Stereo Projection

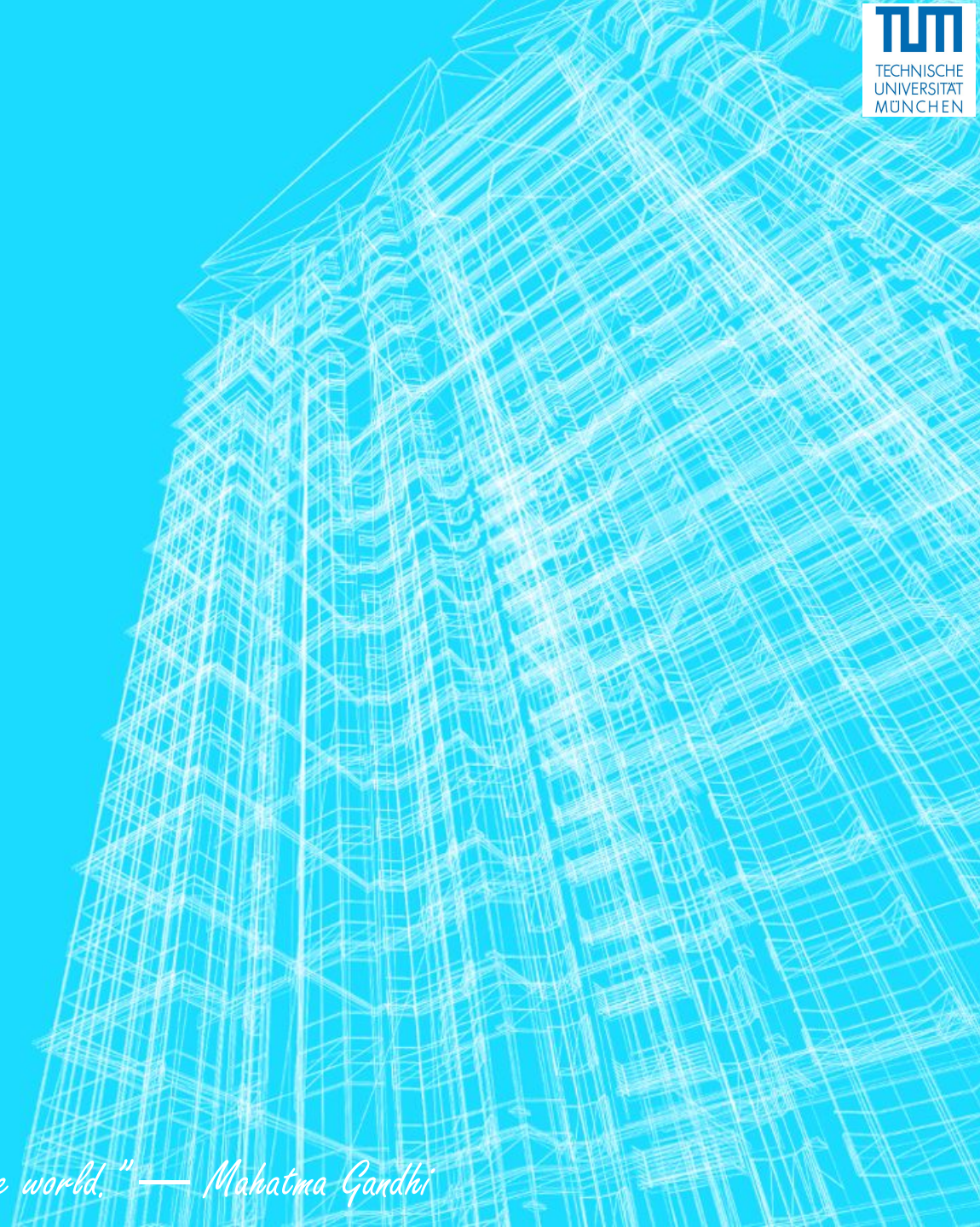
Using Primal-Dual Variational Method

Course :: GPU Programming in Computer Vision (WS2013/14)

Team :: *Warp64* (group 9)

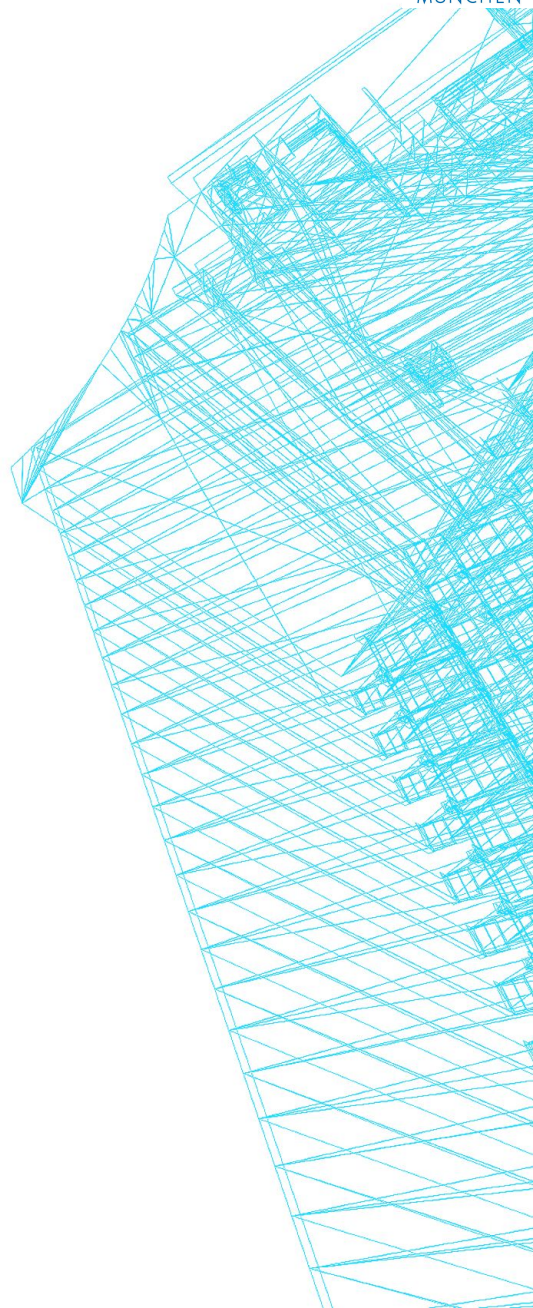
Adviser :: Mohamed S.

Members :: Baishya S. (03649393) Debnath S. (03649492)and Karthik A. ()



"You must be the change you wish to see in the world." — Mahatma Gandhi

TECHNISCHE UNIVERSITÄT MÜNCHEN



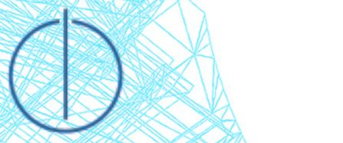
Introduction

".. You can't connect the dots looking forward; you can only connect them looking backwards.. " — Steve Jobs

Aim

- Implementation of a Computer Vision algorithm with the goal focused on performance measure and comparing memory types.

- And a little bit of theory..

