

# Towards Adaptive Scheduling for Real-Time Image Processing

Neri Marschik, Mario Speckert, Thomas Ihme

AMS 2012, September 26-28

Stuttgart, Germany





## Outline

1. The real-time image processing problem
2. Anytime Scheduling
  - Anytime principle, task pair concept
3. Image processing as anytime task
  - SURF algorithm, scheduling statistics
4. Application behavior with different timing constraints
  - Measurements and observations
5. Proposal for adaptive scheduling
6. Results



## The Real-Time Image Processing Problem

### **Problem statement**

Real-time: Guarantee of a response within a given time

Image processing: large data sets, complex algorithms

- prediction of runtime impossible or with unrealistic results
- Processor architecture (e.g. cache) maximizes *throughput*, not *predictability*
- Data dependent run-time prevents WCET modeling

Robustness through guarantee of maximum runtime

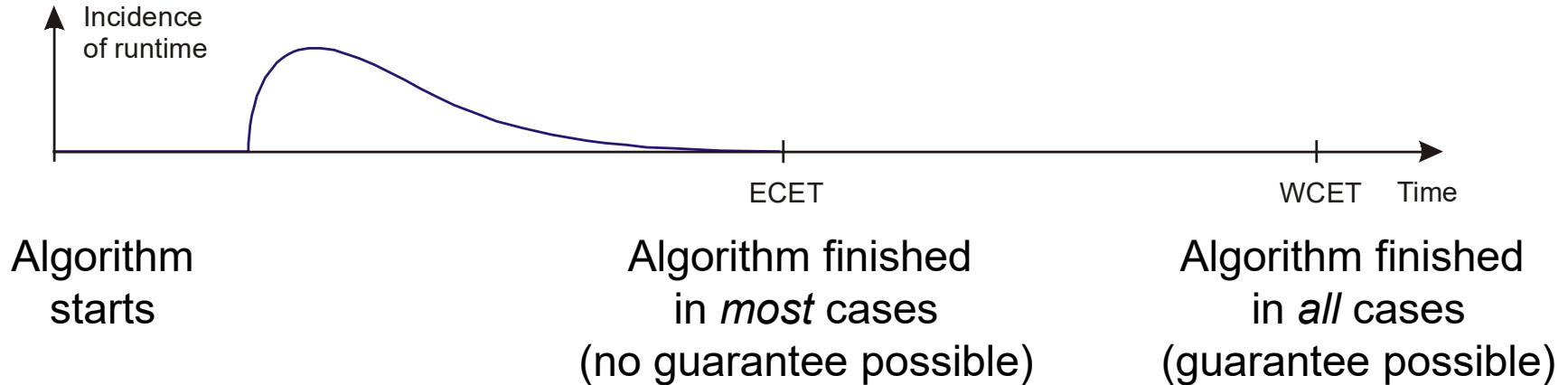
- Anytime algorithms and real-time scheduling
- High-performance computing → not considered here

Scheduling based on task pairs to prevent deadline violation



# The Real-Time Image Processing Problem

## Task behavior and scheduling needs



- If task not finished → exception handling that collects results and resets the task
- Anytime principle that provides first results during run-time
- Start and exception handling initiated by scheduler



## Anytime Scheduling

### **Anytime principle**

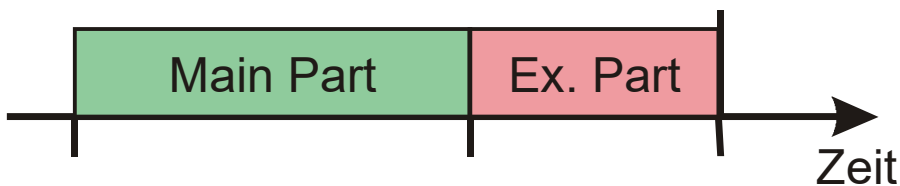
- Image processing algorithms usually provide results at its end
- New strategy: results should be available even if algorithm is stopped and not finished completely → modified image processing algorithms
- Results available except a short period after start
- When first results are available, they will be improved over time
- Algorithm may be stopped even the run is not complete → results are always available
- Modification of the SURF algorithm into an Anytime-SURF algorithm
- Similar to the imprecise computation concept, but here in combination with a scheduling concept aiming to handle exceptions



