COMPLEXITY PREDICTION OF AUTOMATIC IMAGE REGISTRATION:
A CASE STUDY ON MOTION-COMPENSATED DSA

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 Eindhoven, The Netherlands,
b Philips Healthcare, Interventional X-Ray, PO Box 10.000, 5680 DA Best, The Netherlands,
c CycloMedia Technology, PO Box 68, 4180 BB Waardenburg, The Netherlands.

ABSTRACT
Real-time video and Quality-of-Service aspects play an increasing role in the development of medical imaging systems. To avoid resource overload and to guarantee the throughput of dynamic applications, we present a method for complexity prediction of image registration and motion-compensation algorithms, which can have a highly dynamic nature at runtime. As a case study, we explore a medical imaging function to reduce motion-artifacts in X-ray Digital Subtraction Angiography (DSA). Complexity prediction is based on motion estimation, prior to the actual image registration. Experimental results show that it is possible to model a dynamic content-dependent processing task with a high accuracy (95%, standard deviation 5%), thereby facilitating a higher quality for remaining tasks and well defined options for QoS.

Index Terms— Medical Imaging, Quality-of-Service, Image Motion Analysis, X-ray Imaging, Multiprocessing

1. INTRODUCTION

Further advances in medical imaging involve live viewing and editing of high-definition content, intelligent behavior recognition and computer analysis of diseases within the medical domain. With the growing complexity of processing and software-based implementations, the immanent limitations become more apparent: compute-intensive tasks require so much power from the platform that not all remaining tasks can be performed at the desired quality, leading to arbitrary task switching and thus ungraceful degradation of quality. Alternatively, sometimes application-specific dedicated hardware is employed which constrains the general-purpose nature of the architecture.

In professional medical imaging, multiple imaging applications are executed in parallel on the same system, thereby sharing platform resources. To optimize system resources and simultaneously preserve the image quality, real-time processing aspects and Quality-of-Service (QoS) considerations should be applied [1]. This allows close to average-case resource allocation and graceful degradation of quality if deadlines of background applications are occasionally missed. This problem statement has already been addressed in several fields of applications, such as high-performance computing and multimedia [2]. We borrow the developed concepts for these fields and impose them on professional medical imaging and processing, where architectures have different cost constraints and flexibility are of key importance.

We restrict ourselves to X-ray interventional imaging with a case study on image registration for Digital Subtraction Angiography (DSA) [3]. Eye-hand coordination is an important aspect in clinical practice, requiring low-latency imaging. Automatic motion-artifact correction for X-ray DSA is a powerful image registration technique based on template matching [4]. With template matching, a search is executed to match parts of an image to a visual template. As motion-based template matching is highly variable with the image motion, the computing demands and latency vary accordingly, which influence the other applications running in parallel. To control the quality and latency, an ahead-of-time knowledge of the required resource budget is required.

Let us discuss the above fluctuations in more detail. As some steps are content dependent, images with severe motion artifacts will require more processing time than images with only small motion artifacts. From a system architectural perspective, when multiple applications are executed on the same medical system, functions with dynamic resource usage hamper the latency and throughput of other critical applications when resources are shared. Currently, the system designer has no other alternative than to assign a worst-case budget of required resources and over provision the processing platform. To overcome this drawback, we present a method for runtime complexity prediction of motion-compensated DSA, to ensure sufficient response time for medical interaction, while still using a processing platform, dimensioned for average-case resource requirements. The result is a predictive knowledge of the required resource budget.

Earlier work has shown that future systems employing these kind of methods can provide higher QoS by more efficiently utilizing the available system resources [5]. The proposed solution enables close-to-average case resource allocation and graceful degradation of quality. As this technique
3. Given the set of control points, $P = \{p_i\}$, construct a triangular mesh, $D(P)$, by using the incremental Delaunay triangulation algorithm proposed by Watson [7] (Fig. 2(b)).

4. For every image, $I_n(x, y)$, $n \in [1, N-1] \subset \mathbb{N}$, in the sequence, compute the displacements, $d_n(p_i)$, of the selected control points, $p_i \in P$, by maximizing the energy of the histogram-of-differences similarity measure, in a $K \times L$ neighborhood, using hill-climbing optimization.

5. Warp the mask image by deforming every triangle $\triangle p_{ij}k$ in the mesh $D(P)$, by using the displacements $d_n(p_i)$, $d_n(p_j)$, and $d_n(p_k)$ of the constituent control points $p_i$, $p_j$, and $p_k$, and the linearly interpolated displacements at the remaining points, $p \neq p_i, p_j, p_k$, and by using bilinear interpolation of grey values.

6. Compute the motion-corrected subtraction image, corresponding to the contrast image $I_n(x, y)$, $n \in [1, N-1] \subset \mathbb{N}$, by subtraction with the motion-corrected mask image $I_0(x, y)$ (Fig. 2(d,f)).

