EURASIP Journal on Applied Signal Processing 2003:6, 514–529

Memory-Optimized Software Synthesis from Dataflow Program Graphs with Large Size Data Samples

Hyunok Oh

The School of Electrical Engineering and Computer Science, Seoul National University, Seoul 151-742, Korea
Email: oho@comp.snu.ac.kr

Soonhoi Ha

The School of Electrical Engineering and Computer Science, Seoul National University, Seoul 151-742, Korea
Email: sha@comp.snu.ac.kr

Received 28 February 2002 and in revised form 15 October 2002

In multimedia and graphics applications, data samples of nonprimitive type require significant amount of buffer memory. This paper addresses the problem of minimizing the buffer memory requirement for such applications in embedded software synthesis from graphical dataflow programs based on the synchronous dataflow (SDF) model with the given execution order of nodes. We propose a memory minimization technique that separates global memory buffers from local pointer buffers: the global buffers store live data samples and the local buffers store the pointers to the global buffer entries. The proposed algorithm reduces 67% memory for a JPEG encoder, 40% for an H.263 encoder compared with unshared versions, and 22% compared with the previous sharing algorithm for the H.263 encoder. Through extensive buffer sharing optimization, we believe that automatic software synthesis from dataflow program graphs achieves the comparable code quality with the manually optimized code in terms of memory requirement.

Keywords and phrases: software synthesis, memory optimization, multimedia, dataflow.

1. INTRODUCTION

Reducing the size of memory is an important objective in embedded system design since an embedded system has tight area and power budgets. Therefore, application designers usually spend significant amount of code development time to optimize the memory requirement.

On the other hand, as system complexity increases and fast design turn-around time becomes important, it attracts more attention to use high-level software design methodology: automatic code generation from block diagram specification. COSSAP [1], GRAPE [2], and Ptolemy [3] are well-known design environments, especially for digital signal processing applications, with automatic code synthesis facility from graphical dataflow programs.

In a hierarchical dataflow program graph, a node, or a block, represents a function that transforms input data streams into output streams. The functionality of an atomic node is described in a high-level language such as C or VHDL. An arc represents a channel that carries streams of data samples from the source node to the destination node. The number of samples produced (or consumed) per node firing is called the output (or the input) sample rate of the node. In case the number of samples consumed or produced on each arc is statically determined and can be any integer, the graph is called a synchronous dataflow graph (SDF) [4] which is widely adopted in aforementioned design environments. We illustrate an example of SDF graph in Figure 1a. Each arc is annotated with the number of samples consumed or produced per node execution. In this paper, we are concerned with memory optimized software synthesis from SDF graphs though the proposed techniques can be easily extended to other SDF extensions.

To generate a code from the given SDF graph, the order of block executions is determined at compile time, which is called "scheduling." Since a dataflow graph specifies only partial orders between blocks, there are usually more than one valid schedule. Figure 1b shows one of many possible scheduling results in a list form, where $2(A)$ means that block A is executed twice. The schedule will be repeated with the streams of input samples to the application. A code template according to the schedule of Figure 1b is shown in Figure 1c.

When synthesizing software from an SDF graph, a buffer space is allocated to each arc to store the data samples

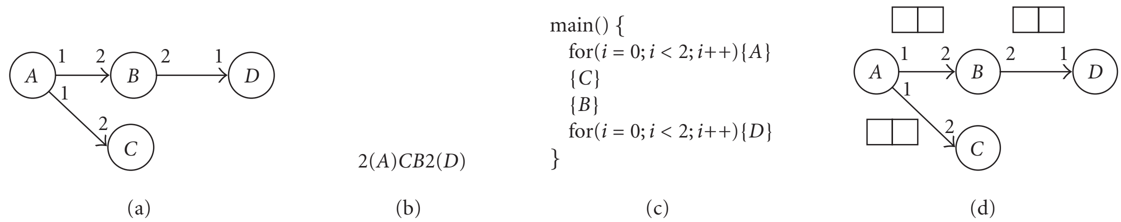


FIGURE 1: (a) SDF graph example, (b) a scheduling result, (c) a code template, and (d) a buffer allocation.

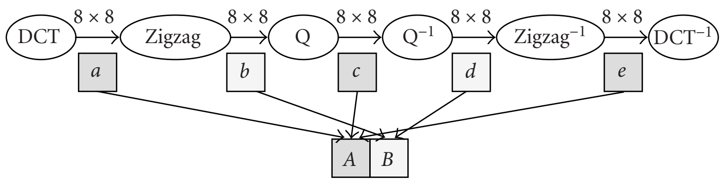


FIGURE 2: Image processing example.

between the source and the destination blocks. The number of allocated buffer entries should be no less than the maximum number of samples accumulated on the arc at runtime. After block A is executed twice, two data samples are produced on each output arc as explicitly depicted in Figure 1d. We define a buffer allocated on each arc as a local buffer that is used for data transfer between two associated blocks. If the data samples are of primitive types, the local buffers store data values and the generated code defines a local buffer with an array of primitive type data.

Required memory spaces in the synthesized code consist of code segments and data segments. The latter stores constants and parameters as well as data samples. We regard memory space for data samples as buffer memory, or shortly buffer, in this paper.

There are several classes of applications that deal with nonprimitive data types. The typical data type of an image processing application is a matrix of fixed block size as illustrated in Figure 2. Graphic applications usually need to deal with structure-type data samples that contain information on vertex coordinates, viewpoints, light sources, and so on. Networked multimedia applications exchange packets of data samples between blocks. In those applications, the buffer requirements are likely to be more significant than others. For example, the code size of H.263 encoder [5] is about 100 K bytes but the buffer size is more than 300 K bytes.

Since the buffer requirement of an SDF graph depends on the execution order of nodes, there have been several approaches [6, 7, 8] to take the buffer size minimization as one of the scheduling objectives. However, they do not consider either buffer sharing possibilities nor nonprimitive data types. Finding out an optimal schedule for minimum buffer requirements considering both is a future research topic. *In this paper, instead, we propose a buffer sharing technique for nonprimitive type data samples to minimize the buffer memory requirement assuming that the execution order of nodes is already determined at compile time*. Thus, this work is complementary to existent scheduling algorithms to further reduce the buffer requirement.

Figure 2 demonstrates a simple example where we can reduce the significant amount of buffer memory by sharing buffers. Without buffer sharing, five local buffers of size 64 ($= 8 \times 8$) are needed. On the other hand, only two buffers are needed if buffer sharing is used so that a, c, and e buffers share buffer A, and b and d buffers share buffer B. Such sharing decision can be made at compile time through lifetime analysis of data samples, which is a well-known compilation technique.

A key difference between the proposed technique and the previous approaches is that we separate the local pointer buffers from global data buffers explicitly in the synthesized code. In Figure 2, we use five local pointer buffers and two global buffers. This separation provides more memory sharing chances when the number of local buffer entries becomes more than one. If the local buffer size becomes one after buffer optimization, no separation is needed. We examine Figure 3a which illustrates a simplified version of an H.263 encoder algorithm where "ME" node indicates a motion estimation block, "Trans" is a transform coding block which performs DCT and Quantization, and "InvTrans" performs inverse transform coding and image reconstruction. Each sample between nodes is a frame of 176×144 byte size which is large enough to ignore local buffer size. The diamond symbol on the arc between ME and InvTrans denotes an initial data sample, which is the previous frame in this example. If we do not separate local buffers from global buffers, then we need three frame buffers as shown in Figure 3b since buffers a and c overlap their lifetimes at ME, a and b at Trans, and b and c at InvTrans. Even though two frames are sufficient for this graph, we cannot share any buffer without separation of local buffers and global buffers. In fact, we can use only two frame buffers if we use separate local pointer buffers. Figure 3c shows the allocation of local buffers and global

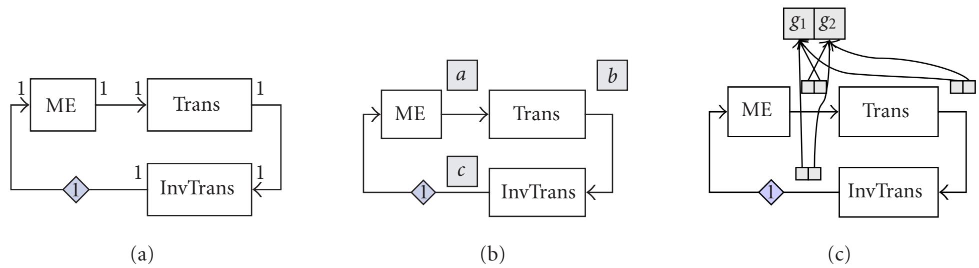


FIGURE 3: (a) Simplified H.263 encoder in which a diamond between InvTrans and ME indicates an initial sample delay, (b) and (c) a minimum buffer allocation without and with separation of global buffers and local buffers, respectively.

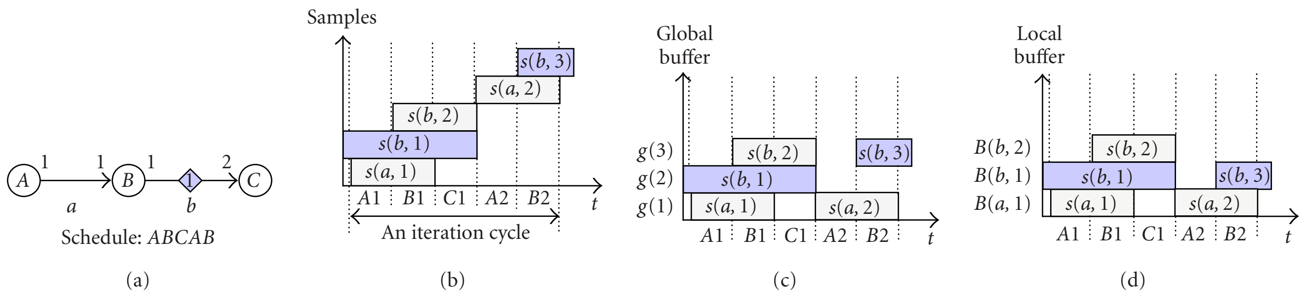


FIGURE 4: (a) An example of SDF graph with an initial delay between B and C illustrated by a diamond, (b) the sample lifetime chart, (c) a global buffer lifetime chart, and (d) a local buffer lifetime chart.

buffers, and the mapping of local buffers to global buffers. The detailed algorithm and code synthesis techniques will be explained in Section 4.