# Anytime Scheduling

## Task Pair Concept

- Clear division of applicational functionality and scheduling.
- Task consists of a task pair (TP) with common deadline (DL)
- Task-Pair consists of Main-Part (MP) and Exception-Part (EP)
- MP: Application, EP: saves results and ensures save state



$$RT_{MP}$$

$$DL_{MP}$$

$$ECET_{MP} \leq DL_{MP} - RT_{MP}$$

$$RT_{EP}$$

$$DL_{EP}$$

$$WCET_{EP} \leq DL_{EP} - RT_{EP}$$

$$RT_{TP}$$

$$DL_{TP}$$

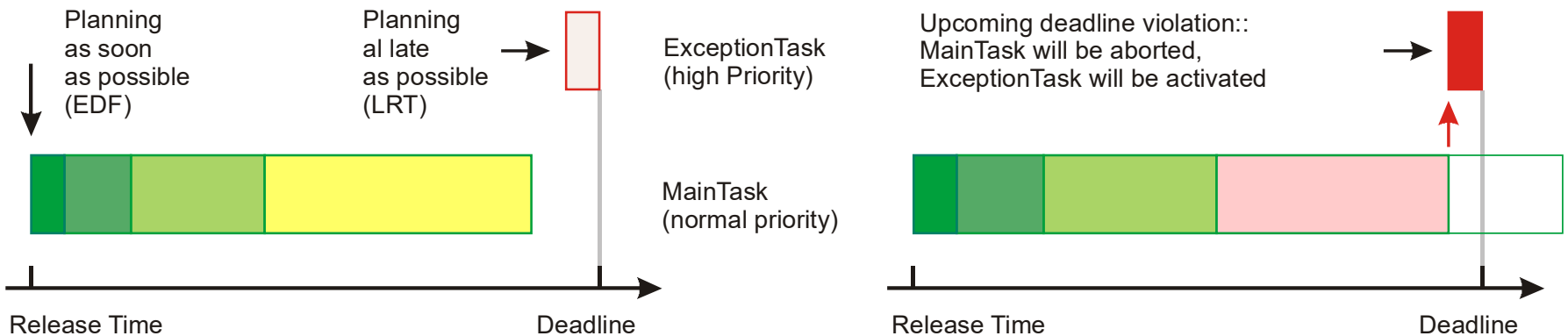
$$P_{TP} = DL_{TP} - RT_{TP}$$



# Anytime Scheduling

## Task Pair Concept

- Main-Part (normal priority)
  - Runtime based on Expected Case Execution Time (ECET)
  - Planning as early as possible (EDF)
- Exception-Part (high priority)
  - Short as possible, runtime based on worst case execution time (WCET)
  - Planning as late as possible (LRT)