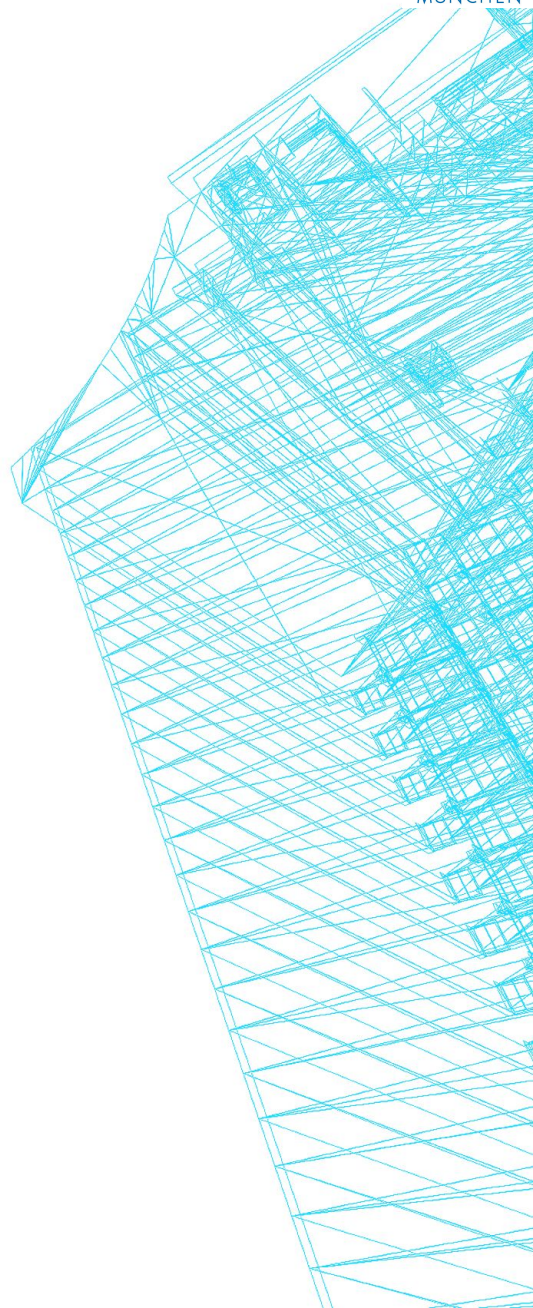


TECHNISCHE UNIVERSITÄT MÜNCHEN

Contents

- Stereo Projection
  - Example
  - Theory
  - Implementation
  - Result and Performance
  - GUI and Demo

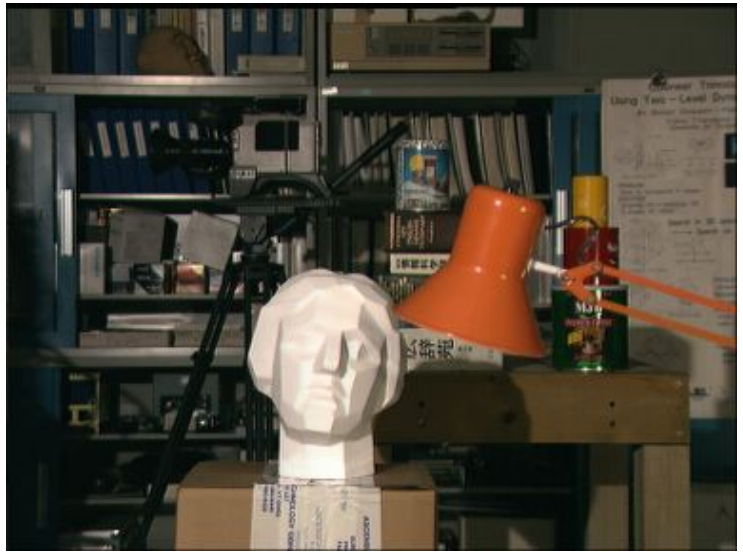
TECHNISCHE UNIVERSITÄT MÜNCHEN



EXAMPLES

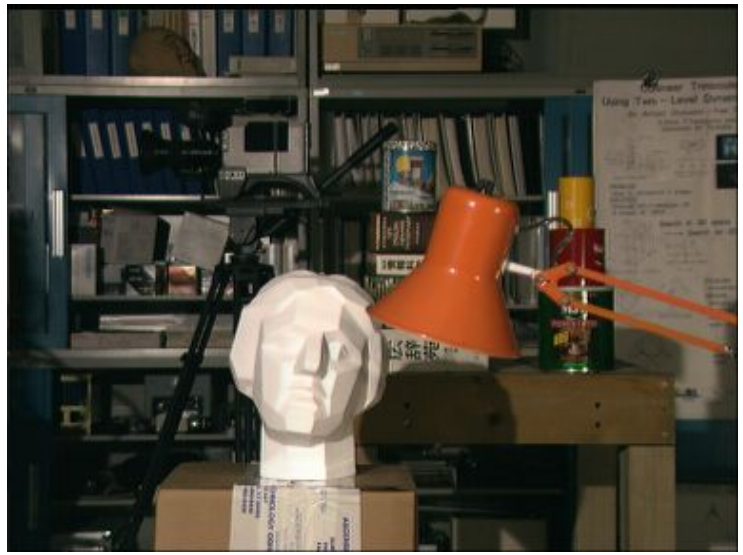
"We are apt to forget that children watch examples better than they listen to preaching." — Roy L. Smith

