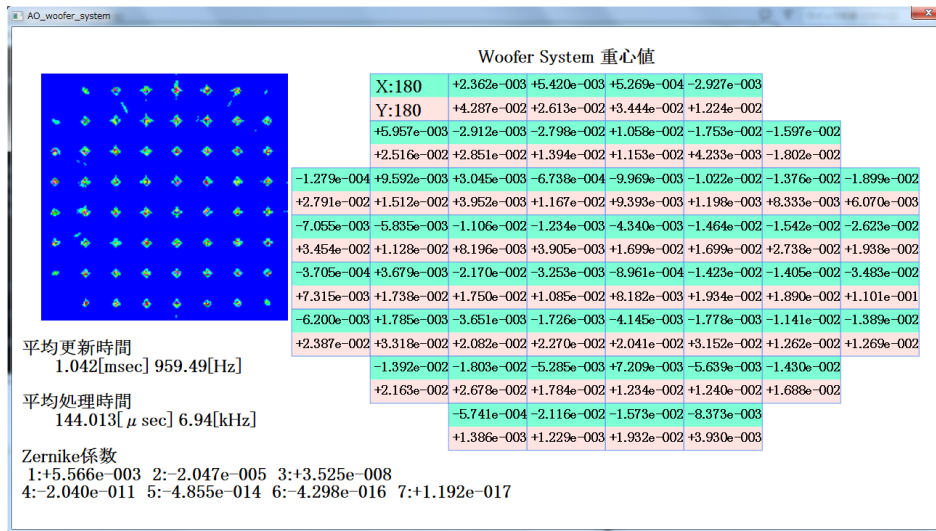


極限補償光学 AO制御実験



大阪電気通信大学 工学部
電子機械工学科
中村 祐一

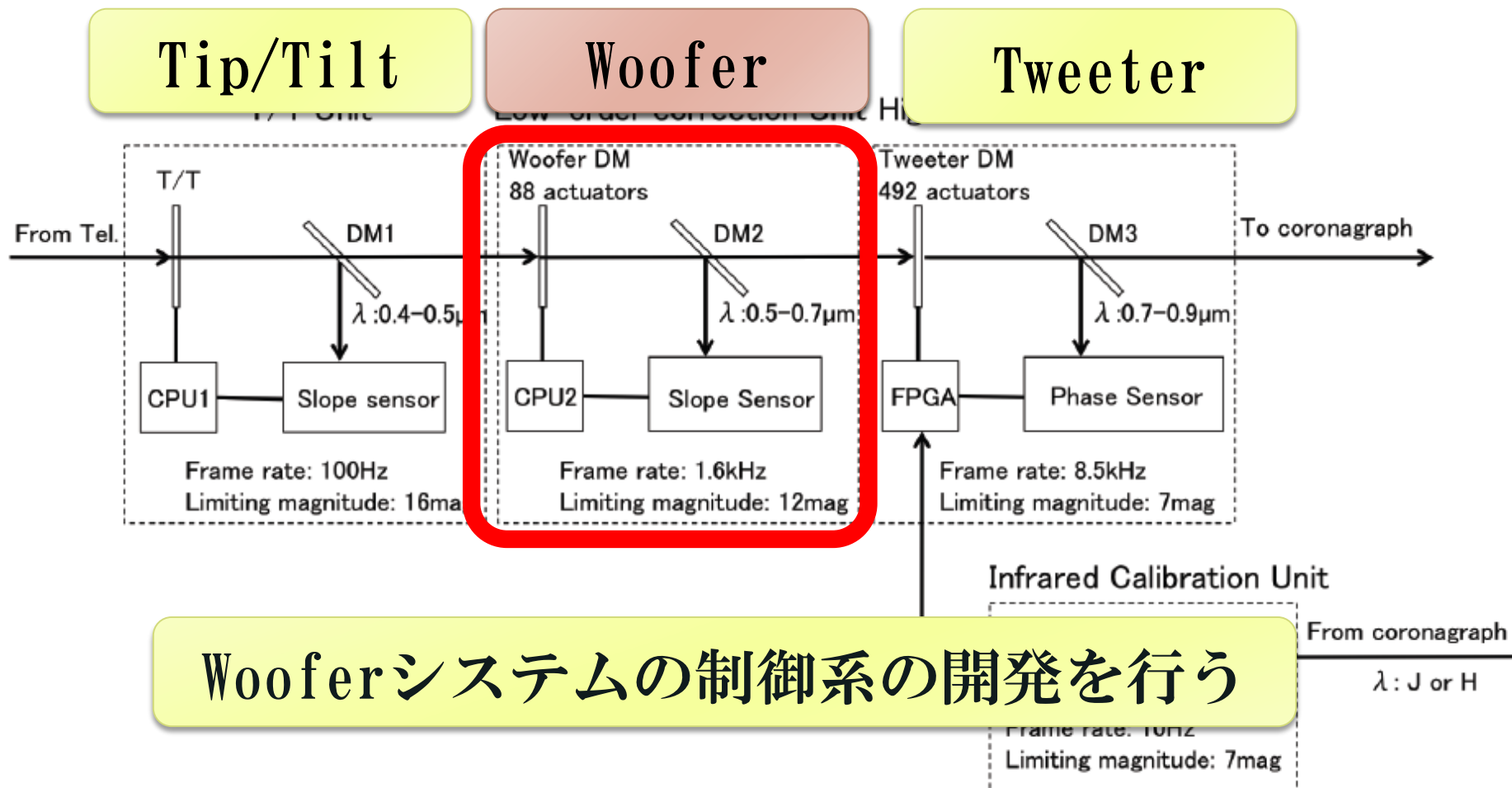
目次

- ▶ 1. 緒言
- ▶ 2. システム概要
 - ▶ 2.1 現在の状況
 - ▶ 2.2 CCDカメラとコンピュータの接続
- ▶ 3. ソフトウェア開発
 - ▶ 3.1 MATLABとC言語のベンチマーク
 - ▶ 3.2 重心計算の分散処理
 - ▶ 3.3 DirectXライブラリを利用して描画処理を高速化
- ▶ 4. 結言