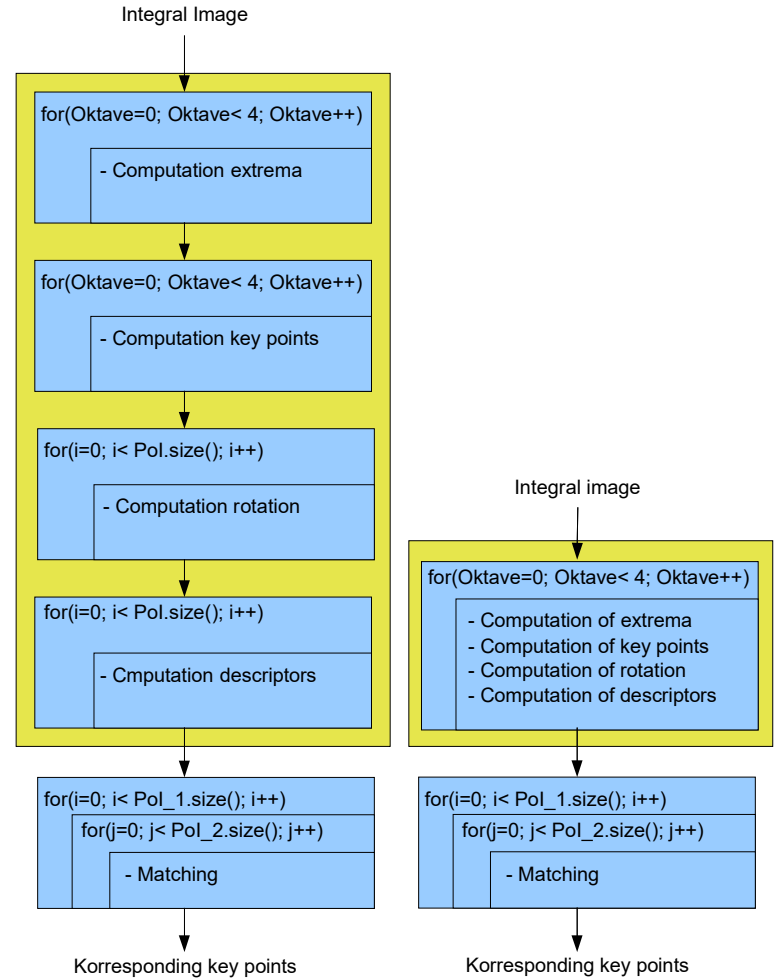




# Image processing with anytime task

## SURF Anytime Modification

- Concentration of all necessary calculations for each octave
- After finishing the first octave, first results will be available
- Modification of the sequence: start with the largest octave (may be compared with low resolution → quick computation)

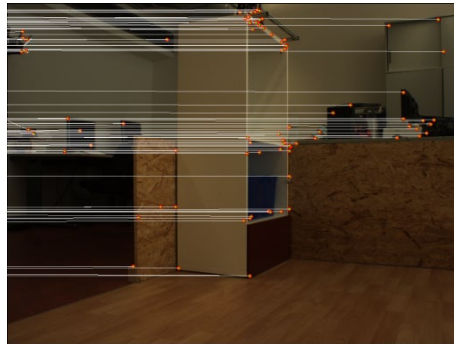
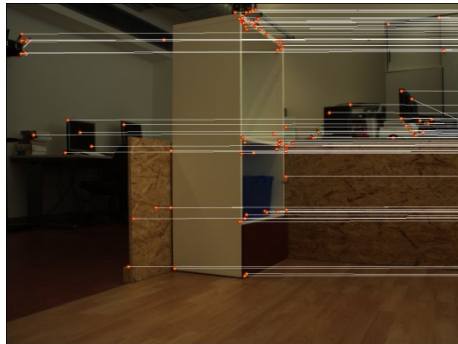