Example 1: STATUE IMAGE



Left Image

Depth image

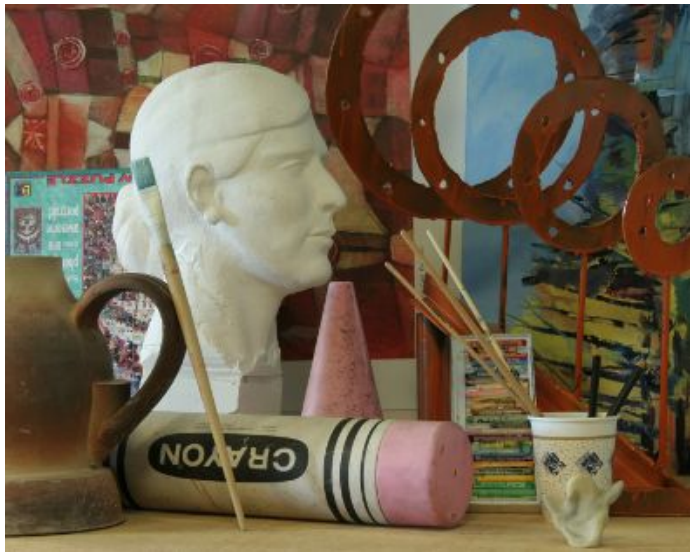


Right Image

Example 2: Man Image

Left Image

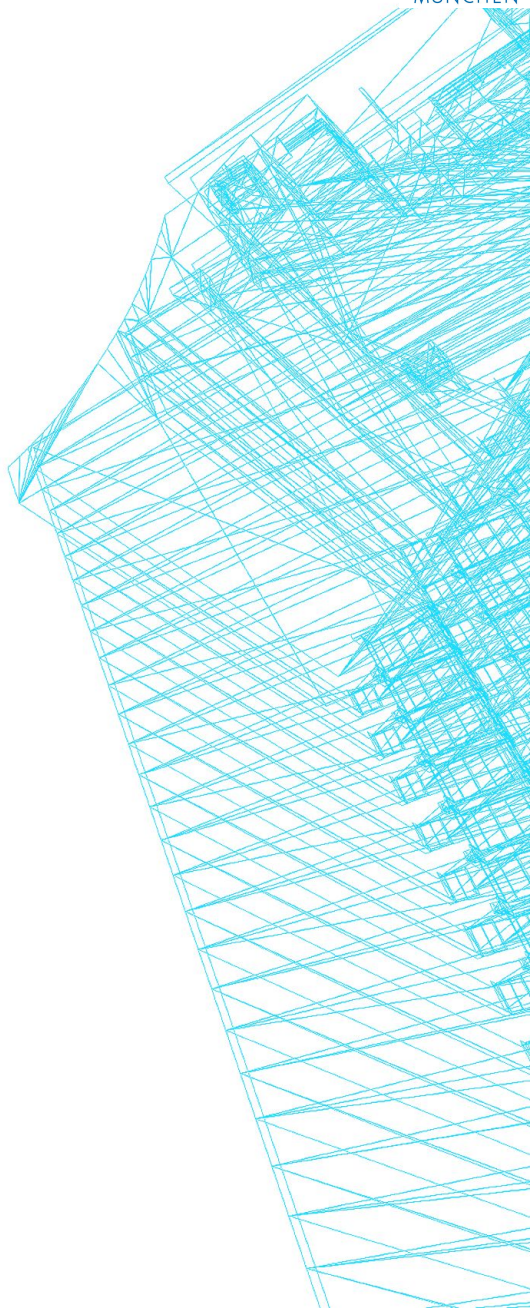
Depth image



Right Image

THEORY

"If the facts don't fit the theory, change the facts." — Albert Einstein



Formulation: Variational Method

- Energy Equation in terms of disparity

$$min\left\{\int_{\Omega} \underbrace{|\nabla u(x)|\, dx}_{\text{Regularizer}} + \int_{\Omega} \underbrace{\rho(u(x), I_l, I_r)}_{\text{Data Term}}\right\}$$

  - Regularizer – Smoothness in the disparity values of the neighboring pixels.
  - Data Term – Stereo image matching for the given disparity level.

Primal Dual Formulation

- Introduce Dual variable φ(3D vector), acts as a diffusion operator of the depth map v.

$$\min_{v}\{\max_{\varphi}\{\int_{\Sigma} \nabla_3 v.\,\varphi\}\}$$

- Primal Dual Update steps

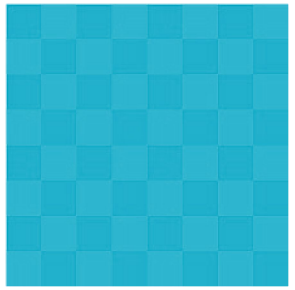
$$\left(\varphi^h\right)^{n+1} = proj_k\left((\varphi^h)^n + \sigma(\nabla v^n)\right)$$

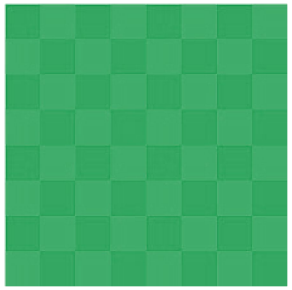
$$(v^h)^{n+1} = proj_c\left(\left(v^h\right)^n + \tau(div\left(\varphi^h\right)^{n+1})\right)$$

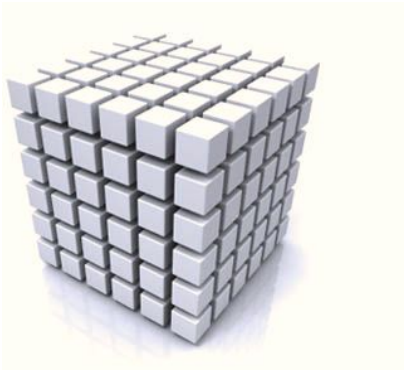
$$(\bar{v}^h)^n = 2(v^h)^{n+1} - (v^h)^n$$

- Projection operator implies mapping of the dual and primal variables on the respective subspaces.

FORMULATION: VISUALISATION





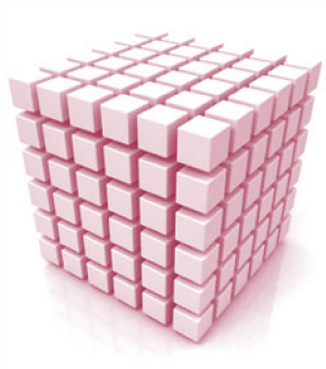


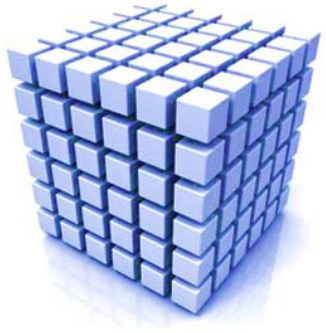
	Left Image, I_{left}	Right Image, I_{right}	Data Term, g
Dimensions	w x h	w x h	w x h x s
Value of a cell	[0 , 1]	[0 , 1]	R
Calculation	<input>	<input>	$g(i, j, k) = \mu(Iright(i, j) - Ileft(i + k, j))$

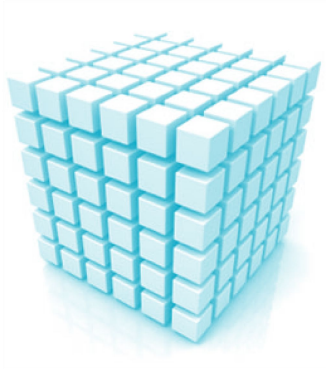
where s = slice depth

initialization

FORMULATION: VISUALISATION



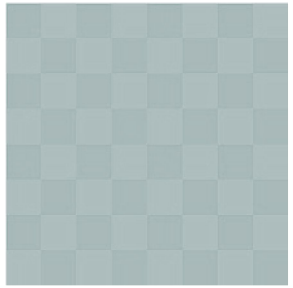




	Dual, Φ	Primal, v	Extrapolation, $\tilde{v}$
Dimensions	w x h x s	w x h x s	w x h x s
Value of a cell	R^3	[0, 1]	[-1, 2]
Calculation	$\Phi_{n+1} = \Phi_n + \sigma\nabla v_n$	$v_{n+1} = v_n - \tau\Delta\Phi_{n+1}$	$v_{n+1} = 2v_{n+1} - v_n$