目次

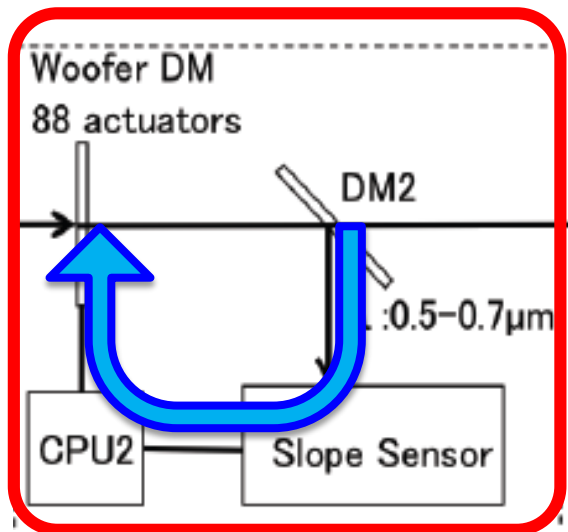
- ▶ **1. 緒言**
- ▶ **2. システム概要**
 - ▶ 2.1 現在の状況
 - ▶ 2.2 CCDカメラとコンピュータの接続
- ▶ **3. ソフトウェア開発**
 - ▶ 3.1 MATLABとC言語のベンチマーク
 - ▶ 3.2 重心計算の分散処理
 - ▶ 3.3 DirectXライブラリを利用して描画処理を高速化
- ▶ **4. 結言**

1. 緒言



Wooferシステムの制御系の開発を行う

1. 緒言



▶ Wooferシステムの制御ソフト設計

wooferシステム制御周期[Hz]

理想	2000
昨年度	5

◆ 原因

▶ SHWFSの重心計算

- ▶ 制御ソフトに**MATLAB（数値計算言語）**を利用している。
 - ▶ カメラの数値変換処理に時間がかかる

MATLAB

アプリケーションを介さず
C言語で直接設計し制御する

必要がある

目次

- ▶ 1. 緒言
- ▶ 2. システム概要
 - ▶ 2.1 現在の状況
 - ▶ 2.2 CCDカメラとコンピュータの接続
- ▶ 3. ソフトウェア開発
 - ▶ 3.1 MATLABとC言語のベンチマーク
 - ▶ 3.2 重心計算の分散処理
 - ▶ 3.3 DirectXライブラリを利用して描画処理を高速化
- ▶ 4. 結言

2. システム概要

88 actuators DM



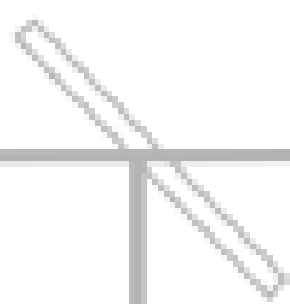
ALPAO DM-88 25



制御コンピュータ



浜松ホトニクス CCDカメラ
C11440-22CU



$\lambda : 0.5 - 0.7 \mu\text{m}$

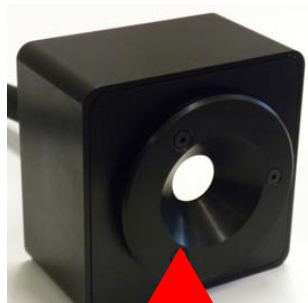
or

2. システム概要

2. 1 現在の状況

10月9日レーザ変位計を利用したDMの動作確認

- ・ 8月2週目 重心計算プログラムの作成
- ・ 8月3週目 CCDカメラとコンピュータの通信を確立
CCDを利用したリアルタイム重心計算
表示用GUIの作成
- ・ 8月4週目 SHWFSの重心計算高速化



ALPAO DM-88 25



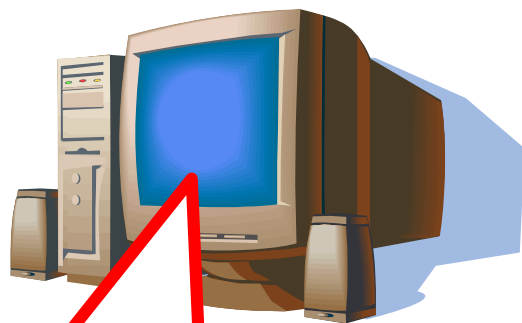
制御コンピュータ



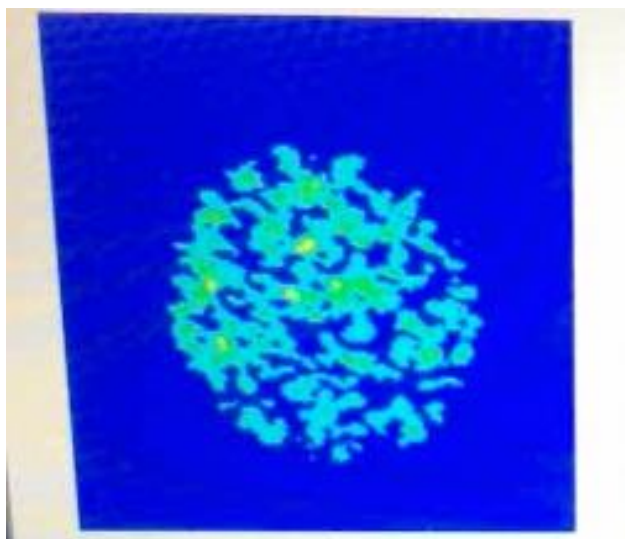
浜松ホトニクス CCDカメラ
C11440-22CU

2. システム概要

2. 2 CCDカメラとコンピュータの通信を確立



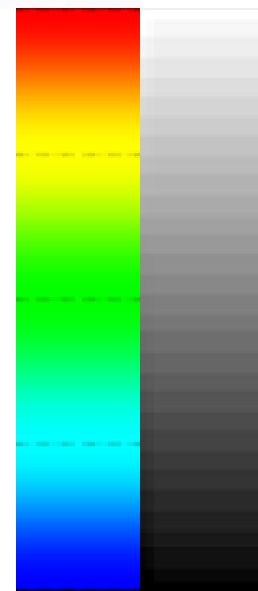
USB 3.0



16bit グレースケールデータ
∵各8bitのRGBカラー表示のみ対応

↓ (変換)

BGRカラーデータ



変換

2. システム概要

2. 2 CCDカメラとコンピュータの通信を確立

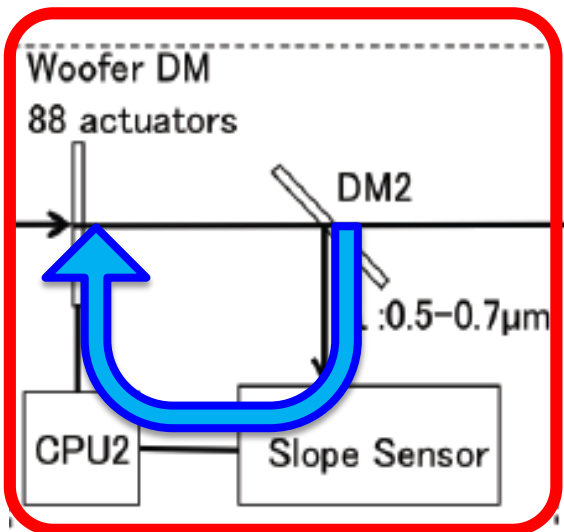


理想的な条件は達成できない

最高速度	500
最低速度	280
平均速度	382

最高速度	1005
最低速度	889
平均速度	951

1. 緒言



▶ Wooferシステムの制御ソフト設計

wooferシステム制御周期[Hz]

