```
<Sen Type="ImFree">
  <EStr>ERX Message! Free config!</EStr>
  <RKorr X="0.0000" Y="0.0000" Z="0.0000" A="0.0000" B="0.0000" _
    C="0.0000" />
  <AKorr A1="0.0000" A2="0.0000" A3="0.0000" A4="0.0000" _
    A5="0.0000" A6="0.0000" />
  <EKorr E1="0.0000" E2="0.0000" E3="0.0000" E4="0.0000" _
    E5="0.0000" E6="0.0000" />
  <Tech T21="1.09" T22="2.08" T23="3.07" T24="4.06" T25="5.05" _
    T26="6.04" T27="7.03" T28="8.02" T29="9.01" T210="10.00" />
  <DiO>125</DiO>
  <IPOC></IPOC>
</Sen>
```

Figura 3: Protocolo RSI - Mensagem Servidor

Assim que as coordenadas X, Y e Z estiverem alinhadas com o centro do alvo, o operador pode acionar um comando para aproximar a garra magnética da chapa. Vale lembrar que a câmera não fica acoplada no centro da garra, então o centro a ser calculado é o centro da ferramenta, e não da imagem capturada pela câmera. Também está sendo desenvolvido o movimento de rotação (parâmetros A, B e C da *tag Rkorr*, na mensagem de resposta), para que o robô seja capaz de rotacionar o efetuador em direção ao centro do alvo. Por enquanto, apenas os movimentos de translação e aproximação do

efetuador robótico foram implementados.

4. Bibliografia

- KUKA.RobotSensorInterface 2.3 For KUKA System Software 5.4, 5.5, 5.6, 7.0
- Open Source Computer Vision (OpenCV):
<http://opencv.org/>
- Cross-Platform Make (Cmake):
<http://www.cmake.org/>
- PugiXML:
<http://pugixml.org/>