3. COMPLEXITY PREDICTION

To be cost effective, it is required that the applications make efficient use of the available resources on the platform. Executing a set of tasks on a multi-core processor platform requires matching the needs of every application with the available platform resources. Due to changing runtime conditions with respect to e.g. user application requirements, or having multiple simultaneously executing applications competing for platform resources, there is a need for a runtime decision-making and QoS arbitration entity: the runtime manager. Also, as an answer to the variable processing rates of applications, performance prediction may be applied in the form of modeling of the execution of the essential processing tasks, to guide the application.

For accurate resource management, static analysis alone will not solve the problem. Instead, an approach is required that combines static and dynamic analysis. To this end, we have developed a model for the DSA algorithm [4]. We require a quantitative metric for the amount of used resources, predicting the cost of the next frame in contrast to the current image frame. Parameters of the algorithm can help in establishing this predictive metric. In the following, we present the predictive algorithm in more detail and propose two metrics for the prediction of the dynamic parts. Prediction is based on estimating the motion of the incoming image frames prior to the actual image registration.

For analysis of static processing, tasks belonging to the generation of the triangular mesh of control points (Tasks 1, 2, 3), and the final image warping and subtraction task (Task 5, 6) are identified as static, having a deterministic computation time. The model can be defined with a constant value. With respect to dynamic tasks, registration of the mask image (Task 4) with the live image depends heavily on the motion error of the mask image with the processed live image data. As this task is based on template matching, the actual

Fig. 1. Flowchart for motion-compensated DSA with (a) control point selection and (b) registration and compensation of the mask and live X-ray images.

can be found in other application areas such as video coding or face recognition, our work is relevant for a broader area than medical imaging only.

This paper is organized as follows: in Section 2, we briefly present the DSA application. Complexity prediction is described in Section 3. Experimental results are explained in Section 4 and Section 5 concludes the paper.

2. DIGITAL SUBTRACTION ANGIOGRAPHY (DSA)

In DSA, a sequence of X-ray projection images is acquired during the passage of a bolus of injected contrast agent through the vessels of interest [6]. By subtracting an image acquired prior to arrival of the contrast agent (the mask image), background structures in the contrast images are largely removed. However, due to patient motion, DSA images often show motion artifacts that may hamper proper diagnosis. In order to reduce these motion artifacts, the misalignment of the successive images in the sequence needs to be determined and corrected. This alignment operation is often referred to as image registration. Many semi- or even fully automatic registration techniques have been developed over the past two decades [3]. In [4], an approach has been developed for registration of angiographic image sequences which is effective and computationally efficient.

The algorithm [4] consists of several steps (Fig. 1). Given an image sequence with $N$ images of size $M \times M$ pixels, the registration of the mask image, $I_0(x, y)$ (Fig. 2(a)), with respect to the contrast images, $I_n(x, y)$, with $n$ the image index and $n \in [1, N-1] \subset \mathbb{N}$, is accomplished by carrying out the following six steps:

1. Compute the gradient magnitude, $\| \nabla L \|$, at scale $s$, of an $M_s \times M_s$ downsized version of the mask image, $I_0(x, y)$, of the sequence, and extract potential artifact regions by means of thresholding at level $\Theta_c$.

2. Extract control points from the exposure region, $R_E$, by using the thresholded gradient magnitude, $\| \nabla L \|$, of the mask image, $I_0(x, y)$, and the control points, $p_i \in P$, of the linear mesh $D(P)$.

3. Compute the motion-corrected subtraction image, corresponding to the contrast image $I_n(x, y)$, $n \in [1, N-1] \subset \mathbb{N}$, by subtraction with the motion-corrected mask image $I_0(x, y)$ (Fig. 2(d,f)).
Fig. 2. (a) Mask image, (b) triangular mesh generation on mask, (c) unregistered DSA image without motion, (d) registered, (e) unregistered DSA image with severe motion, (f) registered.

computation time varies with the number of control points and the required amount of iterations for each matching procedure. The amount of iterations for template matching is heavily correlated with the motion between mask image and live images. Global object motion will have a higher impact on the complexity than local object motion, as the motion error and the number of matching iterations will be higher also.

Let us now discuss the creation of the complexity prediction model. For estimating the complexity of image registration, motion estimation seems to be a logical candidate. We have considered several options for motion estimation. First, we try to calculate the intensities of the pixel differences between the mask image and the live image. But with this approach, we are unable to discriminate between global differences (object motion) and local differences (injection of contrast medium). Therefore, as a second approach, we downscale the image to $8 \times 8$ pixels. We call this an image stamp. Intensity differences between the image stamp of the mask and live image are made insensitive for local injection of contrast medium and possible noise. The difference images correlate to (relatively large) object motions only. As the number of control points are not equally distributed across the image, we multiply each pixel $i$ in the image stamp with a special weight factor $\alpha(i)$, reflecting the amount of control points involved with the image stamp. Factor $\alpha(i)$ is given by:

$$\alpha(i) = 10 \times \frac{P_i}{P_{\text{max}}},$$  \hspace{1cm} (1)

where $i$ represents the current pixel value, $P_i$ the number of control points at pixel $i$, and $P_{\text{max}}$ corresponds to the maximum amount of control points per pixel. The resulting weighted pixel values are called $g$.