It is *NP*-hard to determine the optimal local buffer, global buffer sizes, and their mappings in general cases where there are feedback structures in the graph topology. The problem becomes harder if we consider buffer sharing among different size data samples. Therefore, we devise a heuristic that focuses on global buffer minimization first and applies an optimal algorithm next to find the minimum local pointer buffer sizes and to map the local pointer buffers to the minimum global buffers. The proposed heuristic results in less than 5% overhead than an optimal solution on average.

In Section 2, we define a new buffer sharing problem for nonprimitive data types, and survey the previous works briefly. The overview of the proposed technique is presented in Section 3. Section 4 explains how to minimize the size of local buffers and their mappings to the minimum global buffers assuming that all data samples have the same size. In Section 5, we extend the technique to the case where data samples have the different sizes. Graphs with initial samples are discussed in Section 6. Finally, we present some experimental results in Section 7, and make conclusions in Section 8.

2. PROBLEM STATEMENT AND PREVIOUS WORKS

In the proposed technique, global buffers store the live data samples of nonprimitive type while the local pointer buffers store the pointers for the global buffer entries. Since multiple data samples can share the buffer space as long as their lifetimes do not overlap, we should examine the lifetimes of data samples. We denote $s(a, k)$ as the kth stored sample on arc a and TNSE(a) as the total number of samples exchanged during an iteration cycle. Consider an example of Figure 4a with the associated schedule. TNSE(a) becomes 2 and two samples, $s(a, 1)$ and $s(a, 2)$, are produced and consumed on arc a. Arc b has an initial sample $s(b, 1)$ and two more samples, $s(b, 2)$ and $s(b, 3)$, during an iteration cycle.

The lifetimes of data samples are displayed in the *sample lifetime chart* as shown in Figure 4b, where the horizontal axis indicates the abstract notion of time: each invocation of a node is considered to be one unit of time. The vertical axis indicates the memory size and each rectangle denotes the lifetime interval of a data sample. Note that each sample lifetime defines a single time interval whose start time is the invocation time of the source block and the stop time is the completion time of the destination block. For example, the lifetime interval of sample $s(b, 2)$ is $[B1, C1]$. We take special care of initial samples. The lifetime of sample $s(b, 1)$ is carried forward from the last iteration cycle while that of sample $s(b, 3)$ is carried forward to the next iteration cycle. We denote the former-type interval as a *tail lifetime interval*, or shortly a tail interval, and the latter as a *head lifetime interval*, or a head interval. In fact, sample $s(b, 3)$ at the current iteration cycle becomes $s(b, 1)$ at the next iteration cycle. To distinguish iteration cycles, we use $s_k(b, 2)$ to indicate sample $s(b, 2)$ at the kth iteration. Then, in Figure 4, $s_1(b, 3)$ is equivalent to $s_2(b, 1)$.

And the sample lifetime that spans multiple iteration cycles is defined as a *multicycle lifetime*. Note that the sample lifetime chart is determined from the schedule.

From the sample lifetime chart, it is obvious that the minimum size of global buffer memory is the maximum of the total memory requirements of live data samples over time. We summarize this fact as the following lemma without proof.

Lemma 1. *The minimum size of global buffer memory is equal to the maximum total size of live data samples at any instance during an iteration cycle.*

We map the sample lifetimes to the global buffers: an example is shown in Figure 4c where $g(k)$ indicates the kth global buffer. In case all data types have the same size, an interval scheduling algorithm can successfully map the sample lifetimes to the minimum size of global buffer memory.

Sample lifetime is distinguished from local buffer lifetime since a local buffer may store multiple samples during an iteration cycle. Consider an example of Figure 4a where the local buffer sizes of arcs a and b are set to be 1 and 2, respectively. We denote $B(a, k)$ as the kth local buffer entry on arc a. Then, the *local buffer lifetime chart* becomes as drawn in Figure 4d. Buffer $B(a, 1)$ stores two samples, $s(a, 1)$ and $s(a, 2)$, to have multiple lifetime intervals during an iteration cycle. Now, we state the problem this paper aims to solve as follows.

Problem 1. Determine $\mathrm{LB}(g, s(g))$ and $\mathrm{GB}(g, s(g))$ in order to minimize the sum of them, where $\mathrm{LB}(g, s(g))$ is the sum of local buffer sizes on all arcs and $\mathrm{GB}(g, s(g))$ is the global buffer size with a given graph g and a given schedule $s(g)$.

Since the simpler problems are NP-hard, this problem is NP-hard, too. Consider a special case when all samples have the same type or the same size. For a given local buffer size, determining the minimum global buffer size is difficult if a local buffer may have multiple lifetime intervals, which is stated in the following theorem.

Theorem 1. *If the lifetime of a local buffer may have multiple lifetime intervals and all data types have the same size, the decision problem whether there exists a mapping from a given number of local buffers to a given number of global buffers is NP-hard.*

Proof. We will prove this theorem by showing that the graph coloring problem can be reduced to this mapping problem. Consider a graph $G(V, E)$ where V is a vertex set and E is an edge set. A simple example graph is shown in Figure 5a. We associate a new graph G' (Figure 5b) where a pair of nodes are created for each vertex of graph G and connected to the dummy source node S and the dummy sink node K of the graph G'. In other words, a vertex in graph G is mapped to a local buffer in graph G'. The next step is to map an arc of graph G to a schedule sequence in graph G'. For instance, an arc AB in graph G is mapped to a schedule segment $(A'B'A''B'')$ to enforce that two local buffers on arcs $A'A''$ and $B'B''$ may not be shared. As we traverse all arcs of graph G, we generate a valid schedule of graph G'. Traversing arcs AB and AC in graph G generates a schedule: $S(A'B'A''B'')(A'C'A''C'')K$. From this schedule, we find out that the buffer lifetime on arc $A'A''$ consists of two intervals. The constraint that two adjacent nodes in G may not have the same color is translated to the constraint that two local buffers may not be shared in G'. Therefore, the graph coloring problem for graph G is reduced to the mapping problem for graph G'. □

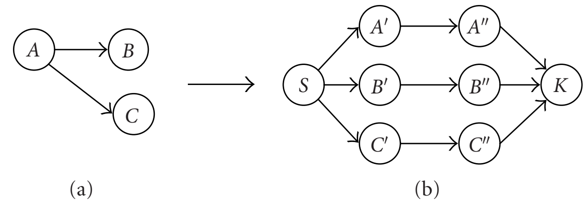


Figure 5: (a) An example instance of graph coloring problem, and (b) the mapped graph for the proof of Theorem 1.

The register allocation problem in traditional compilers is to share the memory space for the variables of nonoverlapped lifetimes [9]. If the variable sizes are not uniform, the allocation problem, known as the dynamic storage allocation problem [10, 11], is NP-complete. In our context, this problem is equivalent to minimize the global buffer memory ignoring the local buffer sizes and mapping problems.

De Greef et al. [12] presented a systematic procedure to share arrays for multimedia applications in a synthesis tool called ATOMIUM. They analyze lifetimes of array variables during a single iteration trace of a C program and do not consider the case where lifetimes span multiple iteration cycles. If the program is retimed, some variables can be live longer than a single iteration cycle. Another extension we make in the proposed approach is that we consider each array element separately for sharing decision when each array element is of nonprimitive type.

Recently, Murthy and Bhattacharyya [13] proposed a scheduling technique for SDF graphs to optimize the local memory size by buffer sharing. Since they assume only primitive type data, their sharing decision considers array variables as a whole. However, their research result is complementary to our work since the schedule reduces the number of live data samples at runtime, which reduces the global memory size in our framework. They compared their research work with Ritz et al.'s [14] whose schedule pattern does not allow nested loop structure. They showed that nested loop structure may significantly reduce the local memory size.

Even though memory sharing techniques have been researched extensively from compiler optimization to high level synthesis, no previous work has been performed, to the authors' knowledge, to solve the problem we are solving in this paper.

```
1: U is a set of sample lifetimes; P is an empty global buffer lifetime chart.
2: While (U is not empty){
3:      Take out a sample lifetime x with the earliest start time from U.
4:      Find out a global buffer whose lifetime ends earlier than the start time of x.
5:          Priority is given to the buffer that stores samples on the same arc if exists.
6:      If no such global buffer exists in P, create another global buffer.
7:      Map x to the selected global buffer
8: }
```

Figure 6: Interval scheduling algorithm.

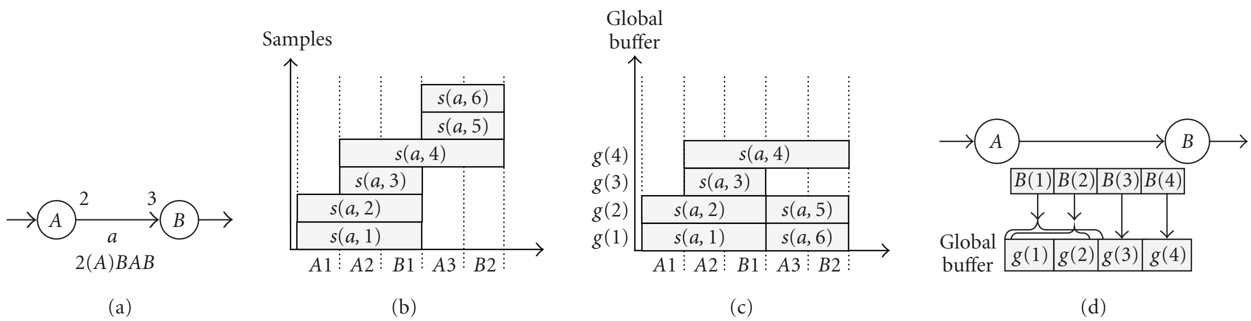


Figure 7: (a) An SDF subgraph with a given schedule, (b) the sample lifetime chart, (c) the global buffer lifetime chart, and (d) local buffer allocation and mapping.