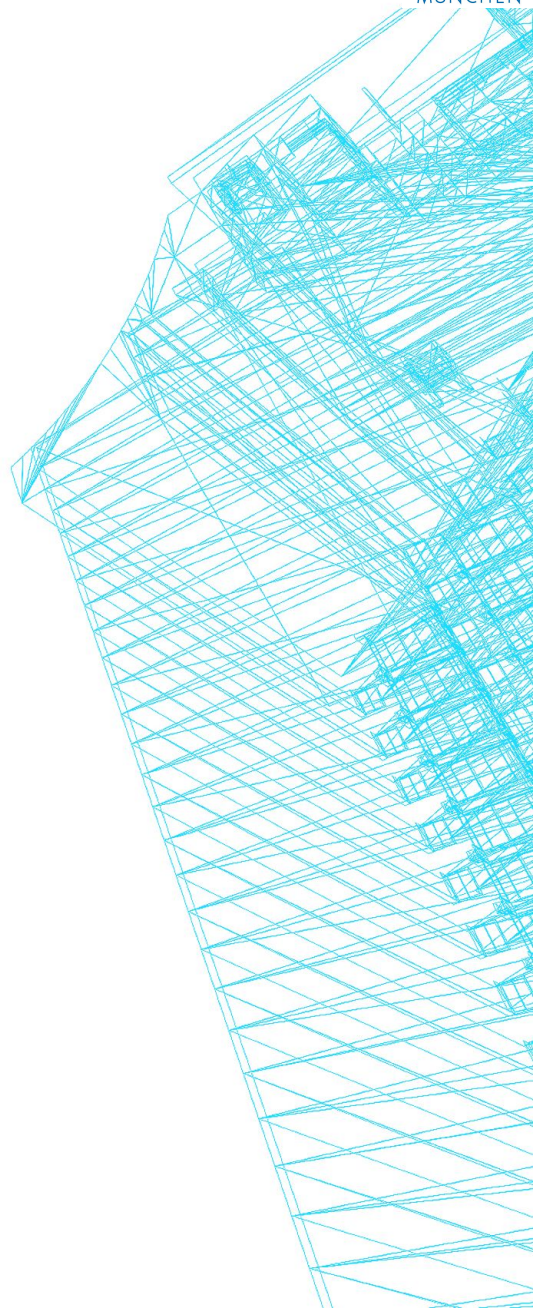
until convergence

FORMULATION: VISUALISATION



	Depth Map, d
Dimensions	w x h
Value of a cell	[0, 255]
Calculation	<table><tr><th></th><th>Depth Map, d</th></tr><tr><td>Dimensions</td><td>w x h</td></tr><tr><td>Value of a cell</td><td>[0, 255]</td></tr><tr><td>Calculation</td><td>$g(i,j) = \sum_{k=0}^{s} v_{n+1}(i,j,k)$</td></tr></table>

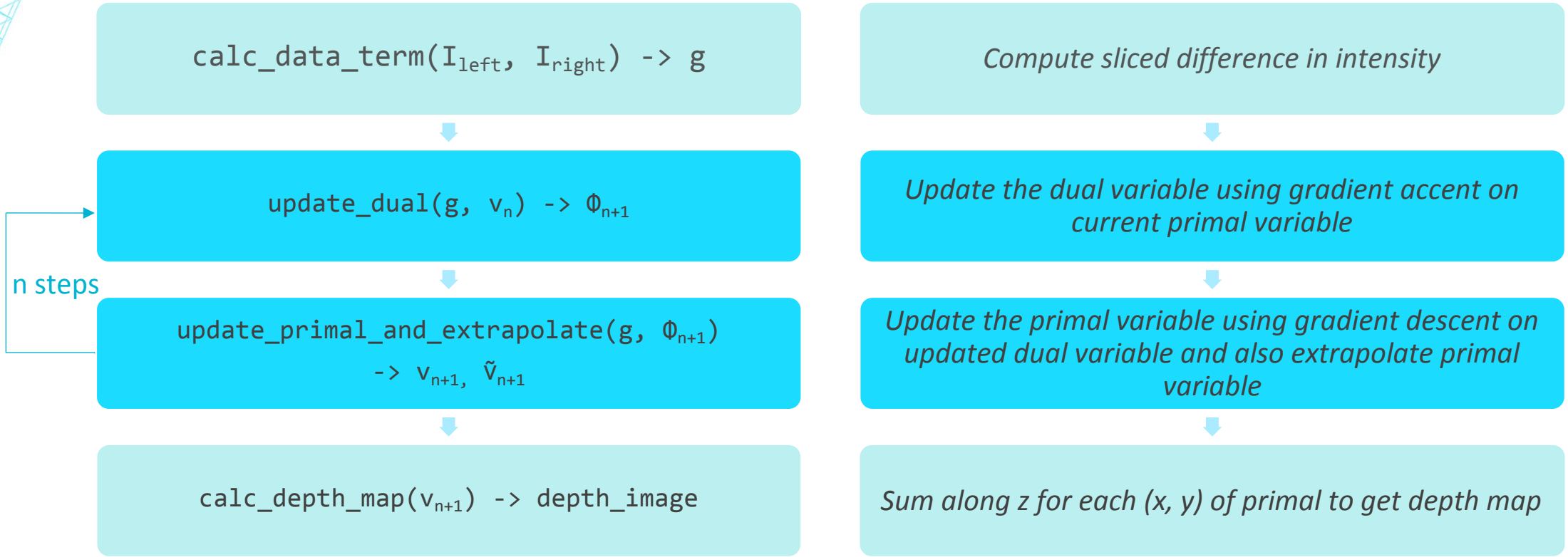
TECHNISCHE UNIVERSITÄT MÜNCHEN



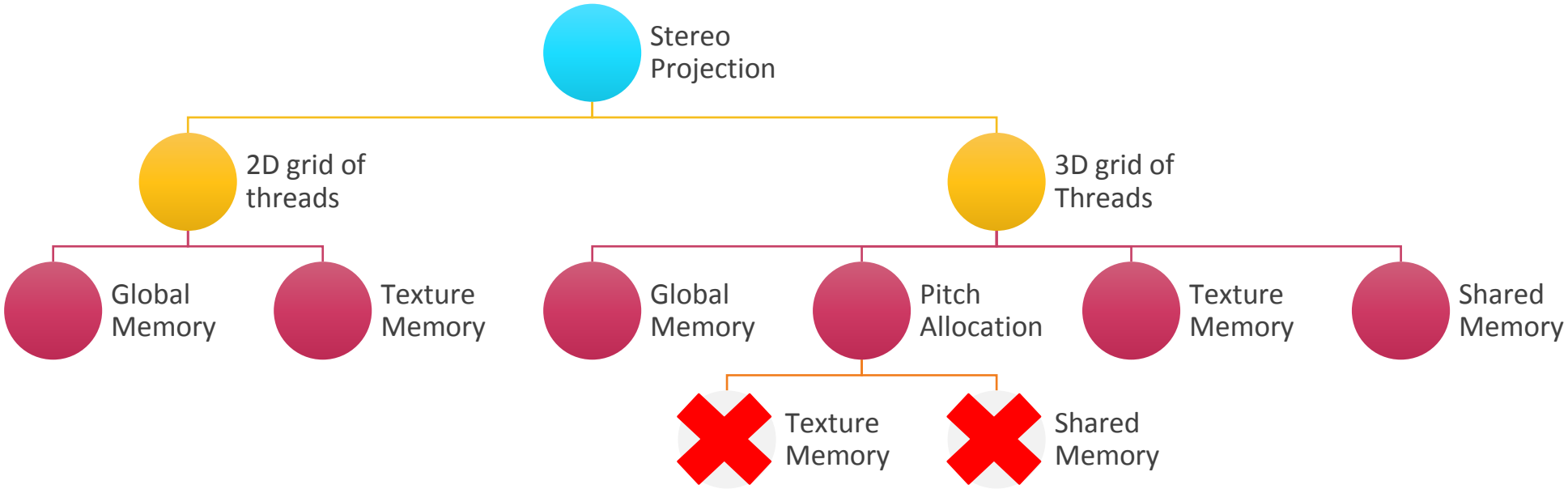
Implementation

"A good idea is about ten percent and implementation and hard work, and luck is 90 percent." — Guy Kawasaki