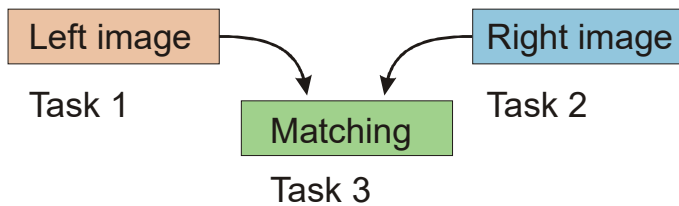


# Image processing with anytime task

## Experiment Set-Up and Application



Example of an image pair



Application tasks



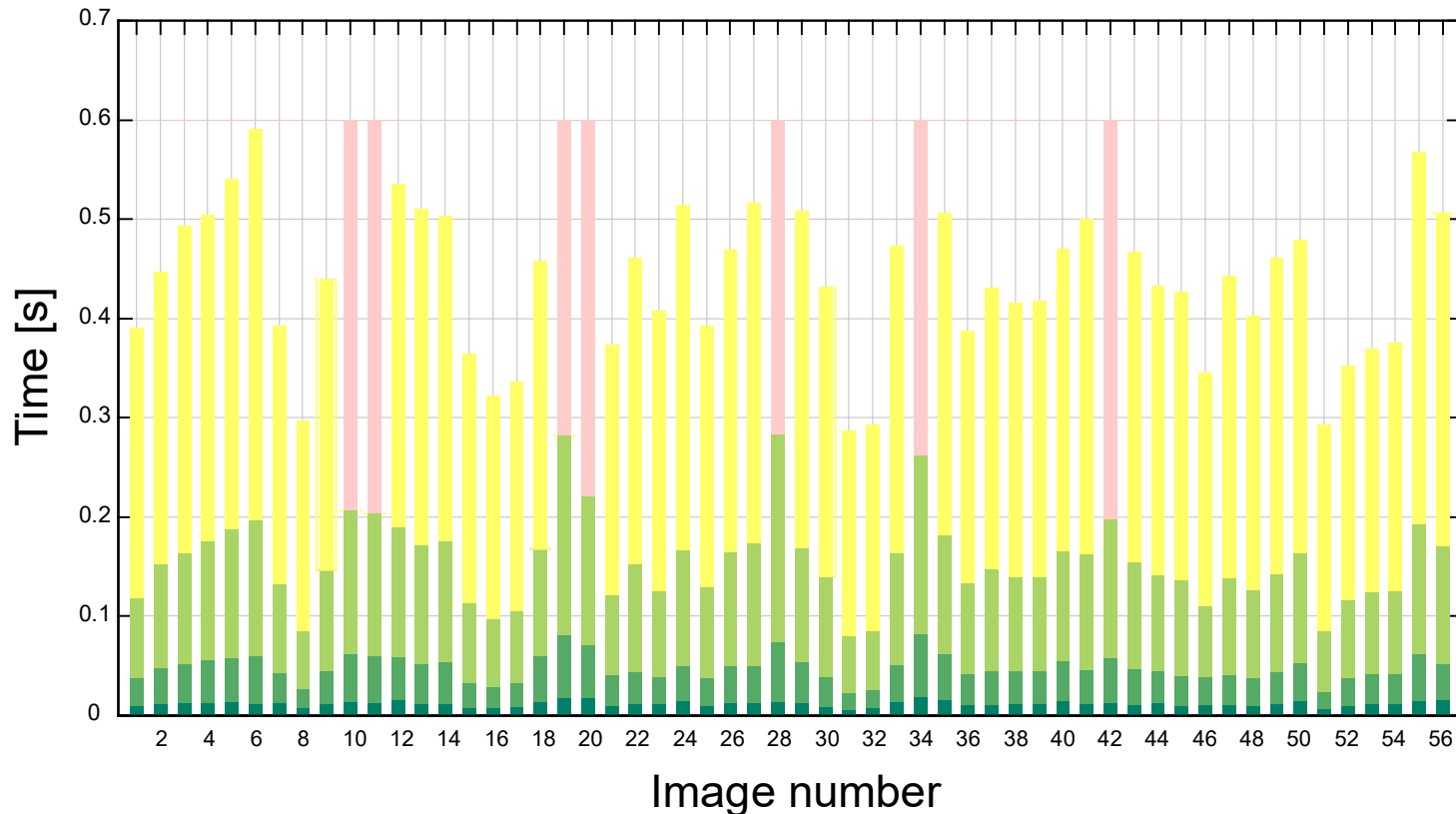
Experimental set-up



