# RESOURCE USAGE PREDICTION FOR GROUPS OF DYNAMIC IMAGE-PROCESSING TASKS USING MARKOV MODELING

*Rob Albers* [a,b], *Eric Suijs* [b] *and Peter H.N. de With* [a,c]

[a] Eindhoven University of Technology, PO Box 513, 5600 MB, The Netherlands,
[b] Philips Healthcare, X-Ray R&D, PO Box 10.000, 5680 DA Best, The Netherlands,
[c] CycloMedia Technology, PO Box 68, 4180 BB Waardenburg, The Netherlands.

## ABSTRACT

With the introduction of dynamic image processing, such as in image analysis, the computational complexity has become data dependent and memory usage irregular. Therefore, the possibility of runtime estimation of resource usage would be highly attractive and would enable Quality-of-Service (QoS) control for dynamic image-processing applications with shared resources. A possible solution to this problem is to characterize the application execution using model descriptions of the resource usage. In this paper, we attempt to predict resource usage for groups of dynamic image-processing tasks based on Markov-chain modeling. As a typical application, we explore a medical imaging application to enhance a wire mesh tube (stent) under X-ray fluoroscopy imaging during angioplasty. Simulations show that Markov modeling can be successfully applied to describe the resource usage function even if the flow graph dynamically switches between groups of tasks. For the evaluated sequences, an average prediction accuracy of 97% is reached with sporadic excursions of the prediction error up to 20-30%.

***Index Terms***— Video signal processing, Software performance, Multiprocessing, Object recognition, Stochastic approximation.

## 1. INTRODUCTION & MOTIVATION

As the number of applications featuring dynamic image processing is increasing steadily, this poses new requirements on the system design. With dynamic image-processing applications, such as in image analysis, the computational complexity has become data dependent and memory usage irregular. Detailed know-how of specific application aspects, such as data-driven complexity and the corresponding memory requirements is relevant for optimal mapping of tasks on a computing platform and optimizing the performance. In order to achieve this, performance prediction may be applied in the form of modeling, in order to guide the mapping and implementation. This paper concentrates on achieving sufficient accuracy in the modeling for applications featuring dynamic execution of tasks. The secondary objective is to use the models for the runtime estimation of the resource usage. In this way, the model descriptions can be used as a prediction for resource planning and possibly the corresponding quality-of-service control of background tasks, avoiding quality degradation, deadline misses or even system breakdowns due to resource overloads [1].

Several techniques are reported in the literature for performance prediction of parallel applications. Thiele [2] describes a analysis method based on real-time calculus. However, it is not suitable for data-dependent processing tasks. Gautama [3] presents an analytical approach for parallel applications having stochastic execution times of workloads. The solution is not able to characterize any long-term dependencies on the input data. Fritzsche *et al.* [4] describes a performance prediction method based on deterministic models. Poplavko [5] and Pastrnak [6] describe a scenario-based prediction paradigm. It is based on the observation that dynamic behavior is typically composed of a limited number of sub-behaviors, called scenarios. In our case, we study a mixture of the previous features and new aspects. A difference is that our application is not based on streaming video, but involves image analysis, which has a more dynamic nature and will be discussed below. The dynamics come from properties within the video processing algorithm, rather than a quality control unit. Secondly, the dynamic decision making is based on outcomes of the image analysis process which heavily depend on the video data. Tasks in the analysis cannot be easily switched off since that would lead to an incomplete or unacceptable result.

In our study, we explore a medical imaging function to enhance a moving wire mesh tube (stent) under X-ray fluoroscopy imaging during a interventional angiography procedure [7, 8, 9]. Because physicians must see their actions directly on the screen (eye-hand coordination), a low latency is a key requirement for the imaging application. A single processor system cannot cope efficiently with the computational complexity of such an advanced application. We assume a chip-multiprocessor (multi-core) system as the target platform. In the above application, modeling of resources is complicated because depending on the image content and intermediate analysis result, the algorithm may switch to a different group of processing tasks. Therefore, the resource-