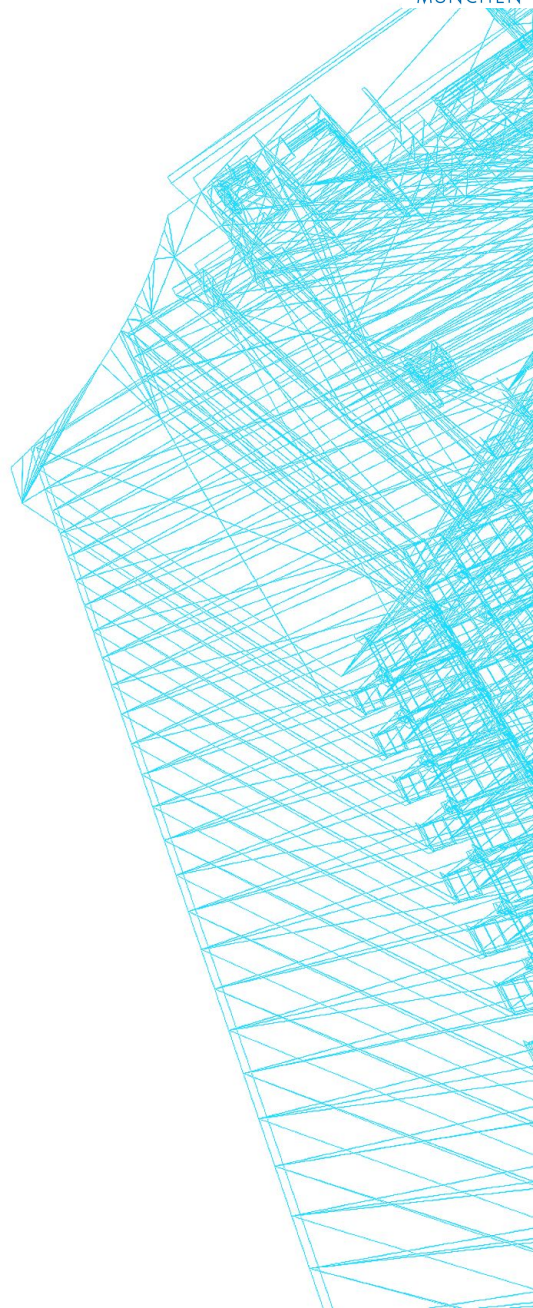
IMPLEMENTATION: Kernels



Implementation: GPU Techniques



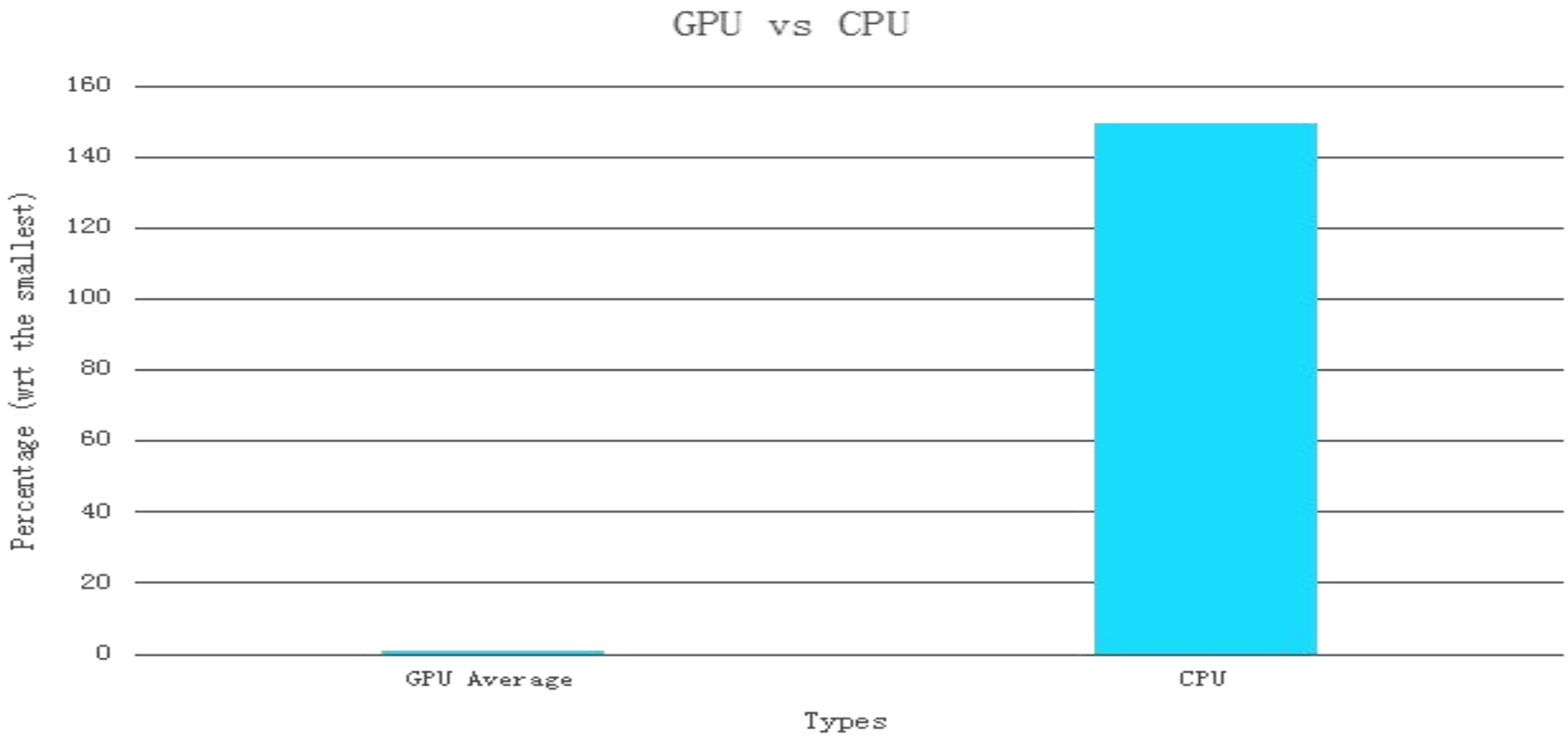
TECHNISCHE UNIVERSITÄT MÜNCHEN



RESULTS and performance

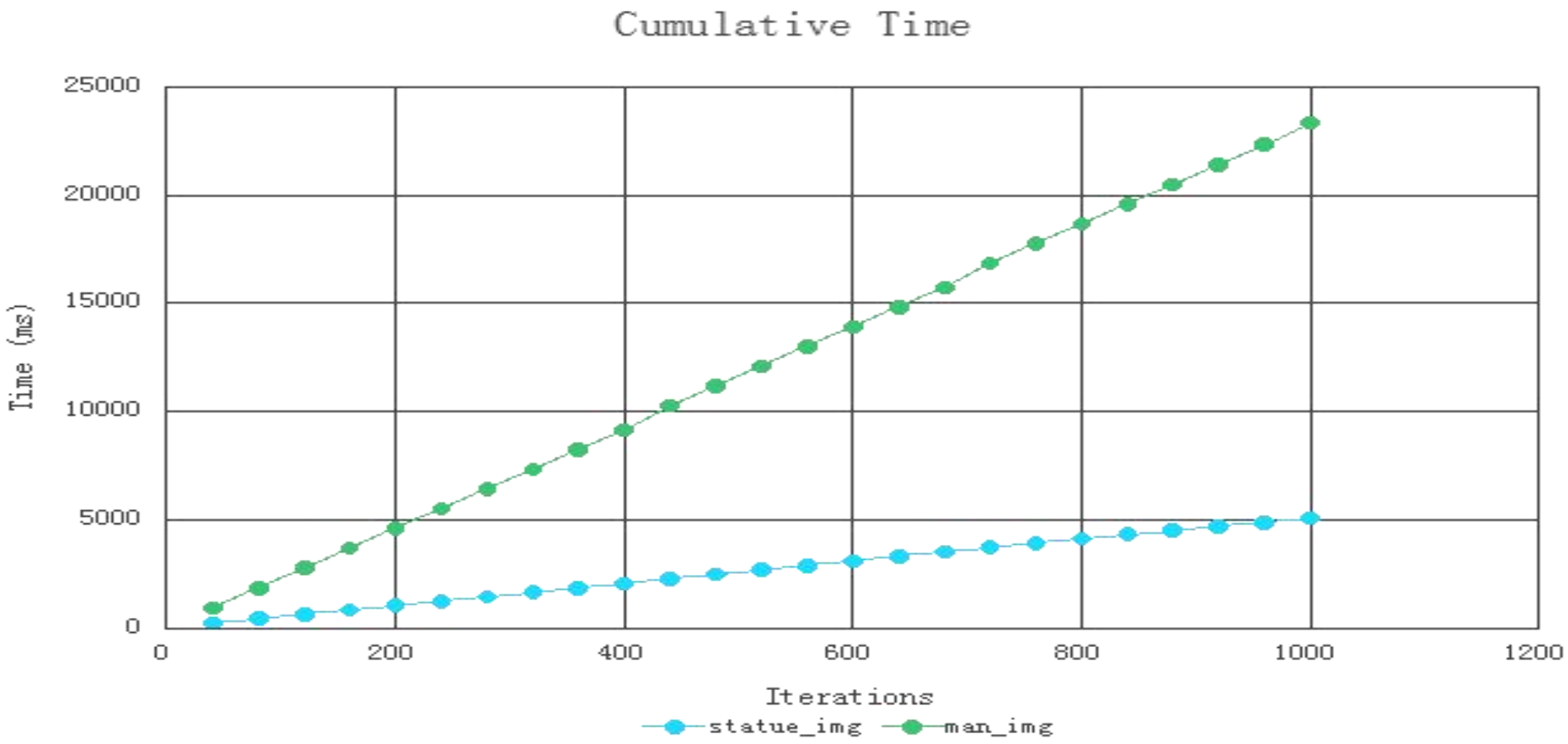
"Insanity: doing the same thing over and over again and expecting different results." — Albert Einstein