暫定目標	500
昨年度	5

◆ 原因

▶ SHWFSの重心計算

- ▶ 制御ソフトに**MATLAB（数値計算言語）**を利用している。
 - ▶ カメラの数値変換処理に時間がかかる

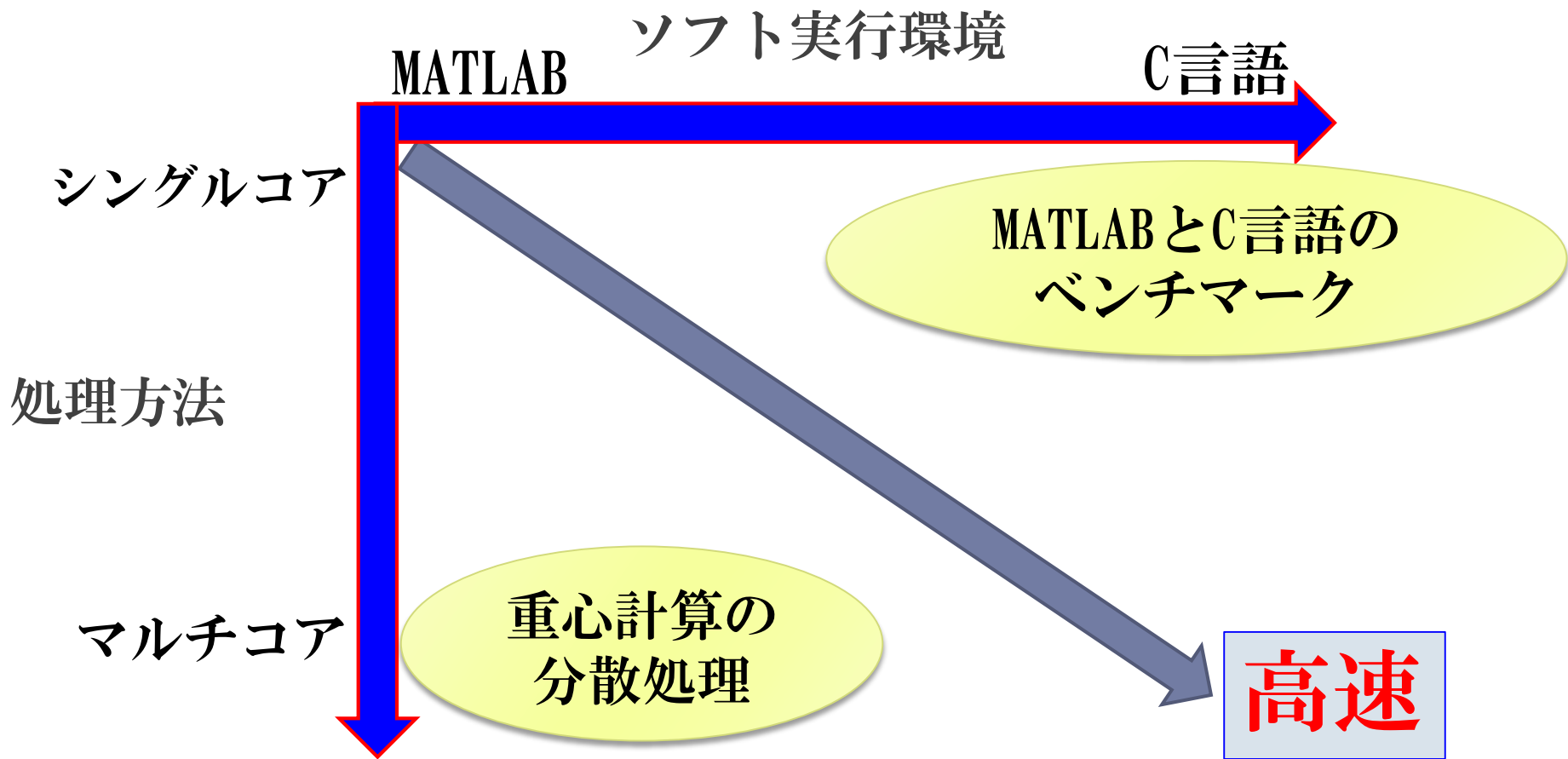
アプリケーションを介さず
C言語で直接設計し制御する

目次

- ▶ 1. 緒言
- ▶ 2. システム概要
 - ▶ 2.1 現在の状況
 - ▶ 2.2 CCDカメラとコンピュータの接続
- ▶ 3. ソフトウェア開発
 - ▶ 3.1 MATLABとC言語のベンチマーク
 - ▶ 3.2 重心計算の分散処理
 - ▶ 3.3 DirectXライブラリを利用して描画処理を高速化
- ▶ 4. 結言