# Image processing with anytime task

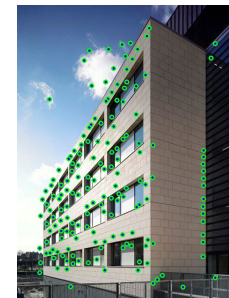
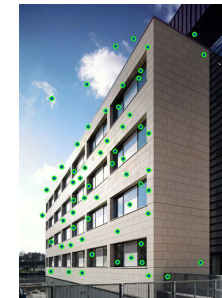
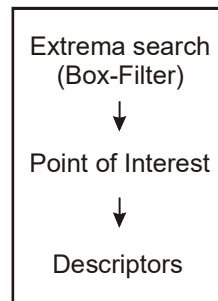
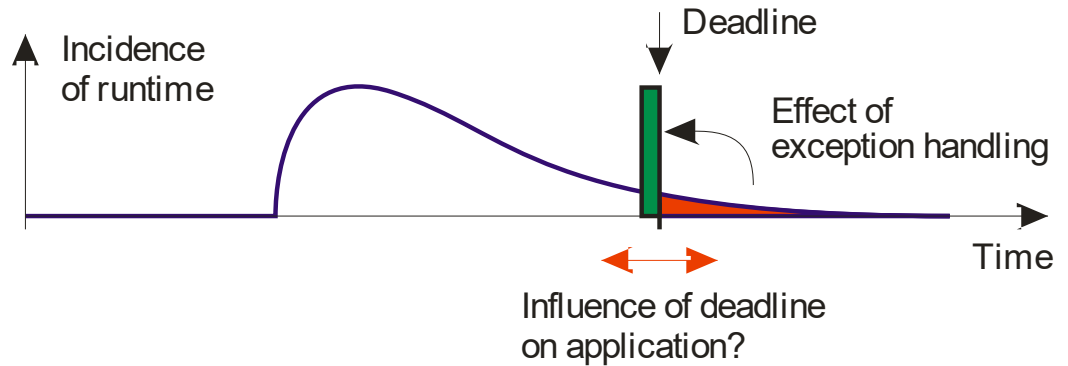
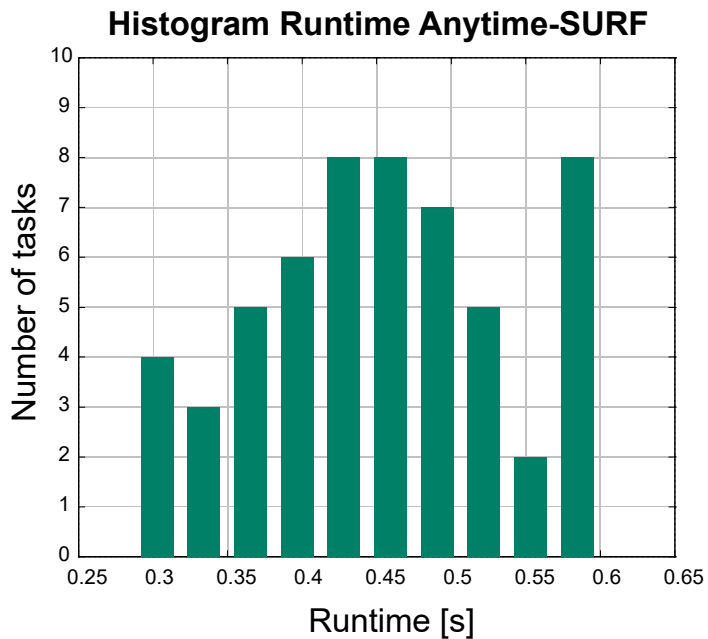
## **SURF algorithm – task runtime**