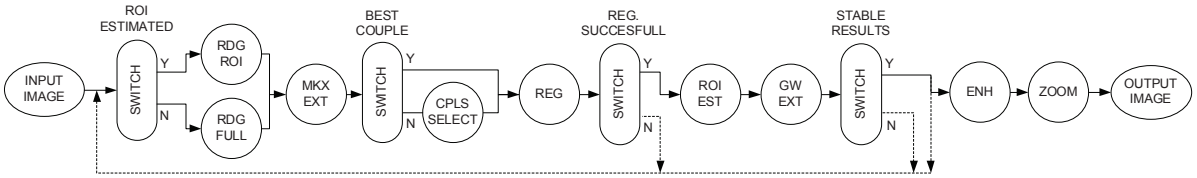


**Fig. 1**. Flow graph for motion-compensated stent enhancement.

usage model is based on stochastic properties and should handle abrupt changes in behavior and statistics. The modeling is reported here for the computation and is under study for cache memory and communication bandwidth. The modeling technique can potentially be used for alternative applications using image analysis, such as in surveillance systems etc.

This paper is organized as follows. In Section 2, the characteristics of the application are described. Section 3 introduces the prediction model including the estimation technique coping with the dynamic behavior. Section 4 presents the obtained results and the last section gives conclusions.

## 2. MEDICAL IMAGE-ANALYSIS APPLICATION

Coronary angioplasty is a catheter-based procedure performed by an interventional cardiologist in order to open up a blocked coronary artery and restore blood flow to the heart muscle. Angioplasty is used as an alternative treatment to coronary artery bypass surgery in more than half the cases. Following balloon angioplasty, a wire mesh tube (stent) can be placed to keep the artery open. The correct deployment of a stent in the coronary arteries is important for ensuring the efficacy of drug-eluting stents. Image analysis and motion-compensation techniques can improve the visualization and measurement of intracoronary stents in X-ray angiography, thereby making it easier to achieve optimum and complete stent deployment, potentially eliminating the need for additional procedures, such as intravascular ultrasound. In this paper, we explore a medical-imaging application to enhance moving stents under X-ray fluoroscopy imaging during a live interventional angioplasty procedure [1].

Motion-compensated stent enhancement consists of several steps, as depicted in Figure 1. The presented flow graph is based on a cascading of four stages which are individually described in [10, 11, 12, 13]. After stent placement, the candidate balloon markers are detected in the image using an automatic marker-extraction algorithm. Ridge detection (RDG) and filtering is applied on the input images such that all other structures, except candidate balloon markers, are removed. Subsequently, marker extraction (MKX EXT) selects punctual dark zones contrasting on a brighter background as candidate markers. Based on *a-priori* known distances between the balloon markers, couples selection (CPLS SELECT) selects the best marker couple from the set of candidate couples. Subsequently, temporal registration (REG) to align respective markers in selected image frames, is based on a motion criterion, where a temporal difference is performed between two succeeding images of the sequence. A Region of Interest is estimated in the original image (ROI EST), where the markers have previously been detected. The guide wire is detected by a ridge filter in guide-wire extraction (GW EXT). If the markers of a possible couple are situated on a track corresponding to a ridge joining them (the guide wire), this is the indication that the results obtained by automatic marker extraction are found stable. Enhancement (ENH) of the stent is performed by temporal integration of the registered image frames according to the balloon markers. The output is presented by zooming (ZOOM) in the ROI containing the stent. The described application is dynamic in three major aspects: (1) At the start, a ROI of variable data-dependent size is chosen for further analysis, and (2) at every stage, a switch function selects a specific flow graph, depending on the previous stage(s). Moreover (3), some of the internal flow graphs require a variable processing intrinsically.

[1]The application is commercially available as StentBoost or IC Stent.

## 3. PREDICTION MODEL OF EXECUTION TIME

In this section we will setup an accurate performance-prediction model for each task that has to be executed on the target platform. This prediction model should be able to follow the dynamics in the processing over time with sufficient accuracy. The application execution is characterized using model descriptions of the resource usage, with the main focus on computations in this paper. For each task in the flow graph, we create a prediction model, based on the study of the algorithm and experiments. The marker extraction, registration, ROI estimation, enhancement and zooming functions are independent of the video content or size of the images. The prediction can be defined with a constant value. There are four data-dependent switch statements in the flow graph. The current state is based on information from previously processed video frames and can be described with a state table. Each switch can signal tasks to process for example only on a region of interest of the video frame or even skip processing. The ridge detection, couples selection and guide-wire extraction have a resource usage that is highly correlated with the video content. Modeling and prediction of the computation time for these functions is less straightforward.