The similarity measure $M$ used for template matching is a very important metric in obtaining a good registration result. Many measures have been proposed for the registration of X-ray angiography images [3]. In our case study, template matching is performed based on the Energy of the Histogram (H) of Differences (EHD), specified by:

$$\mathcal{M}_{\text{EHD}}(d) = \sum_{g=-G}^{G} h^2(g),$$  \hspace{1cm} (2)

with $d$ the displacement in window $G$ for weighted pixel values $g$.

For predicting the computation complexity of template matching, we have to correlate the obtained image stamp to the actually measured computing complexity of template matching for a set of training sequences. We apply the Histogram of Absolute Differences (HAD) metric for this task, defined by:

$$\mathcal{M}_{\text{HAD}}(d) = \sum_{g=-G}^{G} |h|(g).$$  \hspace{1cm} (3)

The HAD is found to be an effective metric to correlate the actual computation time to the prediction model using motion estimation (See Fig. 3). Compared to the EHD metric, the HAD is only slightly less accurate, but computational more efficient [3]. Histogram $H_{\text{HAD}}$ consists of 20 bins with a variance $\sigma = 100$. The first bins represent small motion errors, the last bins represents large motion errors.

Template-matching-based image registration requires a more or less constant amount of iterations $c_m$ for image frames with a small amount of motion or no motion. In the case of medium-to-large object motion, the amount of iterations increases drastically. As medium and large object motion errors are located in the last bins of histogram $H_{\text{HAD}}$, these are directly correlated with the peaks in computing complexity for template matching. We use this knowledge in our complexity prediction model. The prediction algorithm is executed as follows:

(1) The algorithm $C_p$ maps the amount of bin items from histogram $H_{\text{HAD}}$ to a prediction for the computational complexity of a single pixel in the image stamp, which is described by:

$$C_p = \begin{cases} c_{p0}, & (0 \leq H(19) < 2), \\ c_{pi}, & (2i \leq H(19) < 2i + 2), & (1 \leq i \leq 5), \\ c_{p6}, & \text{elsewhere}. \end{cases}$$  \hspace{1cm} (4)
(2) Prediction coefficients $c_p$ ($c_{p0}$ through $c_{p6}$) are described by:

$$c_{p0} = \frac{c_m}{P_{total}},$$
$$c_{pi} = c_{p0} \times (1 + i/10), \quad (5)$$

where $P_{total}$ is the total amount of control points, and $1 \leq i \leq 6$. From the first frame of a particular X-ray image sequence, the constant $c_m$ is determined.

(3) The computational complexity for the entire difference image $C_{total}$ is defined as the sum of the individual pixel contributions $C_p$.

(4) For every subsequent image frame $i$ of a particular sequence, an image stamp is created, (5) similarity measure $\bar{M}_{HAD}$ is computed from Histogram $H_{HAD}$ and (6) finally, complexity prediction $C_{total}$ is defined as the sum of the individual pixel contributions $C_p$.

4. EXPERIMENTAL RESULTS

We apply the Histogram of Absolute Differences (HAD) similarity measure, to define a complexity metric for template matching. As global object motion has the highest impact on the complexity, the method first removes local motion patterns with an image pyramid, to create a downscaled image stamp. Fig. 3 shows that the prediction model has a strong spatial correlation with the actual computation time for template matching, with and without motion. Computation time statistics are obtained by profiling the application on a multicore platform. In Fig. 4, we show the experimental results applying complexity prediction. A data set of 15 patients in total 400 video frames was used. The proposed complexity prediction model shows an average accuracy of 95%, with a standard deviation of 6%.

1 A shared-memory, quad processor-core system, 2.33 GHz, 4 GB RAM.

5. CONCLUSIONS

We have presented a method for runtime prediction of the required resources of image registration for X-ray DSA, for which the computation complexity is highly content dependent. The system is based on modeling the required computations, by employing a combination of (1) motion estimation, based on the Histogram of Absolute Differences (HAD), and (2) a mapping stage, where the estimation results are correlated to the actual computation times of template-matching-based image registration. Experimental results show an average prediction accuracy of 95% with a standard deviation of only 6%. The algorithm is fast and accurate enough to be used for runtime quality control. The results in this paper show that it is possible to model a dynamic content-dependent processing task with a high accuracy, thereby facilitating higher quality for remaining tasks and well defined options for QoS.

6. REFERENCES

[7] D.F. Watson, “Computing the n-dimensional Delaunay tessel-