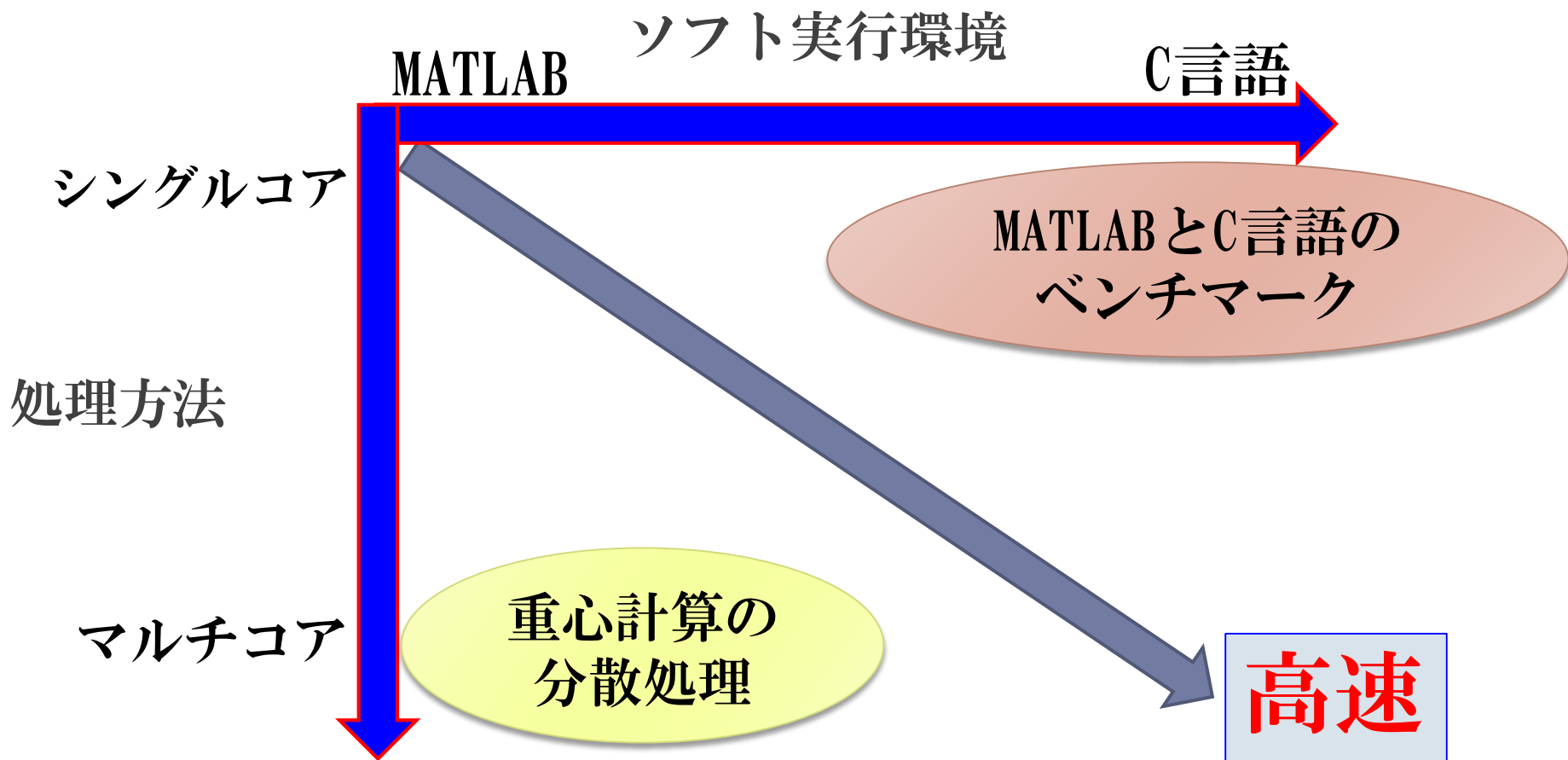
3. ソフトウェア開発

▶ SHWFS重心計算の高速化の手法



3. ソフトウェア開発

▶ SHWFS重心計算の高速化の手法



3. ソフトウェア開発

3. 1 MATLABとC言語のベンチマーク

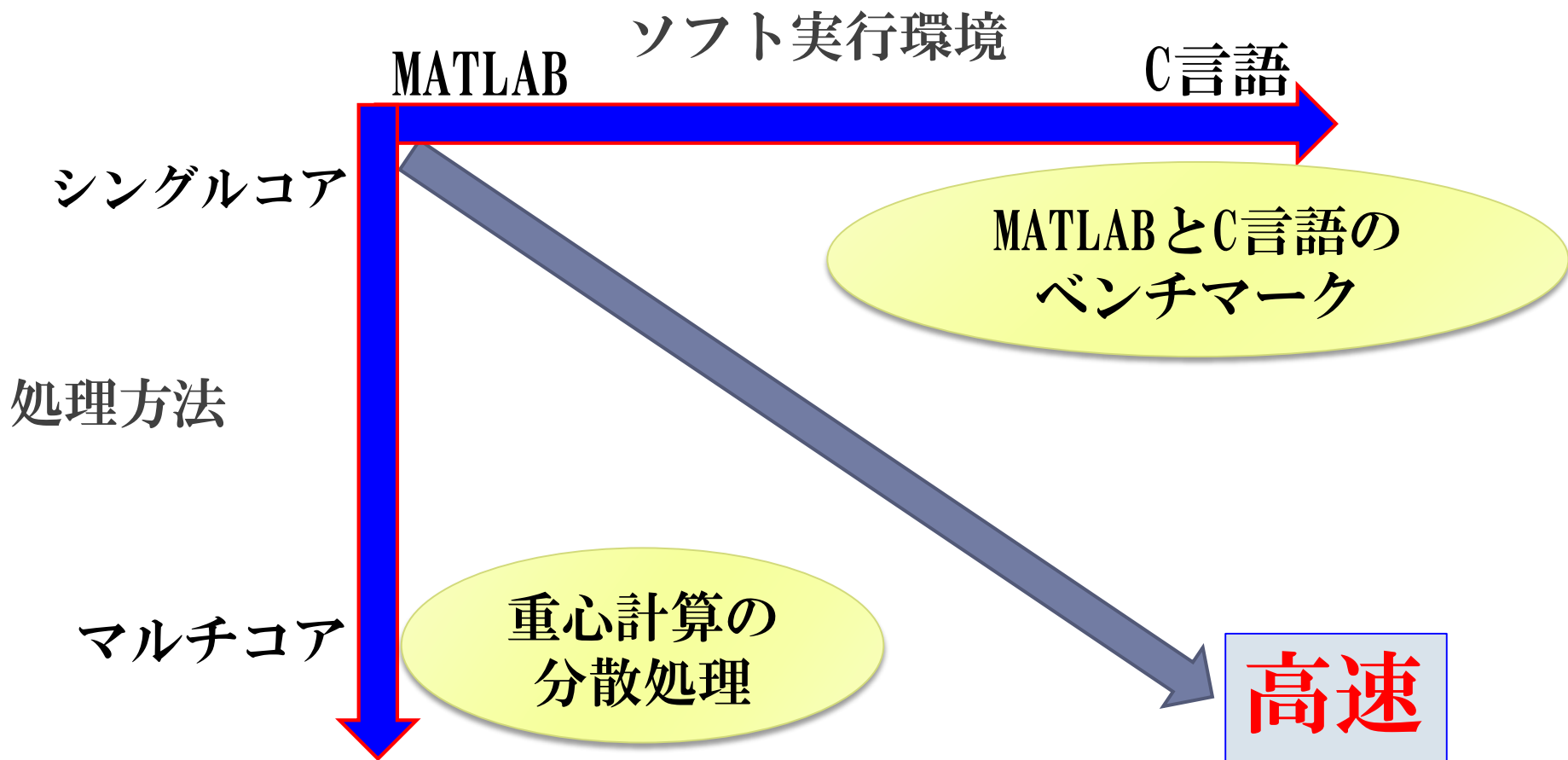
- ▶ MATLAB上で行っていた重心計算をC言語に変更
- ▶ 重心計算はMATLABと同じMoment法を利用