We have considered several options for the modeling of the application. As a first solution, we investigated literature

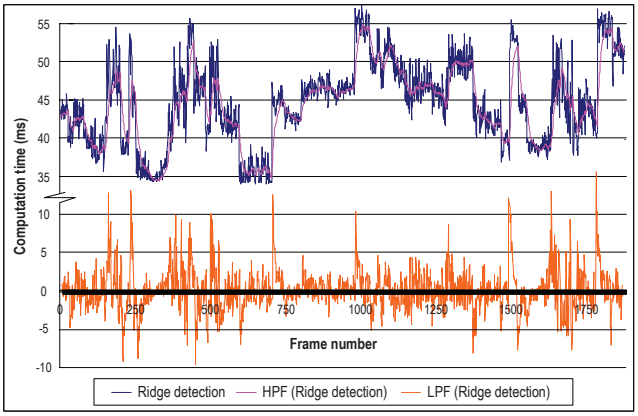


**Fig. 2**. Computation time for the *Ridge-detection* task.

on video traffic modeling [14]. Most of the papers deal with Markov-chain approaches since the estimation of the model parameters is straightforward and there is a large number of analysis techniques available. The main disadvantage of these models are the exponentially decaying autocorrelation functions of the generated sequence. This leads to inaccurate performance estimates if long-term correlation properties exist of the video sequences. Similar to Markov-chain models, autoregressive models suffer from this difficulty. For more complex application models, the estimation of the model parameters is often more difficult than in the Markov-chain case. To deal with long-term correlated frames, higher-order probabilistic processes can be used, but the state space will grow exponentially. An alternative view on the modeling of the system behavior is to consider the timing statistics of the video frames in two categories, as a result from mapping the algorithms on a platform. Hence, we then investigate short-term and structural fluctuations in processing time on the platform. Short-term fluctuations can be caused by cache misses or the overhead imposed by task switching and control. Structural fluctuations are caused by the dependency of the processing time of the tasks on the video content itself over a longer time period. This points to the direction of splitting the computational statistics in categories.

As a consequence of the previous discussions, we have adopted a concept where the long-term statistics are decoupled from the short-term stochastic behavior by employing different models for those statistics.

*Short-term data correlations*. We try to describe the prediction model for the short-term *data-dependent* tasks as a probabilistic process such as a finite state Markov chain. A first-order Markov chain is by definition memoryless, where in the model it is implicitly assumed that the processing times of successive frames are independent. Based on computation of the autocorrelation function, we have concluded that couples selection (CPLS SEL) and guide-wire extraction (GW EXT) tasks can both be modeled with Markov chains. However, Markov-chain prediction falls short if processing times between video frames are correlated over a longer time period. Next, we will describe the modeling of long-term structural data dependencies.

*Long-term data correlations*. We consider the prediction model to consist of long-term low-frequency fluctuations, around which short-term high-frequency fluctuations can take place. Discrimination between the low and high-frequency part can be made by various types of filters, such as Finite Impulse Response (FIR) or Infinite Impulse Response (IIR) filters. We apply the Exponentially Weighted Moving Average (EWMA) filter. As this IIR filter weights recent inputs more heavily than long-term previous ones, it adapts more quickly to the input signal compared to FIR filters. The EWMA filter is defined by:

$$y(t_k) = (1 - \alpha) \times y(t_{k-1}) + \alpha \times t_k \qquad (1)$$

Given the separation of correlation behavior, the short-term fluctuations are modeled with Markov chains. We have validated the applicability of the Markov-chain modeling by analyzing the autocorrelation function. In Fig. 2, the computation-time statistics for the ridge-detection (RDG FULL) task are shown. To model the computation time for the current video frame, the output of the EWMA filter is used for long-term behavior prediction. On top of that, a Markov chain predicts the short-term fluctuations in computation time. The state-space description can be generated by analyzing the computation time over a long time period. The number of states $M$ is $C_{max}/\sigma_C$, where $C_{max}$ denotes the largest measured value and $\sigma_C$ the standard deviation. We have experimentally evolved to a model with approximately $2M$ states to obtain sufficient accuracy. The quantization intervals are adaptively chosen such that each interval contains on the average the same amount of samples. The entries of the transition probability matrix $\{P_{ij}\}$ are estimated by

$$P_{ij} = n_{ij}/(\sum_{k=1}^{M} n_{ik}), \qquad (2)$$

where $n_{ij}$ denotes the number of transitions from interval *i* to interval *j*.