3. PROPOSED TECHNIQUE

In this section, we sketch the proposed heuristic for the problem stated in the previous section. Since the size of nonprimitive data type is usually much larger than that of pointer type in multimedia applications of interest, reducing the global buffer size is more important than reducing the local pointer buffers. Therefore, our heuristic consists of two phases: the first phase is to map the sample lifetimes within an iteration cycle into the minimum number of global buffers ignoring local buffer sizes, and the second phase is to determine the minimum local buffer sizes and to map the local buffers to the given global buffers.

3.1. Global buffer minimization

Recall that a sample lifetime has a single interval within an iteration cycle. When all samples have the same data size, the interval scheduling algorithm is known to be an optimal algorithm [15] to find the minimum global buffer size. We summarize the interval scheduling algorithm in Figure 6.

Consider an example of Figure 4a whose global buffer lifetime chart is displayed in Figure 4c. After samples $s(a, 1)$, $s(b, 1)$, and $s(b, 2)$ are mapped into three global buffers, $s(a, 2)$ can be mapped to all three buffers. Among the candidate global buffers, we select one that already stores $s(a, 1)$ according to the policy of line 5 of Figure 6. The reason of this priority selection is to minimize the local buffer sizes, which will be discussed in the next section.

When the data samples have different sizes, this mapping problem becomes NP-hard since a special case can be reduced to 3-partition problem [10]. Therefore, we develop a heuristic, which will be discussed in Section 5.

3.2. Local buffer size determination

The global buffer minimization algorithm in the previous phase runs for one iteration cycle while the graph will be executed repeatedly. The next phase is to determine the minimum local buffer sizes that are necessary to store the pointers of data samples mapped to the global buffers. Initially we assign a separate local buffer to each live sample during an iteration cycle. Then, the local buffer size on each arc becomes the total number of live samples within an iteration cycle: each sample occupies a separate local buffer. In Figure 4a, for instance, two local buffers are allocated on arc a while three local buffers on arc b.

What is the optimal local buffer size? The answer depends on when we set the pointer values, or when we *bind* the local buffers to the global buffers. If binding is performed statically at compile time, we call it *static* binding. If binding can be changed at runtime, it is called *dynamic* binding. In general, the dynamic binding can reduce the local buffer size significantly with small runtime overhead of global buffer management.

3.2.1 Dynamic binding strategy

Since we can change the pointer values at runtime in dynamic binding strategy, the local buffer size of an arc can be as small as the maximum number of live samples at any time instance during an iteration cycle. Consider another example of Figure 7a with a given scheduling result and a global buffer lifetime chart as shown in Figure 7c. Since the maximum number of live samples is four, we need at least four local buffers on arc a. Suppose we have the minimum number of local buffers on arc a. Local buffer $B(a, 1)$ stores two samples, $s(a,1)$ and $s(a,5)$, which are unfortunately mapped to different global buffers. It means that the pointer value of local buffer $B(a,1)$ should be set to $g(1)$ at the first invocation of node A but to $g(2)$ at the third invocation, dynamically. We repeat this pointer assignment at every iteration cycle at runtime.

If there are initial samples on an arc, care should be taken to compute the repetition period of pointer assignment. Arc b of Figure 4a has an initial sample and needs only two local buffers since there are at most two live samples at the same time. Unlike the previous example of Figure 7, the global buffer lifetime chart may not repeat itself at the next iteration cycle. The lifetime patterns of local buffers $B(b,1)$ and $B(b,2)$ are interchanged at the next iteration cycle as shown in Figure 8. In other words, the repetition periods of pointer assignment for arcs with initial samples may span multiple iteration cycles. Section 4 is devoted to computing the repetition period of pointer assignment for the arcs with initial samples.

Suppose an arc a has M local buffers. Since the local buffers are accessed sequentially, each local buffer entry has at most $\lceil \mathrm{TNSE}(a)/M \rceil$ samples and the pointer to sample $s(a,k)$ is stored in $B(a, k \bmod M)$. After the first phase is completed, we examine the mapping results of the allocated sample in a local buffer to the global buffers at the code generation stage. If the mapping result of the current sample is changed from the previous one, a code segment is inserted automatically to alter the pointer value at the current schedule instance. Note that it incurs both memory overhead of code insertion and time overhead of runtime mapping.

3.2.2 Static binding strategy

If we use static binding, we may not change the pointer values of local buffers at runtime. It means that all allocated samples to a local buffer should be mapped to the same global buffer. For example of Figure 7, we need six local buffers for static binding: two more buffers than the dynamic binding case since $s(a,1)$ and $s(a,5)$ are not mapped to the same global buffer. On the other hand, arc a of Figure 4 needs only one local buffer for static binding since two allocated samples are mapped to the same global buffer. How many buffers do we need for arc b of Figure 4 for static binding?

To answer this question, we extend the global buffer lifetime chart over multiple iteration cycles until the sample lifetime patterns on the arc become periodic. We need to extend the lifetime chart over two iteration cycles as displayed in Figure 8. Note that the head interval of $s_2(b,3)$ is connected to the tail interval of $s_3(b,1)$ in the next repetition period. Therefore, four live samples are involved in the repetition period that consists of two iteration cycles. The problem is to find the minimum local buffer size M such that all allocated samples on each local buffer are mapped to the same global buffer. The minimum number is four in this example since $s_3(b,1)$ can be placed at the same local buffer as $s_1(b,1)$.

How many iteration cycles should be extended is an equivalent problem to computing the repetition period of pointer assignment for dynamic binding case. We refer to the next section for detailed discussion.

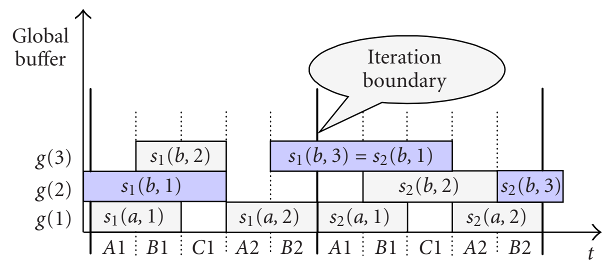


FIGURE 8: The global buffer lifetime chart spanning two iteration cycles for the example of Figure 4.

4. REPETITION PERIOD OF SAMPLE LIFETIME PATTERNS

Initial samples may make the repetition period of the sample lifetime chart longer than a single iteration cycle since their lifetimes may span to multiple cycles. In this section, we show how to compute the repetition period of sample lifetime patterns to determine the periodic pointer assignment for dynamic binding or to determine the minimum size of local buffers for static binding. For simplicity, we assume that all samples have the same size in this section. This assumption will be released in Section 5.

First, we compute the iteration length of a sample lifetime. Suppose d initial samples stay alive on an arc and N samples are newly produced for each iteration cycle. Then, N samples on the arc are consumed from the destination node. If d is greater than N, the newly produced samples all live longer than an iteration cycle. Otherwise, $N - d$ newly created samples are consumed during the same iteration cycle while d samples live longer. We summarize this fact in the following lemma.

Lemma 2. *If there are $d(a)$ initial samples on an arc a, the lifetime interval of $(d(a) \bmod \mathrm{TNSE}(a))$ newly created samples on the arc spans $\lceil d(a)/\mathrm{TNSE}(a) \rceil + 1$ iteration cycles and that of $(\mathrm{TNSE}(a) - (d(a) \bmod \mathrm{TNSE}(a)))$ samples spans $\lceil d(a)/\mathrm{TNSE}(a) \rceil$ iteration cycles.*

Let p be the number of iteration cycles in which a sample lifetime interval lies. Figure 9 illustrates two patterns that a sample lifetime interval can have in a global lifetime chart. A sample starts its lifetime at the first iteration cycle with a head interval and ends its lifetime at the pth iteration with a tail interval. Note that the tail interval at the pth iteration also appears at the first iteration cycle. The first pattern, as shown in Figure 9a, occurs when the tail interval is mapped to the same global buffer as the head interval. The interval mapping pattern repeats every $p - 1$ iteration cycles in this case.

The second pattern appears when the tail interval is mapped to a different global buffer. To compute the repetition period, we have to examine when a new head interval can be placed at the same global buffer. Figure 9b shows a simple case that a new head interval can be placed at the next iteration cycle. Then, the repetition period of the sample

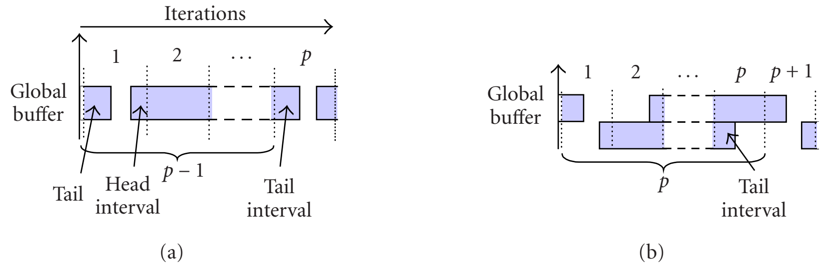


Figure 9: Illustration of a sample lifetime interval: (a) when the tail interval is mapped to the same global buffer as the head interval, and (b) when the tail interval is mapped to a different global buffer and there is no chained multicycle sample lifetime interval.