MATLAB	22Hz	約207倍
C言語	4545Hz	

MATLABで計測したときと同じデータを利用

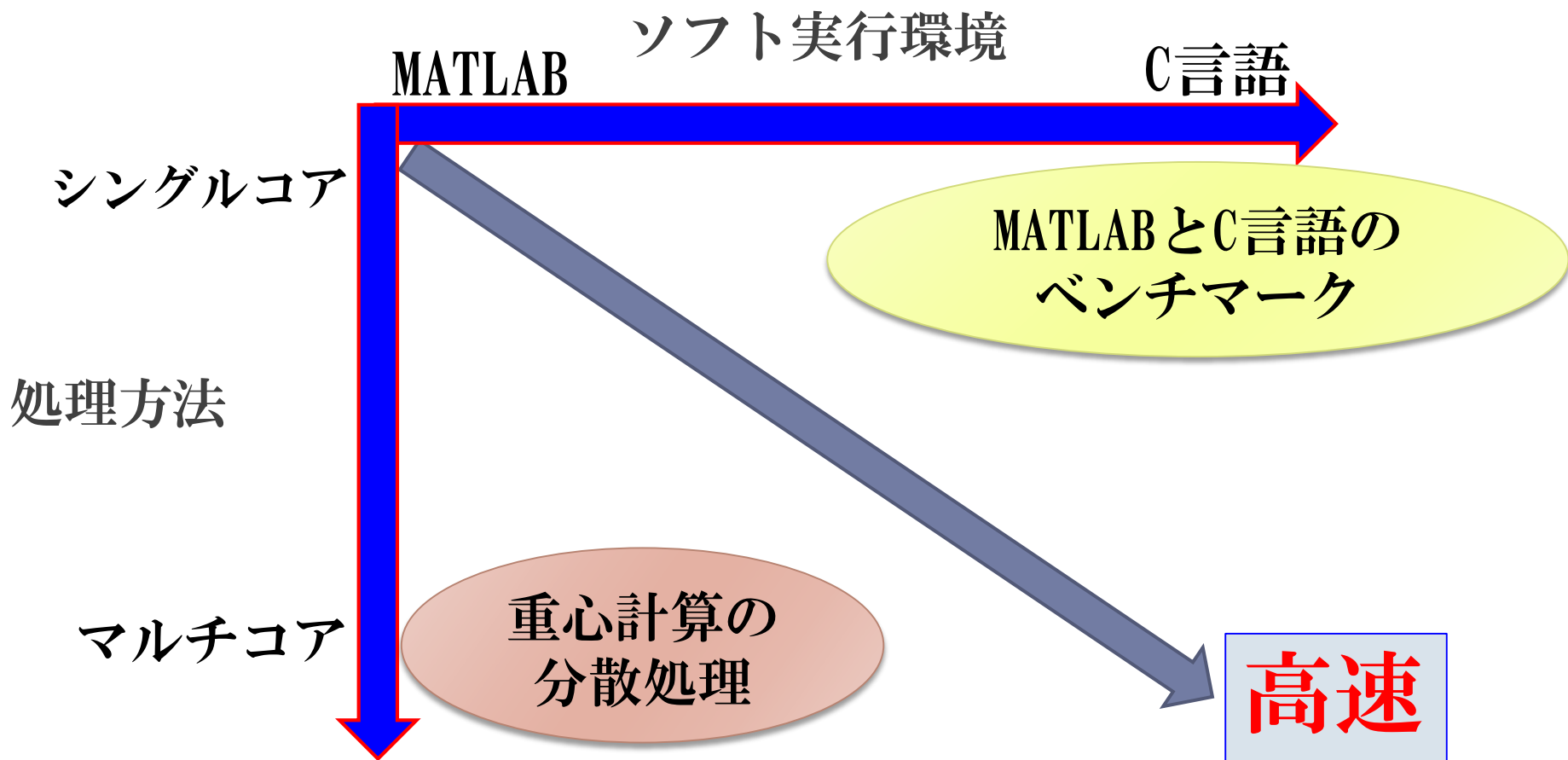
3. ソフトウェア開発

▶ SHWFS重心計算の高速化の手法



3. ソフトウェア開発

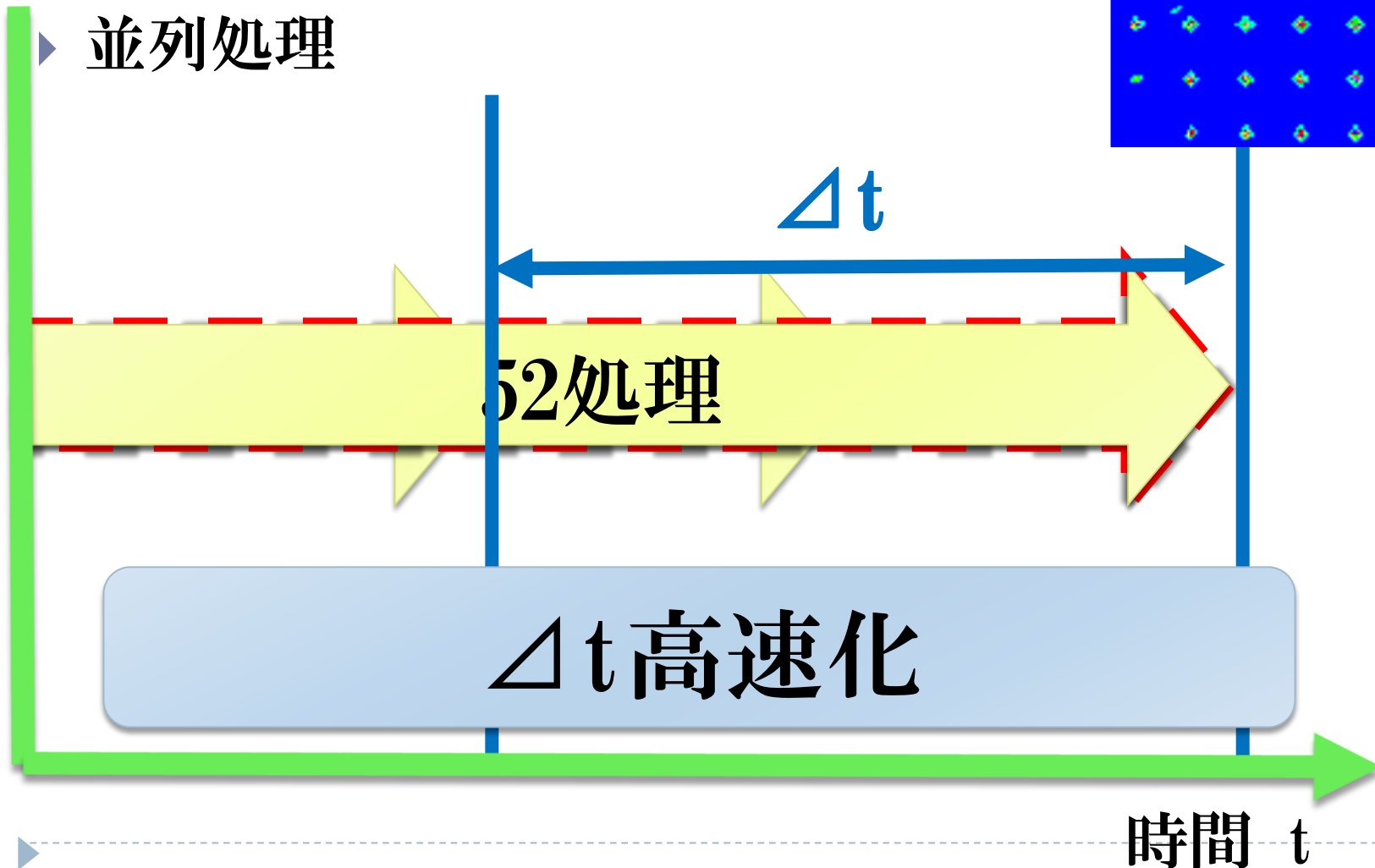
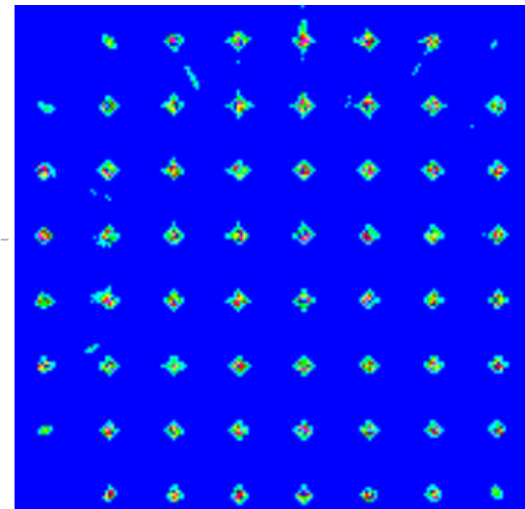
▶ SHWFS重心計算の高速化の手法