**Computation time octaves, left images (1280x960 px), deadline at 0,6s**



# Image processing with anytime task

## SURF algorithm and scheduling statistics

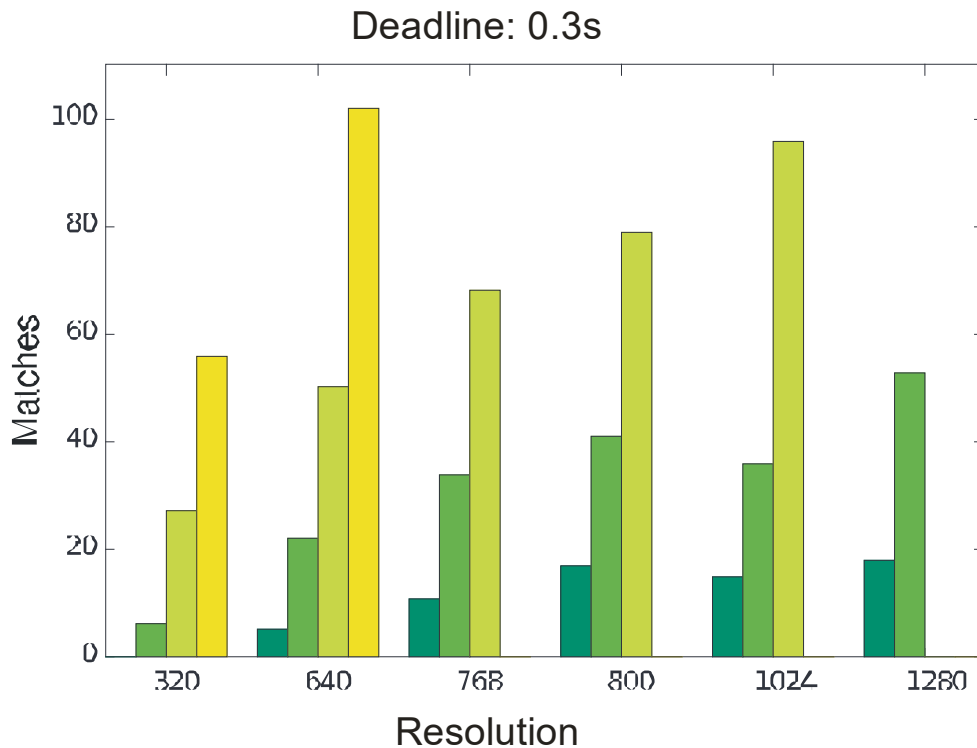


Repeating for 4 octaves



# Application behavior with different timing constraints

## Measurements and observations

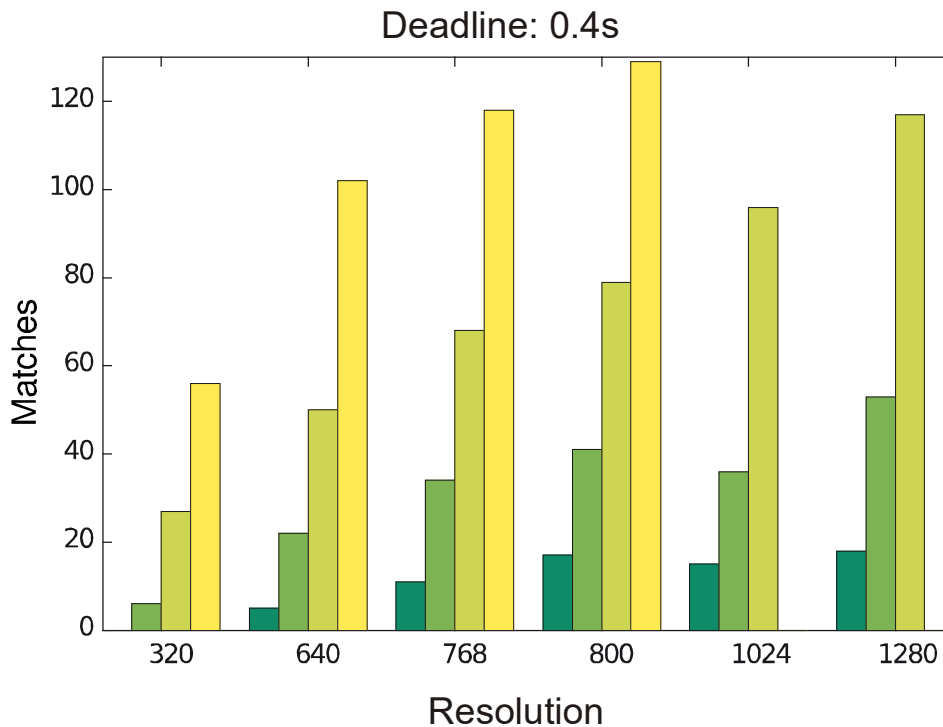


- Difficult to define „best“ features
- Count of corresponding features between two stereo images
- Average matches on a series of indoor images
- Most matches while complete run on lower resolution