RESULTS: GPU v/s CPU

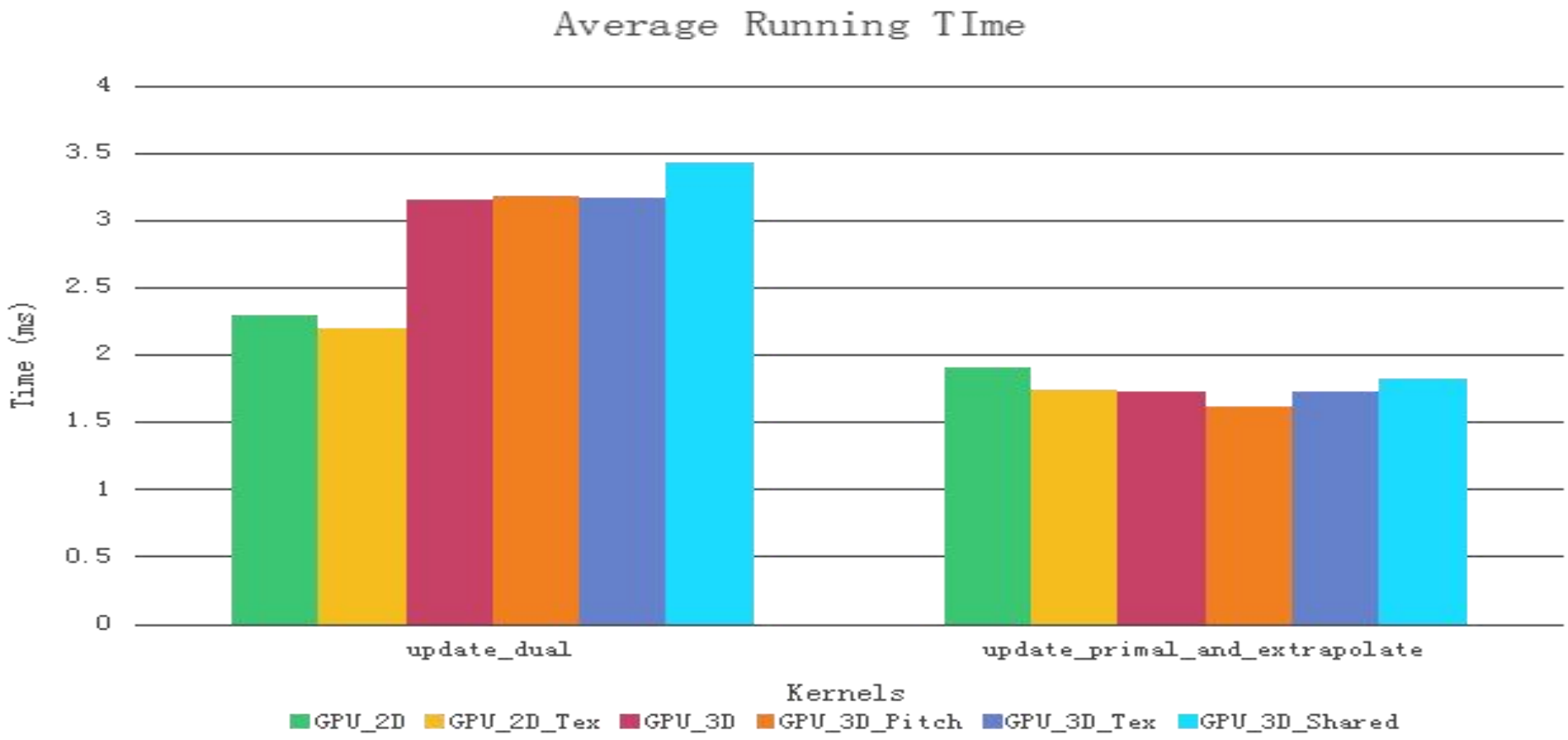


RESULTS: RUN TIME RESULTS

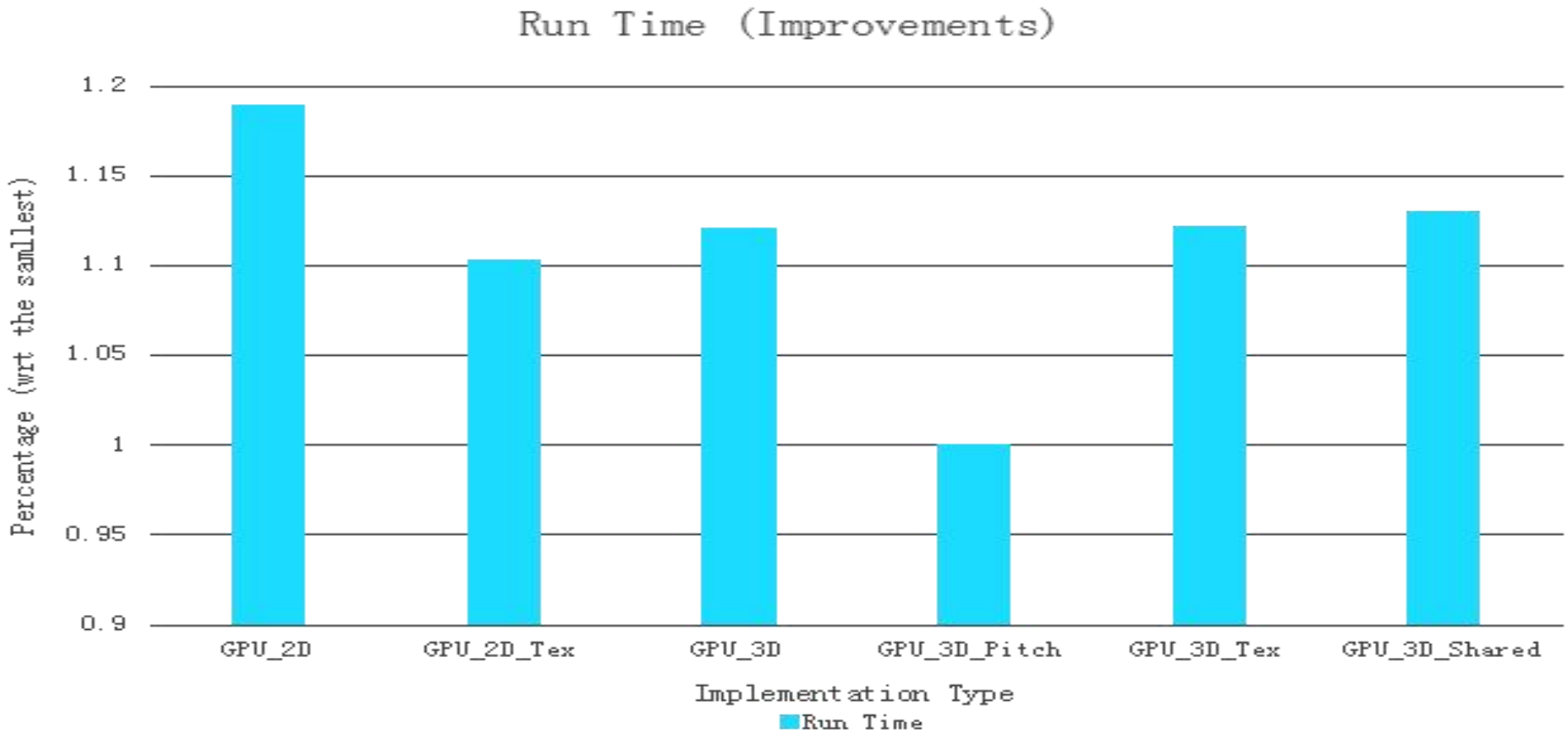
Results: CONVERGENCE



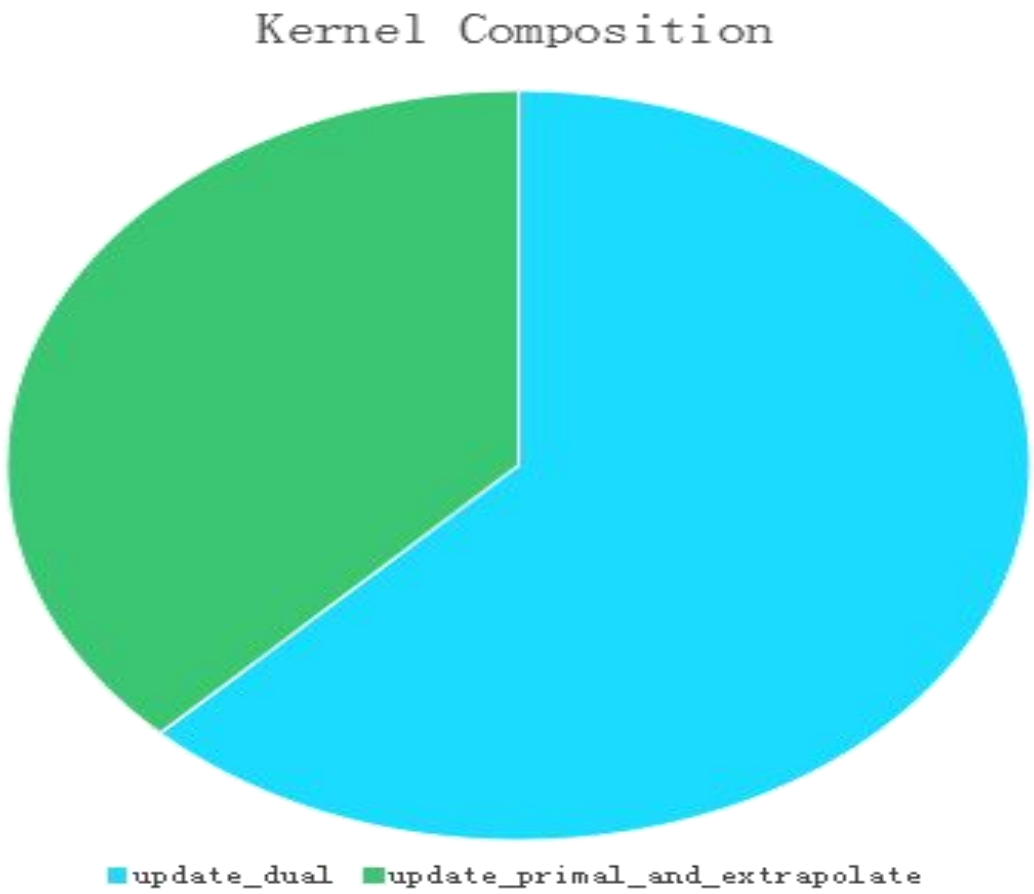
RESULTS: KERNEL AVERAGE RUNTIME



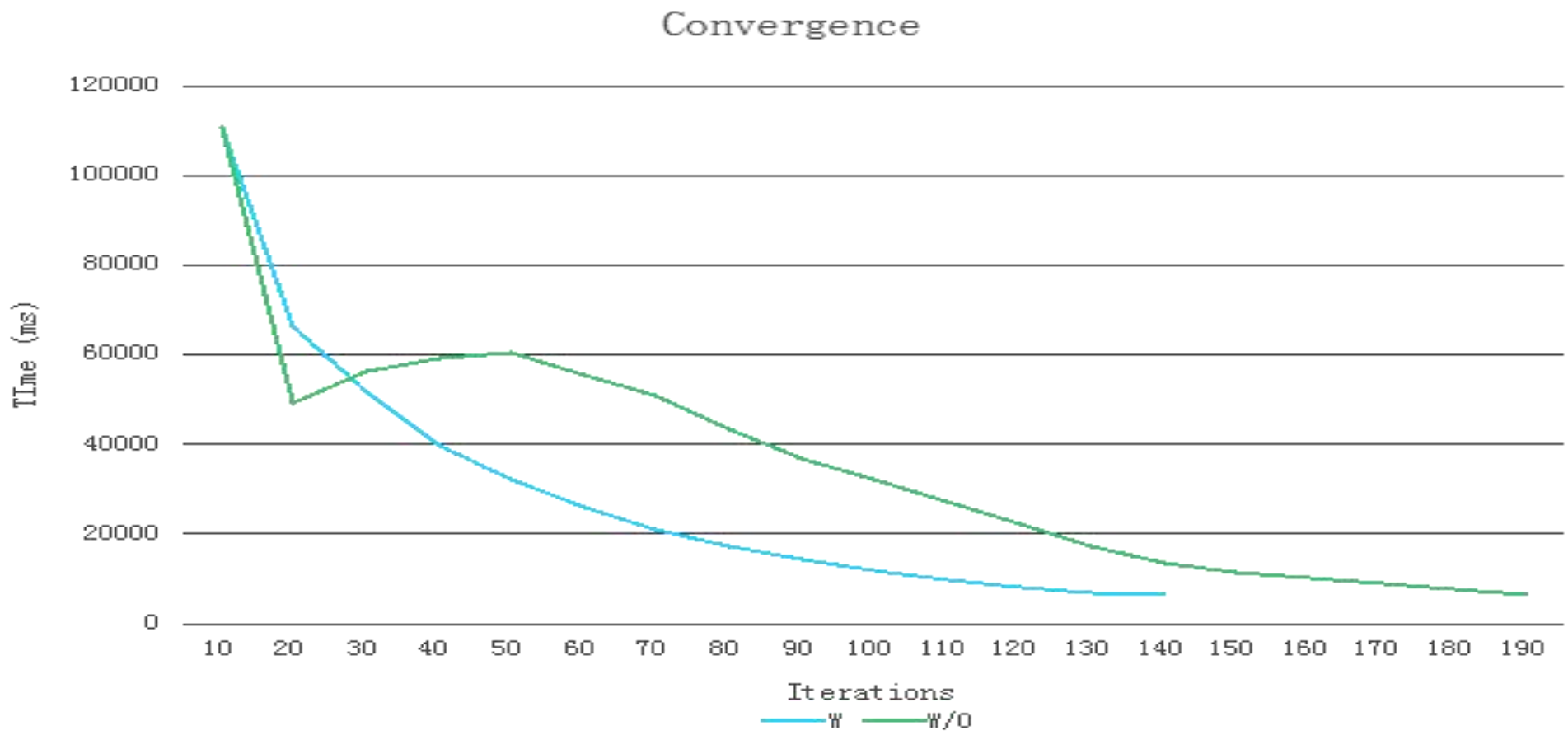
RESULTS: TYPES COMPARISONS



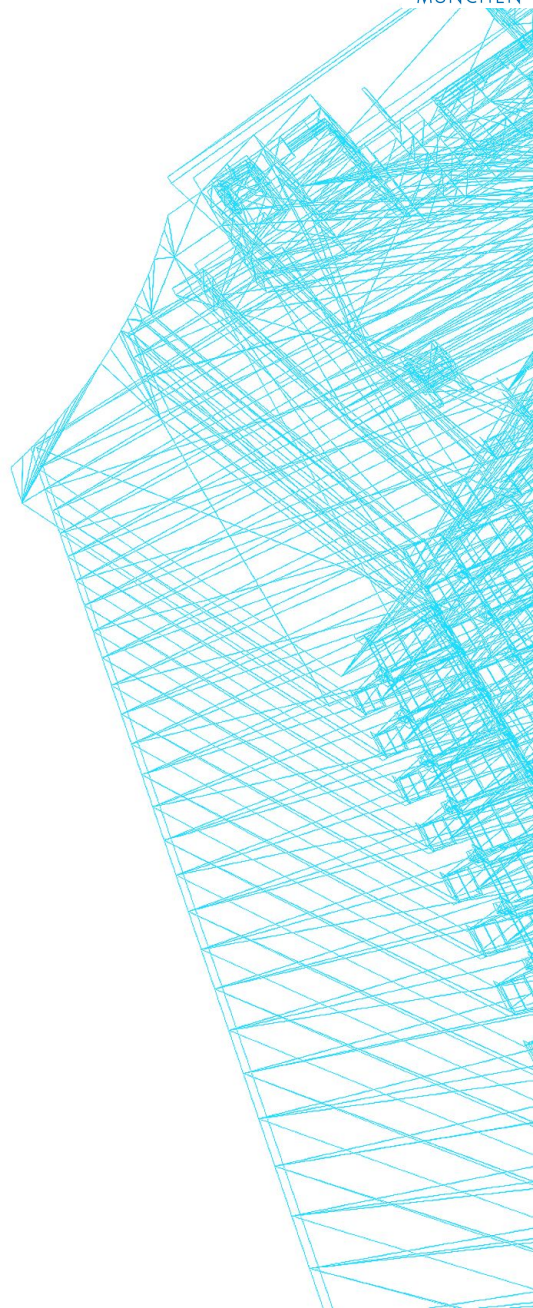
Results: COMPOSITION OF KERNELS



Results: CONVERGENCE (HACK)



TECHNISCHE UNIVERSITÄT MÜNCHEN



GUI and DEMO

"What is it indeed that gives us the feeling of elegance in a solution, in a demonstration?" — Henri Poincare

To Sum Up

- Great performance boost compared to CPU.
- We have seen different types of GPU techniques.
- In cc >= 2.x, texture does not give a big boost.
- Pitch allocation always improves performance (except for already pitched memory).
- Shared Memory is useful in case of $O(n^2)$ access.
- In convex problems, initialization hacks may lead to faster convergence.

References

[1] Pock T., Cremers D., Bischof H., Chambolle A., *Global Solutions of Variational Models with Convex Regularization*

[2] Variational Methods, TUM Class Lectures

[3] CUDA Programming GUIDE