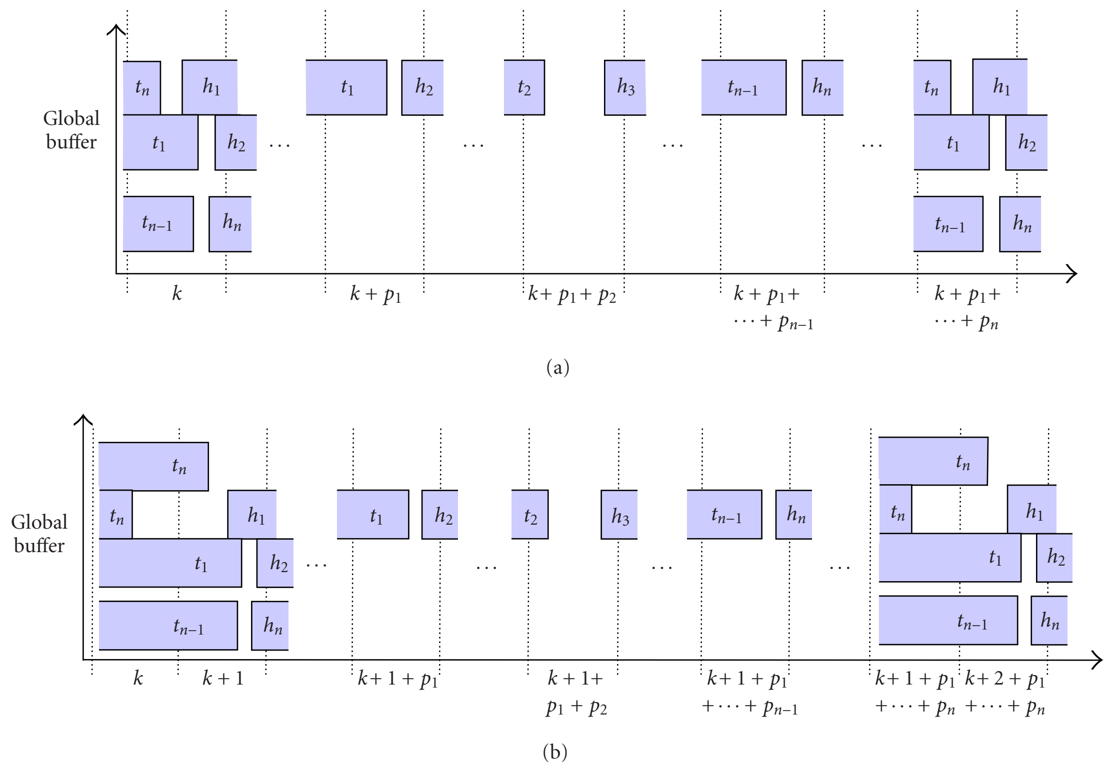
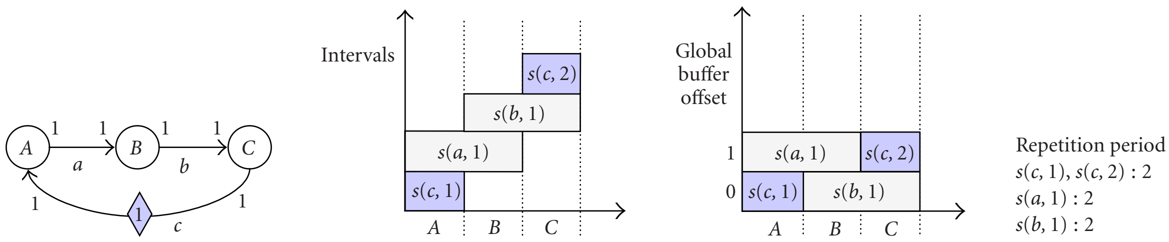


Figure 10: Sample lifetime patterns when multicycle lifetimes are chained so that tail interval t_i is chained to the lifetime of sample $j+1$. (a) Case 1: t_n is chained back to the lifetime of sample 1. The repetition period of sample lifetime patterns becomes $\sum_{i=1}^{n} p_i$. (b) Case 2: t_n is chained to none. The repetition period becomes $\sum_{i=1}^{n} p_i + 1$. Here, we assume that the lifetime of sample k spans $p_k + 1$ iteration cycles.

lifetime pattern becomes p. More general case occurs when another multicycle sample lifetime on a different arc is chained after the tail interval. A multicycle lifetime is called *chained* to a tail interval when its head interval is placed at the same global buffer. The next theorem concerns this general case.

Theorem 2. *Let t_i be the tail interval and h_i the head interval of sample i, respectively. Assume the lifetime of sample i spans $p_i + 1$ and t_i is chained to the lifetime of sample $i+1$ for $i = 1$ to $n-1$. The interval mapping pattern repeats every $\sum_{i=1}^{n} p_i$ iteration cycles if interval t_n is chained back to the lifetime of sample 1. Otherwise it repeats every $\sum_{i=1}^{n} p_i + 1$ iteration cycles.*

Proof. Figure 10 illustrates two patterns where chained multicycle lifetime intervals are placed. The horizontal axis indicates the iteration cycles. The lifetime interval of sample 1 starts at k with head interval h_1 and finishes at $k + p_1$ with tail interval t_1. Since the lifetime of sample 2 is chained, its head interval h_2 is placed at the same global buffer as t_1. The lifetime of sample 2 ends $k + p_1 + p_2$. If we repeat this process, we can find that the lifetime of sample n ends at $k + \sum_{i=1}^{n} p_i$. Now, we consider two cases separately. Case 1: when interval t_n is chained back to the lifetime of sample 1, the repetition period becomes $\sum_{i=1}^{n} p_i$ as illustrated in Figure 10a. Case 2: when interval t_n is chained to no more lifetime, we should prove that sample 1 is mapped to the same global buffer at



(a) (b) (c) (d)

```
struct frame g[2];
main()
{
        struct G *a, *b, *c[2] = {g, g + 1};
        int in_A = 0, out_C = 1;
        for(int i = 0; i < max_iteration; i++) {
                {       a = c[(i + 1)%2];
                        // A's codes. Use c[in_A] and a.
                        in_A = (in_A + 1)%2;
                }
                {       b = c[i%2];
                        // B's codes. Use a and b.
                }
                {       // C's codes. Use b and c[out_C].
                        out_C = (out_C + 1)%2;
                }
        }
}
```

(e)

```
struct frame g[2];
main()
{
    struct G *a[2] = {g + 1, g}, *b[2] = {g, g + 1}, *c[2] = {g, g + 1};
    int in_A = 0, out_A = 0, in_B = 0, out_B = 0, in_C = 0, out_C = 1;
    for(int i = 0; i < max_iteration; i++) {
            {   // A's codes. Use c[in_A] and a[out_A].
                in_A = (in_A + 1)%2; out_A = (out_A + 1)%2;
            }
            {   // B's codes. Use a[in_B] and b[out_B].
                in_B = (in_B + 1)%2; out_B = (out_B + 1)%2;
            }
            {   // C's codes. Use b[in_C] and c[out_C].
                in_C = (in_C + 1)%2; out_C = (out_C + 1)%2;
            }
    }
}
```

(f)

Figure 11: (a) A graph which is equivalent to Figure 3a, (b) lifetime intervals of samples for an iteration cycle, (c) an optimal global buffer lifetime chart, (d) repetition periods of sample lifetime patterns, (e) generated code with dynamic binding, and (f) generated code with static binding.

the next iteration cycle as shown in Figure 10b. Then, the period becomes $\sum_{i=1}^{n} p_i + 1$. Since the sample lifetime patterns over iteration cycles are permutations of each other, sample 1 should be mapped to among n global buffers assigned to samples 1 through n during previous iterations. As illustrated in Figure 10b, other global buffers are occupied by other samples at $k + \sum_{i=1}^{n} p_i + 1$ except the global buffer mapped to t_n. Therefore, sample 1 is mapped to the same global buffer at the next iteration cycle. □

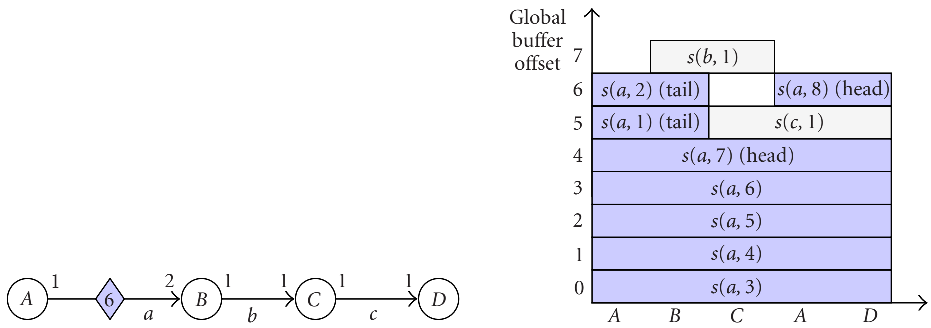
We apply the above theorem to the case of Figure 4b where head interval $s(b, 3)$ and tail interval $s(b, 1)$ are mapped to the different global buffers. And the sample lifetime spans two iteration cycles. Therefore, the repetition period becomes 2 and Figure 8 confirms it.

Another example graph is shown in Figure 11a, which is identical to the simplified H.263 encoder example of Figure 3. There is a delay symbol on arc CA with a number inside which indicates that there is an initial sample $s(c, 1)$. Assume that the execution order is ABC. During an iteration cycle, sample $s(c, 1)$ is consumed by A and a new sample $s(c, 2)$ is produced by C as shown in Figure 11b. If we expand the lifetime chart over two iteration cycles, we can notice that head interval $s_1(c, 2)$ is extended to tail interval $s_2(c, 1)$ at the second iteration cycle. By interval scheduling, an optimal mapping is found like Figure 11c. By Theorem 2, the mapping patterns of $s(c, 1)$ and $s(c, 2)$ repeat every other iteration cycles since head interval $s(c, 2)$ is not mapped to the same global buffer as tail interval $s(c, 1)$.

Initial samples also affect the lifetime patterns of samples on the other arcs if they are mapped to the same global buffers as the initial samples. In Figure 11c, sample $s(b, 1)$ are mapped to the same global buffer with $s(c, 1)$ while $s(a, 1)$ with $s(c, 2)$. As a result, their lifetime patterns also repeat themselves every other iteration cycles. The summary of repetition periods is displayed in Figure 11d.

Recall that the repetition periods determine the period of pointer update in the generated code with dynamic binding strategy, and the size of local buffers in the generated code with static binding strategy. Figures 11e and 11f show the code segments that highlight the difference.