Data-dependent switch statements in the flow graph can cause the total processing time to change rather abruptly. For example, the first switch in the flow graph, can select the RDG task to operate only on a Region-Of-Interest (ROI) instead of the full video frame. Other switch statements trigger or cancel tasks to be executed. The switches are controlled with information extracted from the previously processed video frames, and stored in a state table. At the start of processing each new video frame, the state can be extracted in advance. By exploiting this information prior to the actual processing of the task graph, the prediction model is made adaptive to dynamic changes in the data flow. This part corresponds to the scenario-based switching in [5].

Processing-time statistics for different region-of-interest sizes show that the RDG task has a linear dependency on the

| | s0 | s1 | s2 | s3 | s4 | s5 | s6 | s7 | s8 | s9 |
|---|---|---|---|---|---|---|---|---|---|---|
| **s0** | 0.51 | 0.19 | 0.16 | 0.03 | 0.03 | 0.02 | 0.01 | 0.01 | 0.01 | 0.01 |
| **s1** | 0.21 | 0.21 | 0.17 | 0.15 | 0.07 | 0.08 | 0.02 | 0.06 | 0.00 | 0.03 |
| **s2** | 0.04 | 0.08 | 0.33 | 0.13 | 0.16 | 0.10 | 0.04 | 0.06 | 0.02 | 0.04 |
| **s3** | 0.04 | 0.05 | 0.19 | 0.16 | 0.20 | 0.14 | 0.06 | 0.08 | 0.04 | 0.04 |
| **s4** | 0.03 | 0.04 | 0.16 | 0.15 | 0.23 | 0.12 | 0.09 | 0.12 | 0.03 | 0.02 |
| **s5** | 0.03 | 0.05 | 0.11 | 0.15 | 0.18 | 0.18 | 0.15 | 0.09 | 0.05 | 0.03 |
| **s6** | 0.02 | 0.02 | 0.09 | 0.08 | 0.14 | 0.20 | 0.14 | 0.19 | 0.07 | 0.04 |
| **s7** | 0.02 | 0.04 | 0.08 | 0.07 | 0.10 | 0.09 | 0.16 | 0.25 | 0.12 | 0.07 |
| **s8** | 0.03 | 0.01 | 0.13 | 0.03 | 0.09 | 0.08 | 0.14 | 0.25 | 0.07 | 0.17 |
| **s9** | 0.01 | 0.01 | 0.03 | 0.01 | 0.03 | 0.03 | 0.06 | 0.10 | 0.12 | 0.60 |

(a)

| Task | Prediction Model [ms] |
|---|---|
| RDG FULL | <Eq. 1> + Markov RDG |
| RDG ROI | <Eq. 3> + Markov RDG |
| MKX EXT | 2.5 |
| CPLS SEL | <Eq. 1> + Markov CPLS |
| REG | 2 |
| ROI EST | 1 |
| GW EXT | <Eq. 1> + Markov GW |
| ENH | 24 |
| ZOOM | 12.5 |

(b)

**Table 1**. (a) RDG transition matrix and (b) model summary.

size of the ROI. To analyze load fluctuations, caused by dependencies on the video content itself, we have subtracted a linear growth function from the obtained statistics. This function is specified by

$$y_{t_k} = 0.067 \times t_k + 20.6. \quad (3)$$

For the remaining data-dependent fluctuations after subtraction, we analyzed the autocorrelation function. As the function has a exponential decay, it can again be described with a Markov chain. As the fluctuations are in the same order as the high-frequency behavior described in the previous section, we have included these statistics to the Markov state-generation process, to generate a single Markov chain for the ridge-detection task.

## 4. EXPERIMENTAL RESULTS

In this section, we show the prediction results for all tasks in the flow graph of Fig. 1, based on the short-term and long-term modeling techniques from the previous section. Based on known methods for modeling video traffic performance in networks, we applied probabilistic models for predicting the execution time of data-dependent tasks. For tasks with long-term dependencies on input data frames, we have applied filtering to make the signal suitable for probabilistic modeling. Data-dependent switch statements in the task graph are modeled with state tables. Changes in the processing granularity (ROI processing), are modeled with linear functions. Computation time statistics are obtained by profiling the executed application on a chip-multiprocessor platform[2]. For training the stochastic models, we used a data set of 37 video sequences of in total 1,921 video frames. In the training set, different scenarios exist for which the algorithms can adapt to. In Table 1a, the Markov transition matrix is shown for the ridge-detection task. Similar matrices are generated for the couples selection and guide-wire extraction tasks. A summary of the prediction models can be found in Table 1b. For the test sequences (Fig. 3), an average prediction accuracy of 97% is reached with sporadic excursions of the prediction error up to 20-30%.

