GR-PEACH and MATLAB operations –

Image differences – two jpeg (.jpg) image files. MATLAB using R2014A forward

Example – WebCamera –possible applications:
Motion detection,
Earthquake monitoring (or Camera vibration)
Intruder detection, alarm function

Steganography*, finding hidden messages in original source vs. modified version)

*Can be independent of WebCamera – SURF section can be skipped for better performance

-MATLAB change detection, using general or WebCamera pictures - here is Scale Invariant difference of two similar pictures –

This would be best case results –

File : difference.m (for MatLab) is complex code and Sqrt, Arctangent in SURF algorithms to do jpg to rgb absolute differences – this would not be fast on GR-PEACH. SURF is used as its 3-5x faster than SIFT.

```
// MATLAB CODE to take 2 jpgs and determine the scale invariant (SURF
algorithm used) absolute difference (like xor )
//
tic;
[FileName_f, PathName_f] = uigetfile('*.jpg','Select the fixed image');
fixed = imread([PathName_f, FileName_f]);

[FileName_m, PathName_m] = uigetfile('*.jpg', 'Select the moving image');
moving = imread([PathName_m, FileName_m]);

out_path = uigetdir;
```

```
// convert jpg to gray scale

fixed_g = rgb2gray(fixed);
moving_g = rgb2gray(moving);



movingAdjusted=imadjust(moving_g);
fixedAdjusted=imadjust(fixed_g);


// perform scale invariant feature transform via faster SURF vs. older SIFT

ptsFixed=detectSURFFeatures(fixedAdjusted);

ptsMoving=detectSURFFeatures(movingAdjusted);


[featuresFixed,validPtsFixed]=extractFeatures(fixedAdjusted,ptsFixed);

[featuresMoving,validPtsMoving]=extractFeatures

indexPairs = matchFeatures(featuresFixed,featuresMoving);

matchedFixed=validPtsFixed(indexPairs(:,1));

matchedMoving=validPtsMoving(indexPairs(:,2));


// Get estimates geometric transform
[tform,inlierMoving,inlierFixed]=estimateGeometricTransform(...
matchedMoving,matchedFixed,'affine');

Tinv=tform.invert.T;

ss=Tinv(2,1);

sc=Tinv(1,1);
```

```
// recover scale and angle
scaleRecovered=sqrt(ss*ss + sc*sc);

thetaRecovered=atan2(ss,sc)*180/pi;


movingRegistered=imwarp(moving,tform,'OutputView',imref2d(size(fixed)));


// perform absolute difference matrix calculation - NoN GPU version
diff = imabsdiff(fixed, movingRegistered);

imshow(diff)

cd(out_path);

imwrite(diff, 'difference.jpg', 'jpg');

timeSpent=toc;


clc;
```

Speeded up robust features - Wikipedia, the free encyclopedia
https://en.wikipedia.org/wiki/Speeded_up_robust_features

Wikipedia
In computer vision, **Speeded Up Robust Features** (**SURF**) is a local feature detector and descriptor that can be used for tasks such as object recognition or registration or classification or 3D reconstruction. It is partly inspired by the **scale-invariant feature transform** (**SIFT**) descriptor.
Overview - Algorithm and features - See also - References
Image Processing: What is the difference between SURF and SIFT ...
https://www.quora.com/Image-Processing-What-is-the-difference-between-**SURF**-
...

Quora
Nov 28, 2014 - **SIFT**(**Scale-invariant feature transform**), Lowe, 2004. ... **SURF**(**Speeded Up Robust Features**), Bay, 2006. ... **SIFT** uses Difference of Gaussian over various scales of an image and is the most accurate feature detector and descriptor.

Introduction to SURF (Speeded-Up Robust Features) — OpenCV ...
opencv-python-tutroals.readthedocs.org/en/latest/py...**surf**.../py_**surf**_intro.html

In **SIFT**, Lowe approximated Laplacian of Gaussian with Difference of Gaussian for finding scale-space.**SURF** goes a little further and approximates LoG with ...

[PDF]A Comparison of SIFT and SURF - IJIRCCE
www.ijircce.com/upload/2013/april/21_V1204057_A%20Comparison_H.pdf

A Comparison of **SIFT** and **SURF**. P M Panchal1, S R Panchal2, S K Shah3. PG Student, Department of Electronics & Communication Engineering, SVIT, ...

SIFT & SURF - YouTube

▶ **5:18**

https://www.youtube.com/watch?v=Z3OVMm1knwo

Apr 12, 2013 - Uploaded by Miquel Perelló Nieto
I am not the author of this program. More information in this link: https://code.google.com/p/find-object/

algorithm - SURF vs SIFT, is SURF really faster? - Stack Overflow
stackoverflow.com/questions/11172408/**surf**-vs-**sift**-is-**surf**-really-faster

Jun 23, 2012 - When it was designed it was intended to be faster, but actually the differences are not relevant for real time applications with standard cameras.

[PDF]SURF: Speeded Up Robust Features - Computer Vision Laboratory
www.vision.ee.ethz.ch/~**surf**/eccv06.pdf

ETH Zurich
by H Bay - Cited by 6905 - Related articles
interest point detector and descriptor, coined **SURF** (Speeded Up Ro- ..... ures 4 and 7, we compared our **SURF** descriptor to GLOH, **SIFT** and PCA-**SIFT**,.