3. ソフトウェア開発

3. 2 重心計算の分散処理

- ▶ 52素子順番に行っている処理を**分割**
- ▶ 並列処理



3. ソフトウェア開発

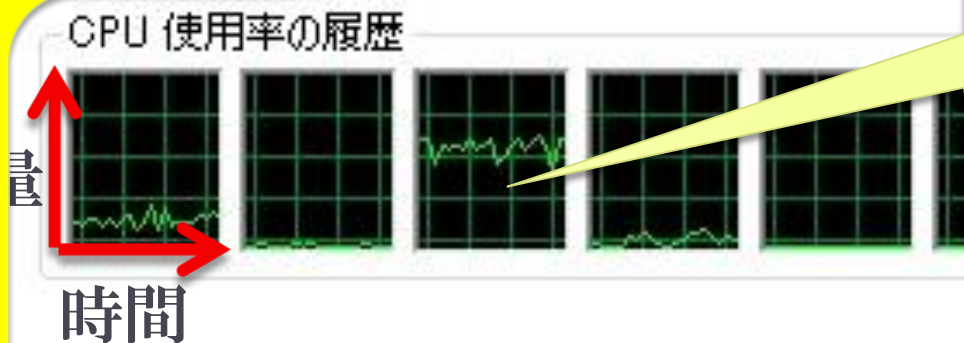
3. 2 重心計算の分散処理

分割なし

演算時間

平均 5.92kHz

重心計算



約1.4倍

演算時間

平均 8.51kHz

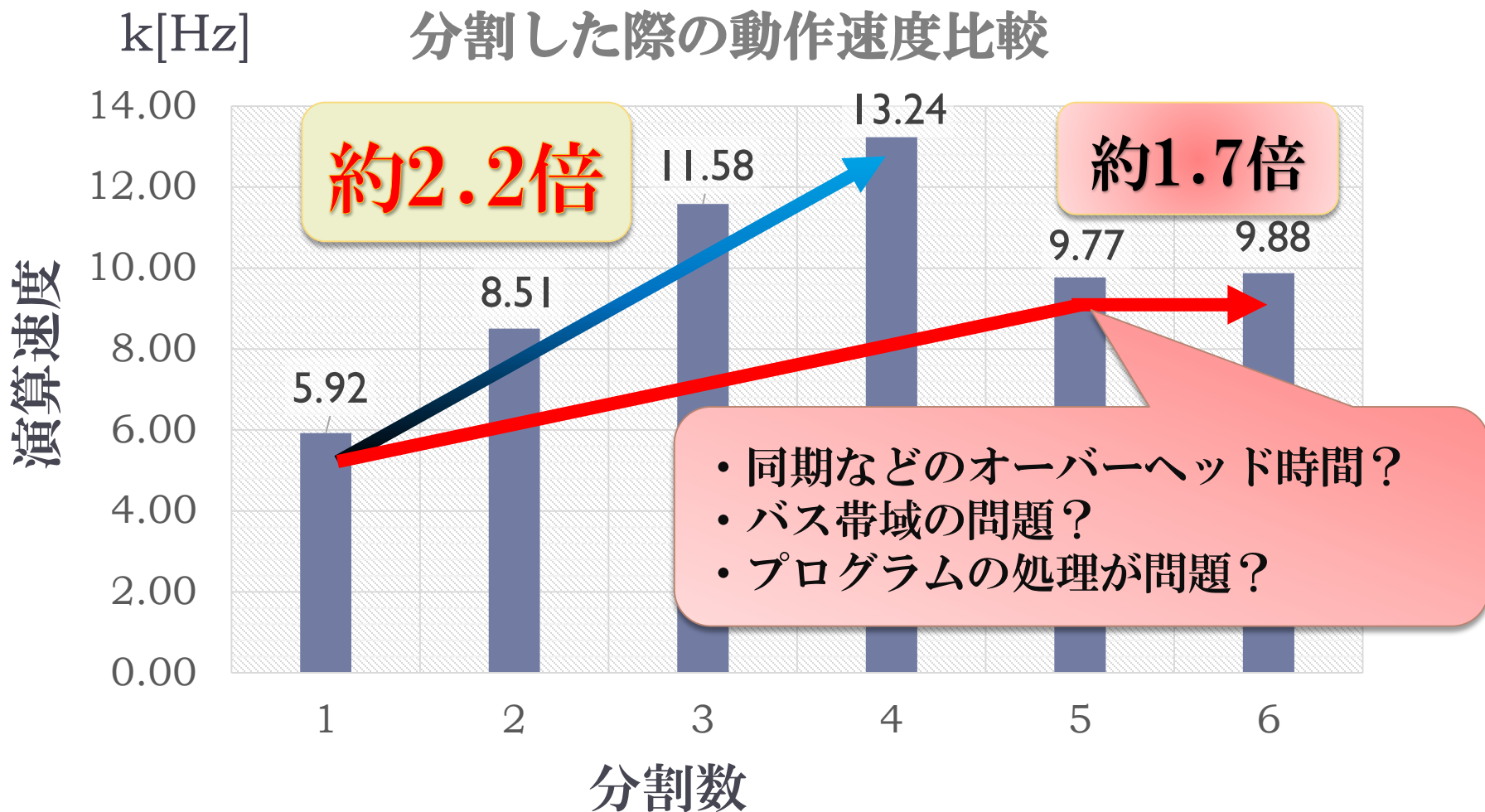
重心計算 1

重心計算 2



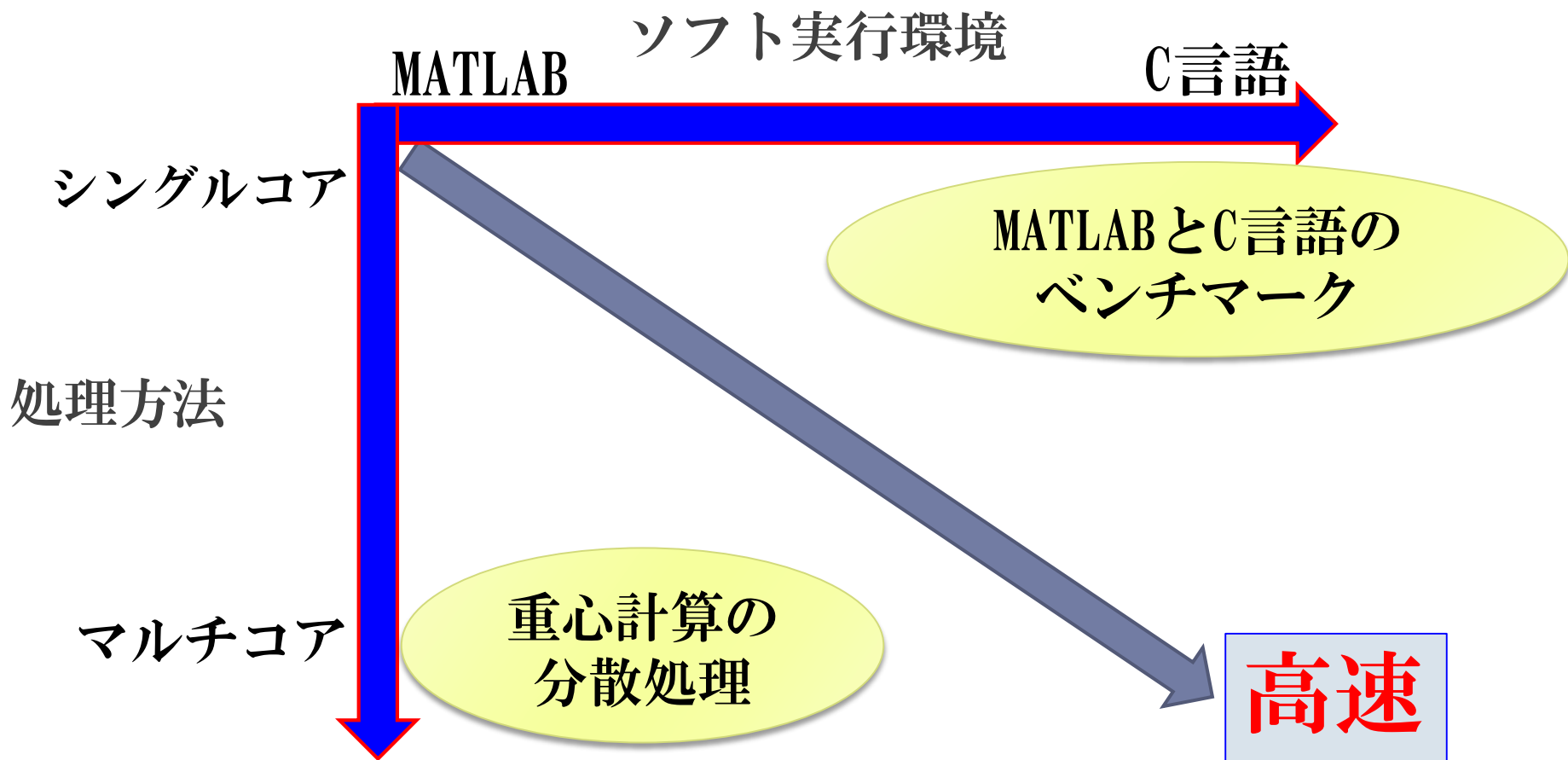
3. ソフトウェア開発

3. 2 重心計算の分散処理



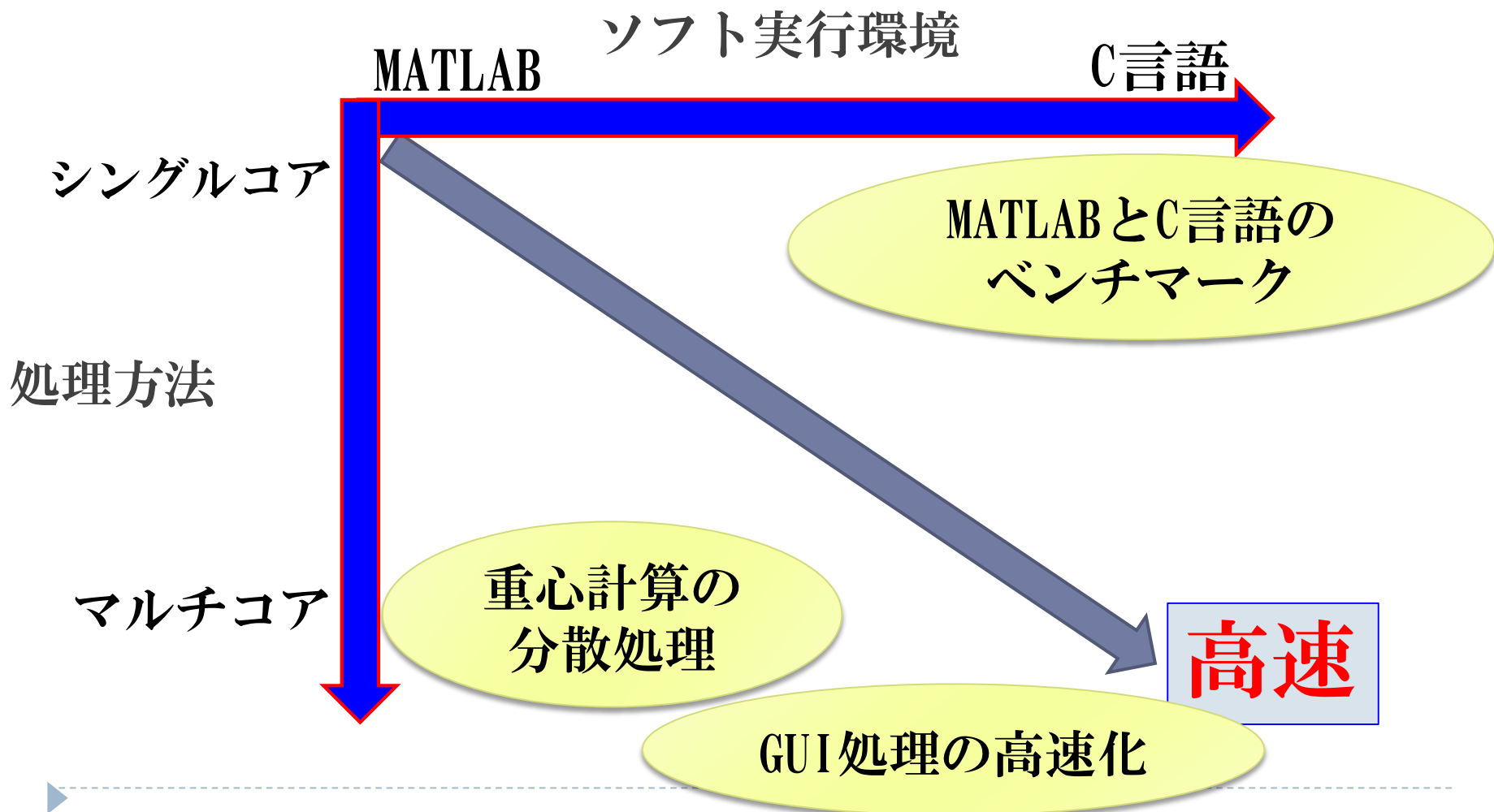
3. ソフトウェア開発

▶ SHWFS重心計算の高速化の手法



3. ソフトウェア開発

▶ SHWFS重心計算の高速化の手法



3. ソフトウェア開発

3. 3 DirectXライブラリを利用して描画処理を高速化

▶ CPUで行う描画(表示用GUI)の大部分をGPUで処理

CPUの負担が低減して速度上昇

マウスで初期座標を選択することで重心計算開始
初期座標の保存も可能

算出された重心位置や各素子のピーク位置の表示

+2.516e-002	+2.851e-002	+1.394e-002	+1.153e-002	-1.802e-002			
-1.279e-004	+9.592e-003	+3.045e-003	-6.738e-004	-9.969e-003	-1.376e-002	-1.899e-002	