The dynamic binding scheme allocates a local pointer buffer onto arc AB since the number of samples accumulated on arc AB is no greater than one. Similarly, a local buffer is allocated on arc BC. Figure 11e shows a code with dynamic



(a) (b)

Repetition period
$s(a,1), s(a,3), s(a,5), s(a,7) : 4$
$s(a,2), s(a,4), s(a,6), s(a,8) : 3$
$s(b,1) : 1$
$s(c,1) : 4$

(c)

```
struct Gg[8];
main()
{
        struct G *a0[4] = {g + 4, g, g + 2, g + 5}, *a1[3] = {g + 6, g + 1, g + 3},
              *b[1] = {g + 7}, *c[1] = {0};

        for(int i = 0; i < max_iteration; i++) {
                {       struct G *output = a0[(i + 3)%4];
                        // A's codes
                }{      struct G *input[2];
                        input[0] = a0[i%4]; input[1] = a1[i%3];
                        // B's codes
                }{      c[0] = a0[i%4];
                        // C's codes
                }{      struct G *output = a1[(i + 2)%3];
                        // A's codes
                }{
                        // D's codes
                }
        }
}
```

(d)

Figure 12: (a) An SDF graph with large initial samples, (b) an optimal global buffer lifetime chart, (c) repetition periods of sample lifetime patterns, and (d) generated code with dynamic binding after dividing local buffers on arc AB into two local buffer arrays.

binding. When the size of a local buffer is the same as the number of newly produced samples within an iteration, no buffer index is needed for the buffer in the generated inlined code. The mapped offset of sample $s(a,1)$ repeats every other cycles as that of $s(c,2)$ does. The mapped offset of $s(b,1)$ follows that of $s(c,1)$. For arc CA, the minimum size of local buffers is one since there is at most a live sample on the arc. But we notice that if we have a local buffer on the arc, we need to update the pointer value of each local buffer at every access since the repetition period is two. Therefore, we allocate two local buffers on arc CA and fix the buffer pointers. Instead, we update the local buffer indices, in_A for block A and out_C for block C. The decision of the binding scheme is automatically taken care of by the algorithm.

The static binding requires two local pointer buffers for arc AB and BC, respectively, since the mapping patterns of samples on AB repeat every other iteration cycles. The local buffer size for arc CA is two and has the same binding as Figure 11e. Figure 11f represents a generated code with static binding, which additionally requires buffer indices for local buffers on arc AB and BC [16]. Hence, we add additional code of updating buffer indices before and after the associated block's execution. We should consider this overhead to compare the static binding with the dynamic binding strategies. In this example, using the dynamic binding strategy is more advantageous.

We illustrate an example graph which has large initial delays and thus has long repetition period of sample lifetime patterns in Figure 12. The schedule is assumed to be given as $ABCAD$. Interestingly enough, samples on the same arc AB have different repetition periods. The mapping patterns repeat every four iteration cycles for samples $s(a,1)$, $s(a,3)$, $s(a,5)$, and $s(a,7)$ since each sample spans four iteration cycles and tail interval $s(a,1)$ is not mapped to the same global

```
1: Procedure LOES(U is a set of sample lifetimes) {
2:      P ← {}
3:      While(U is not empty) {
4:           /* compute feasible offsets of every interval in U with P */
5:           compute lowest offset(U, P);
6:           /* 1st step: choose intervals with the smallest feasible offset from U */
7:           C ←find intervals with lowest offset(U);
8:           /* 2nd step(tie breaking) : interval scheduling */
9:           select interval x with the earliest arrival time from C;
10:          remove x from U.
11:          P ← P ⋃ U{x}.
12:     }
13:}
```

Figure 13: Pseudocode of LOES algorithm.

buffer as head interval $s(a, 7)$. On the other hand, samples $s(a, 2)$, $s(a, 4)$, $s(a, 6)$, and $s(a, 8)$ repeat their lifetime patterns every three iteration cycles since tail interval $s(a, 2)$ and head interval $s(a, 8)$ are mapped to the same global buffer. The static binding method allocates twelve local buffers to arc AB since the overall repetition period of local buffers on arc AB becomes twelve that is equal to the least common multiple of 4 and 3 ($= \mathrm{LCM}(4, 3)$). The dynamic binding method, however, allots two local buffer arrays that have four and three buffers, respectively, to arc AB. Hence the dynamic binding method can reduce five local pointer buffers than the static binding. A code template with inlined coding style is displayed in Figure 12d. The local buffer pointer for arc CD follows that of sample $s(a, 1)$.

Up to now, we assume that all samples have the same size. The next two sections will discuss the extension of the proposed scheme to a more general case, where samples of different sizes share the same global buffer space.

5. BUFFER SHARING FOR DIFFERENT SIZE SAMPLES WITHOUT DELAYS

We are given sample lifetime intervals which are determined from the scheduled execution order of blocks. The optimal assignment problem of local buffer pointers to the global buffers is nothing but to pack the sample lifetime intervals into a single box of global buffer space. Since the horizontal position of each interval is fixed, we have to determine the vertical position, which is called the "vertical offset" or simply "offset." The bottom of the box, or the bottom of the global buffer space has offset 0. The objective function is to minimize the global buffer space. Recall that if all samples have the same size, interval scheduling algorithm gives the optimal result. Unfortunately, however, the optimal assignment problem with intervals of different sizes is known to be NP-hard. The lower bound is evident from the sample lifetime chart; it is the maximum of the total sample sizes live at any time instance during an iteration. We propose a simple but efficient heuristic algorithm. If the graph has no delays (initial samples), we can repeat the assignment every iteration cycle. Graphs with initial samples will be discussed in the next section.

The proposed heuristic is called LOES (lowest offset and earliest start time first). As the name implies, it assigns intervals in the increasing order of offsets, and in the increasing order of start times as a tie breaker. At the first step, the algorithm chooses an interval that can be assigned to the smallest offset, among unmapped intervals. If more than one interval is selected, then an interval is chosen which starts no later than others. The earliest start time first policy allows the placement algorithm to produce an optimal result when all samples have the same size since the algorithm is equivalent to the interval scheduling algorithm.

The detailed algorithm is depicted in Figure 13. In this pseudocode, U indicates a set of unplaced sample lifetime intervals and P a set of placed intervals. At line 5, we compute the feasible offset of each interval in U. Set C contains intervals whose feasible offsets are lowest among unplaced intervals at line 7. We select the interval with the earliest start time in C at line 9 and place it at its feasible offset to remove it from U and add it to P. This process repeats until every interval in U is placed.

Since the LOES algorithm can find intervals with lowest offset in $O(n)$ time and choose the earliest interval among them in $O(n)$, where n is the number of lifetime intervals, it has $O(n)$ time complexity to assign an interval. Therefore the time complexity of the algorithm is $O(n^2)$ for n intervals.

Figure 14 shows an example graph where the circled number on each arc indicates the sample size. Figure 14b presents a schedule result and the resultant sample lifetime intervals. Figure 15 shows the procedure of the LOES algorithm at work. At first, we select d with the earliest start time first among the intervals that can be mapped to lowest offset 0. Next, f is selected and placed since it is the only interval that can be placed at offset 0. In this example, the LOES algorithm produces an optimal assignment result. With randomly generated graphs, it gives near-optimal results most of the time as shown later.

De Greef et al. proposed a similar heuristic that considers the offset first and sample size next in [12]. Even though

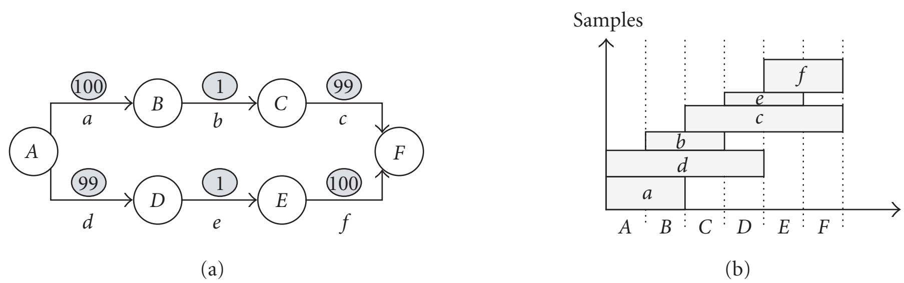


Figure 14: (a) An input graph with samples of different sizes and (b) a schedule (= $ABCDEF$) and the resultant sample lifetime chart.

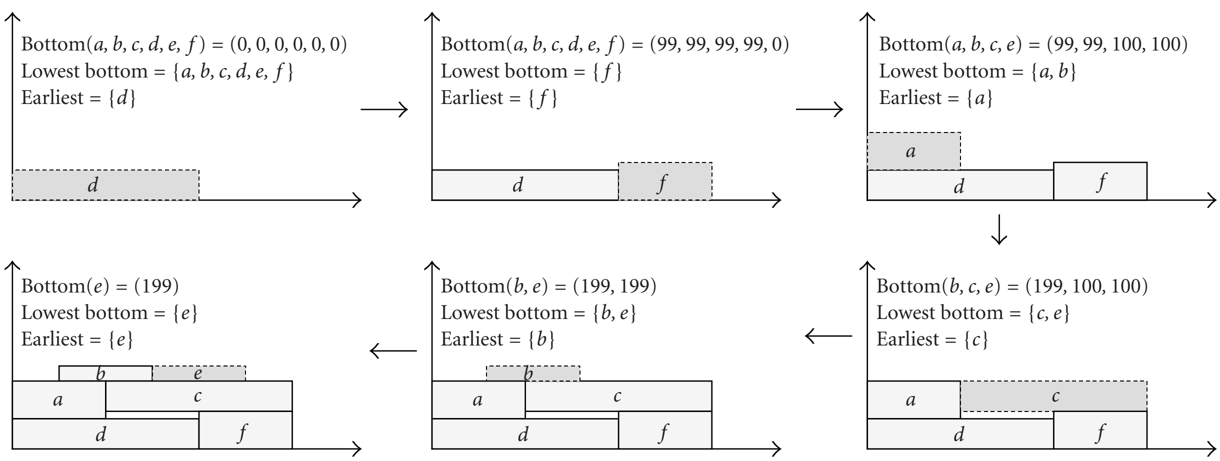


Figure 15: The proposed placement algorithm at work.

their heuristic gives similar performance with randomly generated graphs, it does not guarantee to produce optimal results when all samples have the same size.