[PDF]A Comparison of SIFT, PCA-SIFT and SURF - CiteSeerX
citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.301.7041&rep=rep1..

by L Juan - Cited by 617 - Related articles
International Journal of Image Processing (IJIP) Volume(3), Issue(4). 143. A Comparison of **SIFT**, PCA-**SIFT** and **SURF**. Luo Juan qiuhehappy@hotmail.com.
computer vision - What are some free alternatives to SIFT/ SURF that ...
dsp.stackexchange.com/.../what-are-some-free-alternatives-to-**sift-surf**-that-can-be-use...

Jan 31, 2012 - As far as I understand, both **SURF** and **SIFT** are patent protected. Are there any alternative methods that can be used in a commercial ...
[PDF]Study group SURF: Feature detection & description - Department of ...
cs.au.dk/~jtp/**SURF**/report.pdf

Aarhus University
Abstract—A technical report on Feature detection and implementing the Speeded-Up Robust Features(SURF) algorithm. Index Terms—**SURF**, **SIFT**, feature ...

Thank you for the opportunity to support your organization. Very respectfully,
Xtrm Designs LLC

References:

PC/Server with GPU

Absolute difference of two images - MATLAB imabsdiff - MathWorks
www.mathworks.com/help/images/ref/im**absdiff**.html

MathWorks
gpuarrayZ = **imabsdiff**(gpuarrayX , gpuarrayY) performs the computation on a GPU,
if at least one of the inputs is a gpuArray. The output image is a gpuArray.

Imabsdiff
gpuarrayZ = **imabsdiff**(gpuarrayX , gpuarrayY) performs the ...
More results from mathworks.com »

Getting Started with Embedded Coder Support Package for ARM Cortex-A Processors

Open This Example

This example shows you how to use Embedded Coder Support Package for ARM Cortex-A Processors to run a Simulink® models on an ARM Cortex-A9 emulator.

Introduction

Embedded Coder Support Package for ARM Cortex-A Processors enables you to create and run Simulink models on a QEMU emulator. You can verify the generated code on an emulated ARM Cortex-A9 processor without actual hardware. You can also collect execution time measurements for an algorithm implemented in Simulink to aid refining and optimizing your algorithm.

http://www.mathworks.com/help/supportpkg/armcortexa/examples/getting-started-with-arm-cortex-a-support-package.html

Code Verification and Validation with PIL and External Mode

This example shows you how to use Embedded Coder Support Package for ARM Cortex-A Processors for code verification and validation using PIL and External Mode.

Introduction

In this example you will learn how to configure a Simulink model to run Processor-In-the-Loop (PIL) and External Mode simulations. In a PIL simulation, the generated code runs on a QEMU ARM Cortex-A9 emulator. The results of the PIL simulation are transferred to Simulink to verify the numerical equivalence of the simulation and the code generation results. The PIL verification process is a crucial part of the design cycle to ensure that the behavior of the deployment code matches the design.

Simulink's External mode feature enables you to accelerate the process of parameter tuning by letting you change certain parameter values while the model is running on target hardware, without stopping the model. When you change parameter values from within Simulink, the modified parameter values are communicated to the target hardware immediately. The effects of the parameters tuning activity may be monitored by viewing algorithm signals on scopes or displays in Simulink.

This example introduces the Simulink code generation and verification workflow by showing you how to:

Configure a Simulink model to run PIL simulations on the ARM Cortex-A9 (QEMU) emulator target

Configure a Simulink model to run External Mode simulation on ARM Cortex-A9 (QEMU) emulator target

Prerequisites

We recommend completing Getting Started with ARM Cortex-A Support Package.

You should start QEMU emulator for ARM Cortex-A by executing the following command in MATLAB® command prompt, if not already done:

```
pid = codertarget.arm_cortex_a.runQemu()
```

http://www.mathworks.com/help/supportpkg/armcortexa/examples/code-verification-and-validation-with-pil-and-external-mode.html

Embedded Coder Support Package for ARM Cortex-A Processors

Generate code optimized for Cortex A processors

Examples

Functions and Other Reference

Release Notes

Embedded Coder® Support Package for ARM® Cortex®-A Processors lets you generate NEON optimized code for math operations using the Ne10 Library. Use this generated code for ARM Cortex-A processors. For DSP filter support, use the ARM Cortex A Ne10 Library Support from DSP System Toolbox™.

Setup and Configuration

Install hardware support

Modeling

Prepare model for hardware connection, add blocks to support hardware protocols

Deployment

Build the model that loads and runs an executable on the target hardware.

Performance
Improve simulation speed, increase accuracy, reduce data loss.

Develop a Target
Create a target based on the ARM Cortex-A processor

http://www.mathworks.com/help/supportpkg/armcortexa/index.html

MATLAB to C Made Easy
Generating readable and portable C code from your MATLAB algorithms by Bill Chou

Agenda

Motivation
–
Why translate MATLAB to C?
–
Challenges of manual translation

Using MATLAB Coder
–
Three-step workflow for generating code

Use cases
–
Integrate algorithms using source code/libraries
–
Accelerate through MEX
–
Prototype by generating EXE

Conclusion
–
Integration with Simulink and Embedded Coder
–
Other deployment solutions