## Application behavior with different timing constraints

### Measurements and observations

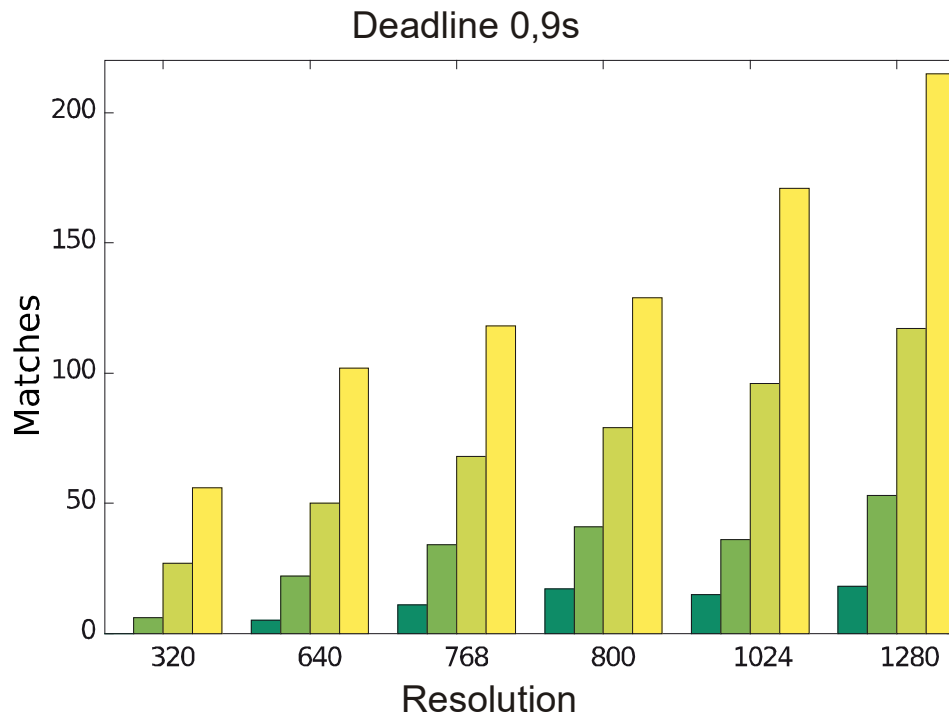


- Another deadline
  - Same result: most matching features found on lower resolution while complete run
  - Processed octaves are more important than resolution
- Adaptation of resolution necessary