6. BUFFER SHARING FOR DIFFERENT SAMPLE SIZES WITH INITIAL SAMPLE DELAYS

In this section, we discuss the most general case where a graph has initial samples and samples have different sizes. The LOES algorithm is not directly applicable to this case. Figure 16 illustrates the difficulty with a simple example. Figure 16a shows a mapping result after the LOES algorithm is applied to the first iteration period. We assume that "h" and "t" indicate the head interval and the tail interval of the same sample lifetime, respectively. At the second iteration, interval h should be placed as shown in Figure 16b since it is extended from the first cycle. The head interval h prohibits interval x from lying on contiguous memory space at the second iteration. Such splitting is not allowed in the generated code since the code regards each sample as a unit of assignment. To overcome this difficulty, we enforce that multicycle intervals do not share the global buffer space with other intervals with different sample size.

Figure 17 displays the main procedure, ASSIGN_MAIN, for the proposed technique. We first classify intervals into several groups (lines 2–13 in Figure 17); a new group is formed with all intervals of the same size if there is at least one multicycle interval, and is denoted as $D(x)$ where x is the sample size. If there is no multicycle interval, all remaining intervals form the last group R. Consider an example of Figure 18a where sample sizes are indicated as circled numbers on the arcs. The sample lifetimes are displayed in Figure 18b. We make three groups of intervals for this graph. Group $D(100)$ includes all sample intervals associated with arcs b, c, and d while group $D(200)$ includes all intervals associated with arcs a and e. Initially group R is empty in this example.

The next step is to apply the LOES algorithm for each group $D(x)$ since $D(x)$ contains samples of the same size only (lines 14–17). We slightly modify the LOES algorithm so that the algorithm finishes the mapping as soon as all multicycle intervals are mapped: compare line 24 of Figure 17 and line 12 of Figure 13. The remaining unmapped intervals are moved to group R. In Figure 18c, the modified LOES algorithm places intervals in $D(100)$ in the order of $[s(c,1), s(b,2), s(d,2), s(d,1), s(b,1), s(c,3)]$. After placing $s(c,3)$, it completes and moves a remaining interval $s(c,2)$

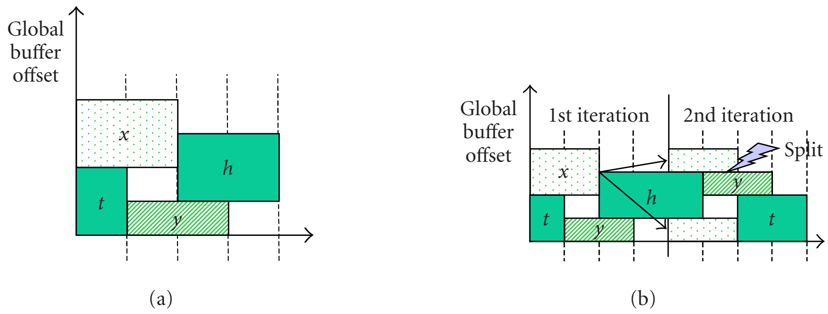


FIGURE 16: (a) A global buffer lifetime chart with different size samples where t is a tail interval and h is a head interval. (b) Interval x should be split at the second iteration, which is not allowed in the generated code.

```
1: Procedure ASSIGN_MAIN(U is a set of sample lifetimes) {
2:      R ← { }
3:      for each x in U{/* classify intervals and sort them */
4:           if (x is delayed interval)
5:                D(size(x)) ← D(size(x)) ⋃{x}
6:           else      R ← R⋃{x}
7:      }
8:      for each x in R{/* add ordinary intervals into D(x) */
9:           if(there is a delayed interval whose size is equal to that of x) {
10:               D(size(x)) ← D(size(x)) ⋃{x}
11:               R ← R − {x}
12:          }
13:     }
14:     for each D(size){  /* place intervals in D(x) */
15:          call M_LOES (D(size)).
16:          R ← R⋃D(size)
17:     }
18:     /* place ordinary intervals in R */
19:     call LOES(R).
20:     }
21:}
22: Procedure M_LOES(U is a set of sample lifetimes) {      /* slightly modified LOES */
23:     P ← { }
24:     While(U has delayed intervals) {
             ...
```

FIGURE 17: Pseudocode of the proposed algorithm for a graph with delays.

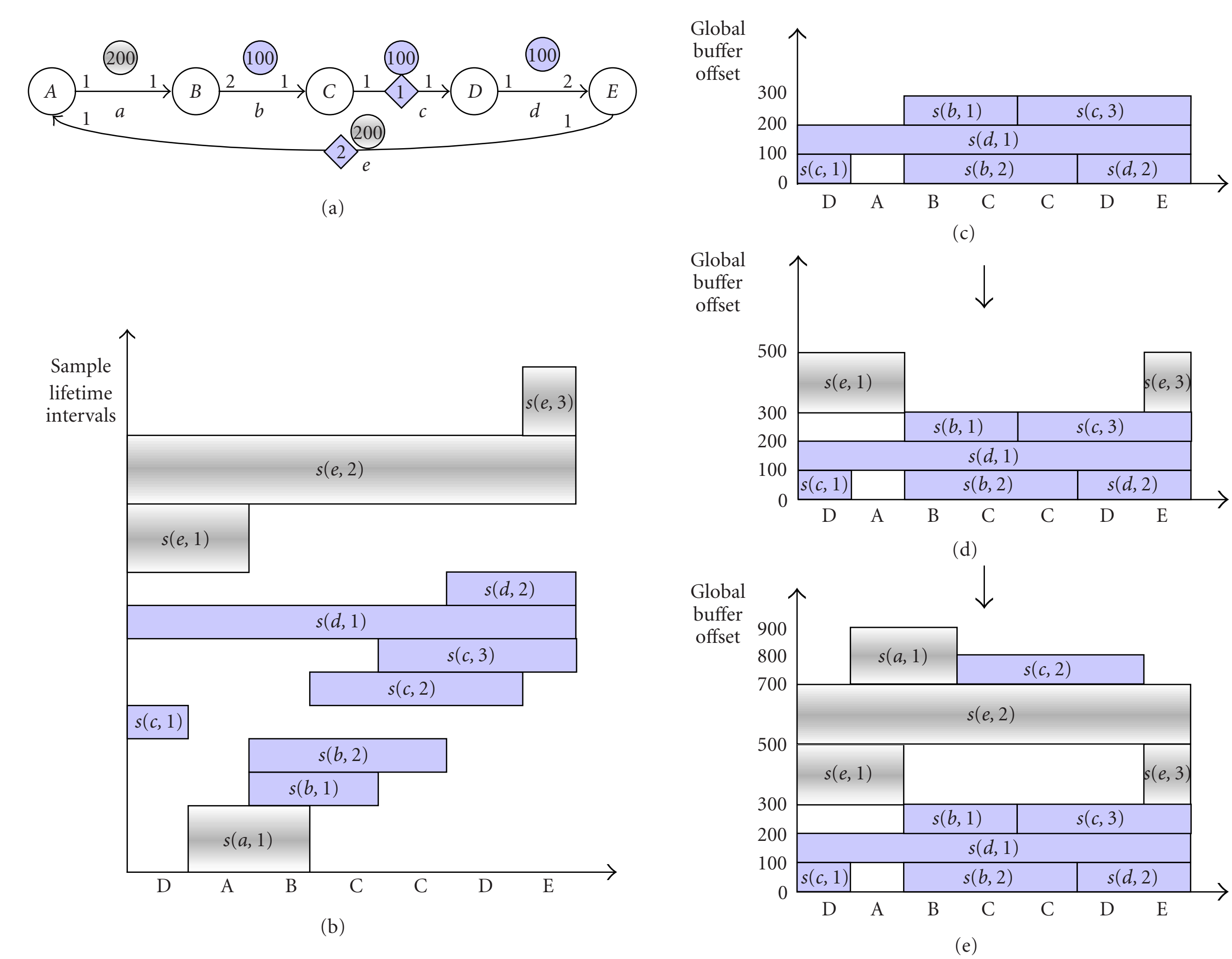
into R. Similarly, ordinary interval $s(e, 2)$ and $s(a, 1)$ are moved into R after $s(e, 3)$ is placed for group $D(200)$. At last we apply the original LOES algorithm to group R (line 19 of Figure 17) as shown in Figure 18e. The algorithm locates intervals $s(c, 2)$, $s(e, 2)$, and $s(a, 1)$ in R as shown in Figure 18e.

After all intervals are mapped to the global buffers, we move to the next stage of determining the local buffer sizes, which is already discussed in Section 4. Repetition periods for $s(c, 1)$ and $s(c, 3)$ become two since tail interval $s(c, 1)$ spans two iteration cycles and is not mapped to the same global buffer as $s(c, 3)$ is. Repetition periods of $s(e, 1)$, $s(e, 2)$, and $s(e, 3)$ become all two. A generated code with static binding is displayed in Figure 18f.

7. EXPERIMENT

In this section, we demonstrate the performance of the proposed technique with three real-life examples and randomly generated graphs.