+2.791e-002	+1.512e-002	+3.952e-003	+1.167e-002	+9.393e-003	-1.198e-003	+8.333e-003	+6.070e-003
-7.055e-003	-5.835e-003	-1.106e-002	-1.234e-003	-4.340e-003	-1.464e-002	-1.542e-002	-2.623e-002
+3.454e-002	+1.128e-002	+8.196e-003	+3.905e-003	+1.699e-002	+1.699e-002	+2.738e-002	+1.938e-002
-3.3e-003	-8.961e-004	-1.423e-002	-1.405e-002	-3.483e-002			
-3e-002	+8.182e-003	+1.934e-002	+1.890e-002	+1.101e-001			
-3e-003	-4.145e-003	-1.778e-003	-1.141e-002	-1.389e-002			
-3e-002	+2.041e-002	+3.152e-002	+1.262e-002	+1.269e-002			
-3e-003	+7.209e-003	-5.639e-003	-1.430e-002				
+2.103e-002	+2.870e-002	+1.784e-002	+1.234e-002	+1.240e-002	+1.688e-002		
-5.741e-004	-2.116e-002	-1.573e-002	-8.373e-003				
+1.386e-003	+1.229e-003	+1.932e-002	+3.930e-003				

平均更新時間
1.042[msec]

平均処理時間
144.013[μ sec] 6.94[kHz]

Zernike係数
1:+5.566e-003
4:-2.040e-011

カメラの映像
16ビットグレースケール
をBGRに変更

速度等の付加情報を
デバッグ用に表示

目次

- ▶ 1. 緒言
- ▶ 2. システム概要
 - ▶ 2.1 現在の状況
 - ▶ 2.2 CCDカメラとコンピュータの接続
- ▶ 3. ソフトウェア開発
 - ▶ 3.1 MATLABとC言語のベンチマーク
 - ▶ 3.2 重心計算の分散処理
 - ▶ 3.3 DirectXライブラリを利用して描画処理を高速化
- ▶ 4. 結言

結言

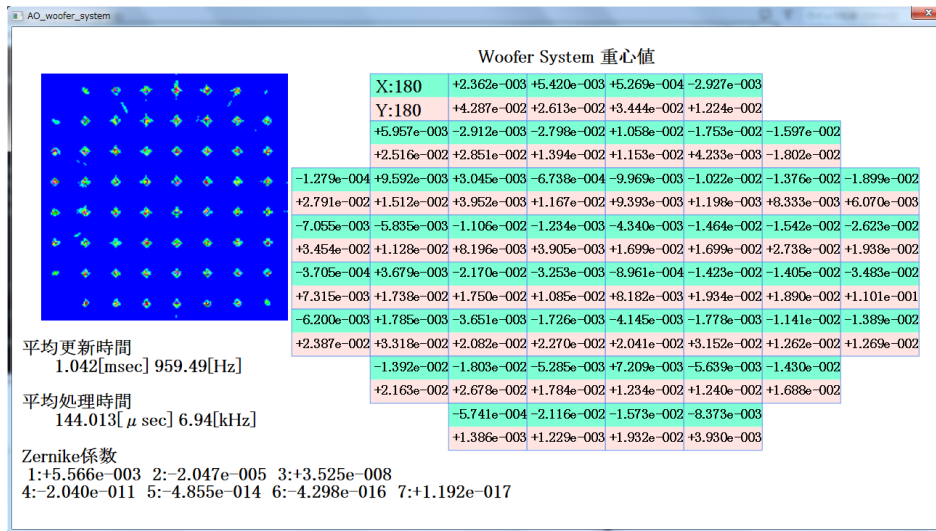
作業内容まとめ

- ▶ ①CCDカメラとコンピュータの接続テスト
- ▶ ②アプリケーション開発
- ▶ ③DirectXライブラリを利用して描画処理を高速化
- ▶ ④重心計算の分散処理

今後の作業

- ▶ ①光学系が完成後，補償光学プログラムを作成
 - ▶ 作成したZernike関数のチェック
- ▶ ②プログラムのデバッグ
- ▶ ③制御工学理論を用いた処理を追加

極限補償光学 AO制御実験



大阪電気通信大学 工学部
電子機械工学科
中村 祐一