## Application behavior with different timing constraints

### Measurements and observations



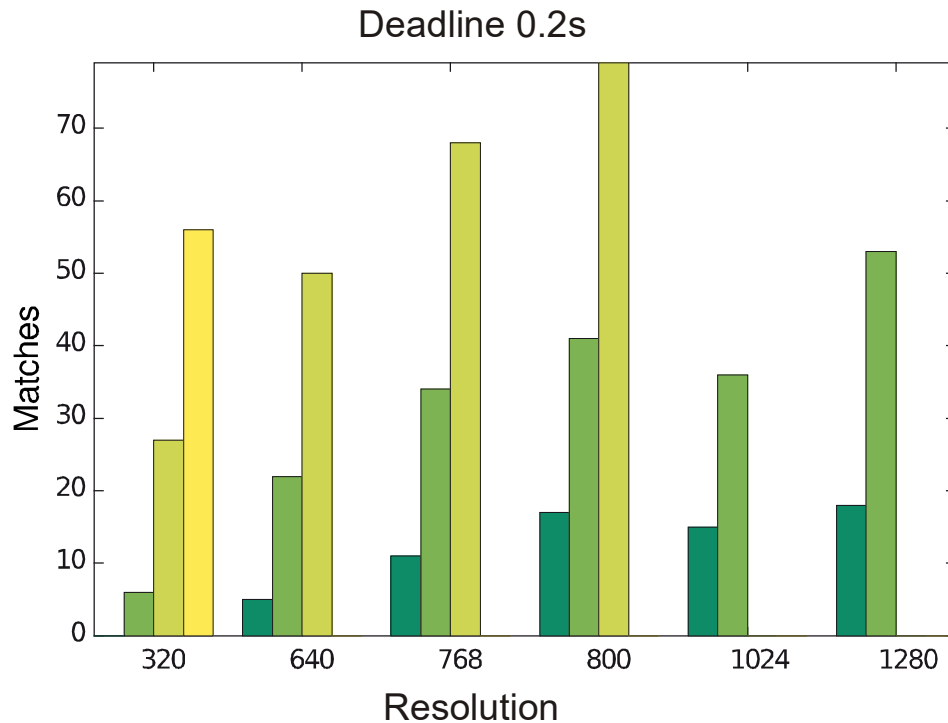
- Very late deadline: all tasks can finish regularly
  - Highest count of matching features at highest resolution
- Always use highest resolution that allow a complete run





# Application behavior with different timing constraints

## Measurements and observations



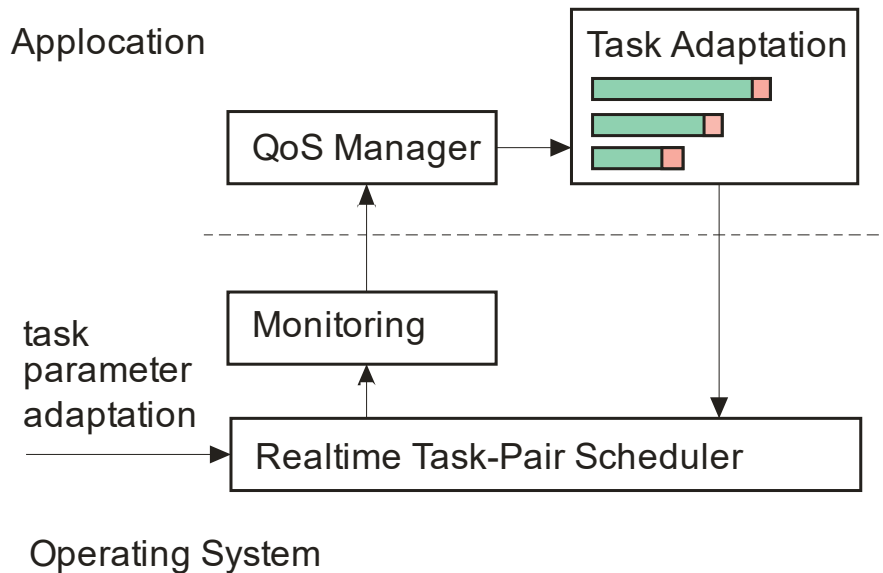
- Very early deadline: only smallest resolution can finish
  - Count of matching features at lowest resolution is lower than with incomplete run at higher resolution
- Resolution should not be too low, even if it provides a complete run
- use higher resolution





# Task adaptation

## Proposal for task adaptation



- Monitoring component observes task completion
- Quality of service manager selects task parameters (e.g. resolution) or task variants based on monitoring results
- Ensuring separation of (hard) real-time scheduling and application
- Adaptation of task parameters during run-time



## Results

- Task-pair scheduling enables real-time operation of image processing tasks, based on anytime concept
- Not limited to SURF algorithm, works e.g. for block matching
- With exception tasks based on ECET, without exemption task only WCET possible
- Quality of results may increased by adjusting processing parameters dependent on available computing time
- Task monitoring provides statistical information of completed tasks
- QoS manager adjusts task processing parameters for best results
- Adjusting task parameters with less impact on processing results