First, we experimented three real-life applications: a JPEG encoder, an MP3 decoder, and an H.263 encoder. We have implemented the proposed technique in a design environment called PeaCE Ptolemy extension as Codesign Environment (http://peace.snu.ac.kr/research/peace) in which dataflow program graphs for a JPEG encoder and an H.263 encoder are displayed in Figures 19 and 20, respectively. From the automatically synthesized codes, we compute the buffer memory requirements and summarize the




```
char g[900];
main()
{
        struct G100 *b[4] = {g + 200, g, g, g + 200},
                    * c[4] = {g, g + 700, g + 200, g + 700},
                    * d[4] = {g + 100, g, g + 100, g + 200};
        struct G200 *a[1] = {g + 700}, *e[2] = {g + 300, g + 500};

        for(int i = 0; i < max_iteration; i + +) {
                    {       // D's codes
                    }{      // A's codes
                    }{      // B's codes
                    }{      // C's codes
                    }{      // C's codes
                    }{      // D's codes
                    }{      // E's codes
                    }
        }
}
```

(f)

Figure 18: (a) A graph with samples of different sizes and delays, (b) a sample lifetime chart, (c) LOES placement of samples whose size is 100, (d) LOES placement of samples whose size is 200, (e) LOES placement of the remained samples without delay, and (f) a generated code with static binding.

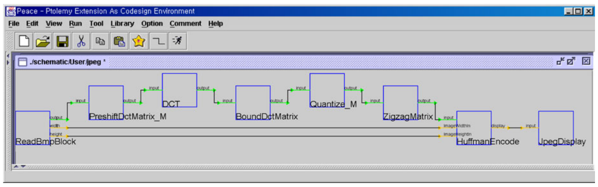


Figure 19: JPEG encoder example that represents a graph of same size samples without delays.

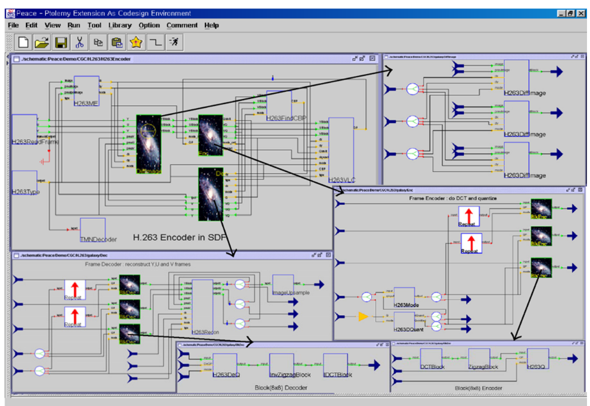


Figure 20: H.263 encoder example that represents a graph of different size samples with delays.

performance comparison results in Table 1. Since the function body of each dataflow node is equivalent in all experiments, only buffer memory requirements are the main item of comparison and the execution times are all similar except for the buffer copy overhead to be discussed below.

A JPEG encoder example represents the first and the simplest class of program graphs in which all nonprimitive data samples have the same size and no initial delay exists. Since the local buffer size of each arc is one in this example, we do not have to separate the local pointer buffer and the global data buffer, which is taken into account in the implementation of the proposed technique. We can reduce the memory requirements to one third as the third and the fourth rows of Table 1 illustrate. The last row indicates the lower bound of global buffer requirements that a given execution sequence of nodes needs. The lower bound is nothing but the maximum total size of data samples live at any instance of time during an execution period. No better result is possible since it is optimal.

An MP3 decoder example is composed of three kinds of different size samples. It represents the second class of graphs that have different size samples but no initial delay sample. In this example, we also do not have to separate local and global buffers because the local buffer size of each arc is one. The proposed algorithm that shares the buffer space between different size samples reduces the memory requirement by 52% compared with any sharing algorithm that shares the buffer space among only equal size samples. The fourth and the fifth rows show the performance difference. The proposed algorithm also achieves the lower bound in this example.

As for an H.263 encoder example, we make two versions: the simplified version as discussed in the first section and the full version. The simplified version is an example of the third class of graphs in which all nonprimitive data samples have the equal size and initial delay samples exist. As discussed earlier, separation of local buffers from global buffers allows us to reduce the buffer space to optimum as the sixth row reveals. The full version of an H.263 encoder example represents the fourth and the most general class of graphs that consist of different size samples and initial samples. The H.263 encoder example include four different size sample sizes and eight initial delay samples on eight different arcs. The proposed technique can reduce the memory requirement by 40% compared with the unshared version. On the other hand, a sharing technique reduces the buffer size only 23% if neither buffer separation nor sharing between different samples is considered. In this sample, we cannot achieve the lower bound but 256 bytes larger buffer space. Note that the lower bound is usually not achievable if different size samples and initial samples are involved in the same graph.

The SDF model has a limitation that it regards a sample of nonprimitive type as a unit of data delivery. In an H.263 encoder example, the SDF model needs an additional block that explicitly divides a frame into 99 macroblocks, paying nonnegligible data copy overhead and extra frame-size buffer space. In a manually written reference code, such data copy is replaced with pointer operation. Table 2 reveals this observation: that even the lower bound of memory requirements of the synthesized code from the SDF model is greater than that of the reference code. Therefore, we apply the proposed technique to an extended SDF model, called cyclo-static dataflow (CSDF) [17]. With the CSDF model, we could remove such data copy overhead. And the proposed buffer sharing technique further reduce the memory requirement by 17% more than the reference code.

In the experiments, we choose the better binding strategy, static or dynamic, for each data samples, considering the buffer memory and the code overhead of index updates. In the H.263 encoder example, static binding is preferred for places where the repetition periods of sample lifetimes span more than one iteration cycle. In this example, the pointer referencing through local buffers incurs runtime overhead, which is about 0.16% compared with the total execution time in the H.263 encoder.

The second set of experiments as shown in Table 3 have been performed to evaluate the proposed LOES heuristic. We compare it with an integer linear programming (ILP) solver, CPLEX (http://www.cplex.com). We randomly generate the sample lifetimes within a first iteration interval, varying the start/end times, sizes, and initial delays. When the number of sample intervals exceeds 20, the ILP solver takes prohibitively long execution times. With small size problems below 20 sample intervals, the overhead of LOES heuristic is less than 1% on the average for 150 experiments.

Table 1: Comparison of buffer memory requirements for three real-life examples.

Example	JPEG	MP3	Simplified H.263 encoder	H.263 encoder
Class	Same size No delay	Different size No delay	Same size With delay	Different size With delay
# of samples	6	336	3	1804
Without buffer sharing	1536 B	36 KB	111 KB	659 KB
Sharing buffers of same size only	512 B	23 KB	111 KB	510 KB
Buffer sharing without buffer separation	512 B	11 KB	111 KB	510 KB
Buffer sharing with buffer separation	—	—	74 KB	396 KB
Lower bound of global buffer size	512 B	11 KB	74 KB	396 KB

Table 2: Comparison of synthesize codes with reference code for the H.263 encoder.

Example	Reference code	Buffer sharing without buffer separation in CSDF	Buffer sharing with buffer separation in CSDF
H.263 Encoder	350 KB	291 KB	290 KB

Table 3: Performance comparison of LOES algorithm with integer linear programming (optimal) for randomly generated graphs (unit: %).

# of intervals	5	7	9	11	13	15	20	Avg.	max
(LOES-ILP)/ILP	0.0	0.0	0.1	0.1	0.5	0.3	0.3	0.2	14

8. CONCLUSIONS

We have proposed a buffer sharing technique for data samples of nonprimitive type to minimize the buffer memory requirements from graphical dataflow programs based on the SDF model or its extension assuming that the execution order of nodes is already determined at compile time. In order to share more buffers, the proposed technique separates global memory buffers from local pointer buffers: the global buffers store live samples and the local buffers store the pointers to the global buffer entries. The technique minimizes the buffer memory by sharing global buffers for data samples of different size. No previous work is known to us to solve this sharing problem especially for the graphs with initial samples. It also involves three subproblems of mapping local pointer buffers onto the global buffer space, determining the local buffer sizes, and finding the repeating mapping patterns.

We first obtain the minimum size of global buffer spaces assuming that local pointer buffers take negligible amount of buffer space compared with the global buffer space. A LOES algorithm has been developed for buffer sharing between samples of different sizes. The next step was to bind the local pointer buffers to the given global buffers for the graph. We present both dynamic binding and static binding methods and compare them in terms of memory requirements and code overheads. The proposed technique, including automatic code generation and memory optimization, has been implemented in a block diagram design environment called PeaCE. No manual intervention is necessary for the proposed code generation technique in PeaCE.

The experimental results show that the proposed algorithm is useful, especially for the graphs with initial delays. The proposed algorithm that separates local buffers and global buffers reduce more memory by 33% in the simplified H.263 encoder and 22% in the H.263 encoder than the sharing algorithm that does not separate local buffers and global buffers. Through extensive buffer sharing optimizations, automatic software synthesis from a dataflow program graph achieves the comparable code quality with the manually optimized code in terms of memory requirement.

In this paper, we assume that the execution order of blocks is given from the compile-time scheduling. In the future, we will develop an efficient scheduling algorithm which minimizes the memory requirement based on the proposed algorithm.

ACKNOWLEDGMENTS

This research is supported by National Research Laboratory Program (no. M1-0104-00-0015) and Brain Korea 21 Program. The RIACT at Seoul National University provides research facilities for this study.

REFERENCES

[1] *COSSAP User's Manual, Synopsys*, Mountain View, Calif, USA.

[2] R. Lauwereins, M. Engels, J. A. Peperstraete, E. Steegmans, and J. Van Ginderdeuren, "GRAPE: a CASE tool for digital signal parallel processing," *IEEE ASSP Magazine*, vol. 7, no. 2, pp. 32–43, 1990.

[3] J. T. Buck, S. Ha, E. A. Lee, and D. G. Messerschmitt, "Ptolemy: a framework for simulating and prototyping heterogeneous systems," *International Journal of Computer Simulations*, vol. 4, no. 2, pp. 155–182, 1994.

[4] E. A. Lee and D. G. Messerschmitt, "Static scheduling of synchronous dataflow programs for digital signal processing," *IEEE Trans. on Computers*, vol. 36, no. 1, pp. 24–35, 1987.

[5] Telenor Research, "TMN (H.263) Encoder/Decoder Version 2.0," June 1997, ftp://bonde.nta.no/.

[6] P. K. Murthy, S. S. Bhattacharyya, and E. A. Lee, "Joint minimization of code and data for synchronous dataflow programs," *Journal of Formal Methods in System Design*, vol. 11, no. 1, pp. 41–70, 1997.