[2]A shared-memory, quad processor-core system, 2.33 GHz, 4 GB RAM.

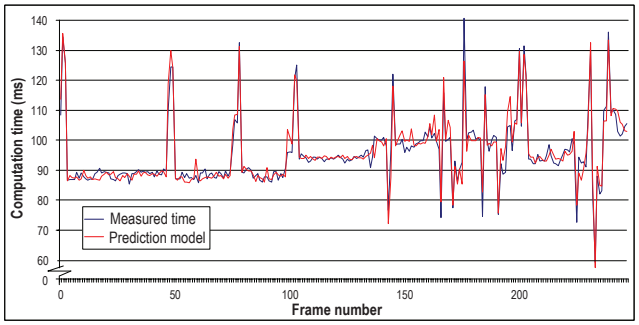


**Fig. 3**. Prediction model results vs. actual computation time.

## 5. CONCLUSIONS

In this paper, we created a prediction model for groups of dynamic image-processing tasks based on Markov-chain modeling. Resource modeling is relevant for optimal mapping of tasks on a computing platform and optimizing the performance. Furthermore, the models can be used for accurate runtime estimation of the resource usage. In this way, an application manager can be initiated for resource planning and corresponding quality-of-service control of background tasks, avoiding quality degradation, deadline misses or even system breakdowns due to resource overloads [1].

Experimental results for a medical imaging function show that Markov modeling can be successfully applied to describe the resource usage function even if the flow graph dynamically switches between groups of tasks. Results show an average prediction accuracy of 97% with sporadic excursions of the prediction error up to 20-30%. The modeling technique can potentially be used for alternative applications using image analysis, such as in surveillance systems etc.

### 6. REFERENCES


[1] C.C. Wüst *et al.*, "Qos control strategies for high-quality video processing," *Real-Time Syst.*, vol. 30, no. 1-2, pp. 7–29, 2005 .

[2] L. Thiele, E. Wandeler, and S. Chakraborty, "Performance analysis of multiprocessor dsps," *Signal Proc., IEEE*, vol. 22, no. 3, 2005 .

[3] H. Gautama and A.J.C. van Gemund, "Performance prediction of data-dependent task parallel programs," in *Euro-Par*, 2001, vol. 2150 of *LNCS* , pp. 106–116.

[4] P. Fritzsche *et al.*, "A performance prediction methodology for data-dependent parallel applications," *Cluster Comp., IEEE Int. Conf. on*, pp. 1–8, Sept. 2006 .

[5] P. Poplavko, T. Basten, and J. van Meerbergen, "Execution-time prediction for dynamic streaming applications with task-level parallelism," in *DSD*, 2007 .

[6] M. Pastrnak and P.H.N. de With, "Data storage exploration and bandwidth analysis for distributed mpeg-4 decoding," *Cons. Elec., IEEE Int. Symp. on*, 2004.

[7] "Image guidance method coronary stent deployment," Patent application WO/2002/002173 .

[8] "X-ray identification of interventional tools," Patent application US/2008/137923.

[9] V. Bismuth and R. Vaillant, "Elastic registration for stent enhancement in x-ray image sequences," *Image Processing: ICIP, IEEE Int. Conf. on*, Oct. 2008.

[10] "Viewing system for control of ptca angiograms," Patent application WO/2005/104951 .

[11] "Medical viewing system and method for spatially enhancing structures in noisy images," Patent application US/2005/058363 .

[12] "System and method for enhancing an object of interest in noisy medical images," Patent application WO/2004/066842 .

[13] "Medical viewing system and method for enhancing structures in noisy images," Patent application WO/2003/045263 .

[14] V.S. Frost and B. Melamed, "Traffic modeling for telecommunications networks," *Comm. Mag., IEEE*, vol. 32, no. 3, pp. 70–81, Mar 1994 .