[7] W. Sung, J. Kim, and S. Ha, "Memory efficient software synthesis from dataflow graph," in *11th International Symposium on System Synthesis (ISSS '98)*, pp. 137–144, Hsinchu, Taiwan, December 1998.

[8] S. S. Bhattacharyya, P. K. Murthy, and E. A. Lee, "APGAN and RPMC: Complementary heuristics for translating DSP block diagrams into efficient software implementations," *Journal of Design Automation for Embedded Systems*, vol. 2, no. 1, pp. 33–60, 1997.

[9] F. J. Kurdahi and A. C. Parker, "REAL: a program for register allocation," in *Proc. 24th ACM/IEEE Design Automation Conference (DAC '87)*, pp. 210–215, Miami Beach, Fla, USA, June 1987.

[10] M. R. Garey, D. S. Johnson, and L. J. Stockmeyer, "Some simplified NP-complete graph problems," *Theoretical Computer Science*, vol. 1, no. 3, pp. 237–267, 1976.

[11] J. Gergov, "Algorithms for compile-time memory optimization," in *Proc. 10th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '99)*, pp. 907–908, Baltimore, Md, USA, January 1999.

[12] E. De Greef, F. Catthoor, and H. D. Man, "Array placement for storage size reduction in embedded multimedia systems," in *Proc. International Conference on Application-Specific Array Processors (ASAP)*, pp. 66–75, Zurich, Switzerland, July 1997.

[13] P. K. Murthy and S. S. Bhattacharyya, "Shared buffer implementations of signal processing systems using lifetime analysis techniques," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 20, no. 2, pp. 177–198, 2001.

[14] S. Ritz, M. Willems, and H. Meyr, "Scheduling for optimum data memory compaction in block diagram oriented software synthesis," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, pp. 2651–2653, Detroit, Mich, USA, May 1995.

[15] F. Gavril, "Algorithms for minimum coloring, maximum clique, minimum covering by cliques, and maximum independent set of a chordal graph," *SIAM Journal on Computing*, vol. 1, no. 2, pp. 180–187, 1972.

[16] S. S. Bhattacharyya and E. A. Lee, "Memory management for dataflow programming of multirate signal processing algorithms," *IEEE Trans. Signal Processing*, vol. 42, no. 5, pp. 1190–1201, 1994.

[17] G. Bilsen, M. Engels, R. Lauwereins, and J. A. Peperstraete, "Cyclo-static data flow," *IEEE Trans. Signal Processing*, vol. 44, no. 2, pp. 397–408, 1996.

Hyunok Oh is a Ph.D. candidate in the School of Computer Science and Engineering at Seoul National University, Korea. He received his B.A. degree (1996) and M.A. degree (1998) in computer engineering from Seoul National University, and he is in a Ph.D. program from 1998 to 2002. He is interested in hardware-software codesign, model of computation, hardware-software cosynthesis, memory optimization, and multimedia applications.

Soonhoi Ha is currently an Associate Professor in the School of Electrical Engineering and Computer Science at Seoul National University. From 1993 to 1994, he worked for Hyundai Electronics Industries Corporation. He received his B.A. degree (1985) and M.A. degree (1987) in electronics engineering from Seoul National University, and Ph.D. (1992) degree in electrical engineering and computer science from University of California, Berkeley. He has worked on Ptolemy project. His research interests include hardware-software codesign, design methodology for embedded systems, and PC clusters. He is a member of the ACM and IEEE Computer Society.

Special Issue on
Human-Activity Analysis in Multimedia Data

Call for Papers

Many important applications of multimedia revolve around the detection of humans and the interpretation of human behavior, for example, surveillance and intrusion detection, automatic analysis of sports videos, broadcasts, movies, ambient assisted living applications, video conferencing applications, and so forth. Success in this task requires the integration of various data modalities including video, audio, and associated text, and a host of methods from the field of machine learning. Additionally, the computational efficiency of the resulting algorithms is critical since the amount of data to be processed in videos is typically large and real-time systems are required for practical implementations.

Recently, there have been several special issues on the human detection and human-activity analysis in video. The emphasis has been on the use of video data only. This special issue is concerned with contributions that rely on the use of multimedia information, that is, audio, video, and, if available, the associated text information.

Papers on the following and related topics are solicited:

- Video characterization, classification, and semantic annotation using both audio and video, and text (if available).
- Video indexing and retrieval using multimedia information.
- Segmentation of broadcast and sport videos based on audio and video.
- Detection of speaker turns and speaker clustering in broadcast video.
- Separation of speech and music/jingles in broadcast videos by taking advantage of multimedia information.
- Video conferencing applications taking advantage of both audio and video.
- Human mood detection, and classification of interactivity in duplexed multimedia signals as in conversations.
- Human computer interaction, ubiquitous computing using multimedia.
- Intelligent audio-video surveillance and other security-related applications.

Authors should follow the EURASIP JASP manuscript format described at the journal site bellow http://www.hindawi.com/GetJournal.aspx?journal=ASP. Prospective authors should submit an electronic copy of their complete manuscript through the EURASIP JASP manuscript tracking system at the following site http://www.hindawi.com/mts/, according to the following timetable:

Manuscript Due	February 1, 2007
Acceptance Notification	June 1, 2007
Final Manuscript Due	October 1, 2007
Publication Date	1st Quarter, 2008

GUEST EDITORS:

A. Enis Cetin, Department of Electrical and Electronics Engineering, Bilkent University, Ankara 06800, Turkey; cetin@ee.bilkent.edu.tr

Eric Pauwels, Signals and Images Research Group, Centre for Mathematics and Computer Science (CWI), 1098 SJ Amsterdam, The Netherlands; eric.pauwels@cwi.nl

Ovidio Salvetti, Institute of Information Science and Technologies (ISTI), Italian National Research Council (CNR), 56124 Pisa, Italy; ovidio.salvetti@isti.cnr.it

Special Issue on

Signal Processing for Location Estimation and Tracking in Wireless Environments

Call for Papers

In recent years, the proliferation of mobile computing devices and wireless technologies has fostered a growing interest in location-aware systems and services. The availability of location information on objects and human beings is critical in many military and civilian applications such as emergency call services, tracking of valuable assets, monitoring individuals with special needs in assisted living facilities, location-assisted gaming (e.g., Geocaching), etc.

Existing positioning systems can be categorized based on whether they are intended for indoor or outdoor applications. Within both of these application areas, there are two major categories of position estimation techniques, as discussed below.

- *Geometric techniques*–Position is estimated by exploiting time of arrival (TOA), time difference of arrival (TDOA), angle of arrival (AOA) or other information derived from the relationship between the geometry of an array of receivers and the modeled propagation characteristics of the transmitted signal.
- *Mapping approaches*–Position is estimated based on comparison of local measurements to a "map" of expected distribution of the measured values. For example, in a wireless LAN application, received signal strength (RSS) might be observed either at the location of the client or at a remote reference point. Mapping approaches are also known as location fingerprinting.

Although geometric approaches have the potential to achieve higher precision than mapping approaches, they generally require direct-path signal reception or accurate environmental information at the receiver and often perform poorly in complex multipath environments. On the other hand, estimation accuracy of mapping approaches is limited by both the accuracy of the reference map and the accuracy of observed measurements. Furthermore, frequent and extensive site-survey measurements are often needed to accommodate the time varying nature of wireless channels, structural changes in the environment, and upgrades of wireless infrastructure.

In addition to snapshots of AOA, TOA, TDOA or RSS measurements, motion models or prior knowledge of structural constraints can often be used to enhance location estimation accuracy for mobile objects by "tracking" location estimates over time. Trackers that integrate such information into the computation of location estimates are generally implemented using techniques such as Kalman filters, particle filters, Markov chain Monte Carlo methods, etc.

The purpose of the proposed special issue is to present a comprehensive picture of both the current state of the art and emerging technologies in signal processing for location estimation and tracking in wireless environments. Papers are solicited on all related aspects from the point of view of both theory and practice. Submitted articles must be previously unpublished and not concurrently submitted for publication on other journals.

Topics of interest include (but are not limited to):

- Received signal strength (RSS), angle-of-arrival (AOA), and time-based location estimation
- Ultrawideband (UWB) location estimation
- Bayesian location estimation and tracking
- Pattern recognition and learning theory approaches to location estimation
- Applications of expectation-maximization (EM) and Markov chain Monte Carlo (MCMC) techniques
- Applications of electromagnetic propagation modeling to location estimation
- Mitigation of errors due to non-line-of-sight propagation
- System design and configuration
- Performance evaluation, performance bounds, and statistical analysis

- Computational complexity and distributed computation
- Distributed location estimation
- Synchronization issues
- Testbed implementation, real-world deployment, and measurement

Authors should follow the EURASIP JASP manuscript format at http://www.hindawi.com/journals/asp/. Prospective authors should submit an electronic copy of their complete manuscript through the EURASIP JASP Manuscript Tracking System at http://www.hindawi.com/mts/, according to the following timetable:

Manuscript Due	March 1, 2007
Acceptance Notification	July 1, 2007
Final Manuscript Due	October 1, 2007
Publication Date	4th Quarter, 2007

GUEST EDITORS:

Richard J. Barton, Department of Electrical and Computer Engineering, University of Houston, Houston, TX 77204, USA; rbarton@uh.edu

Rong Zheng, Department of Computer Science, University of Houston, Houston, TX 77204, USA; rzheng@cs.uh.edu

Sinan Gezici, Mitsubishi Electric Research Labs, Cambridge, MA 02139, USA; gezici@ieee.org

Venugopal V. Veeravalli, Department of Electrical and Computer Engineering, University of Illinois, Urbana, IL 61801, USA; vvv@uiuc.edu

Special Issue on Track Before Detect Algorithms

Call for Papers

Seamless detection and tracking schemes are able to integrate unthresholded (or below target detection threshold) multiple sensor responses over time to detect and track targets in low signal-to-noise ratio (SNR) and high clutter scenarios. These schemes, also called "track-before-detect (TBD)" algorithms are especially suitable for tracking weak targets that would only very rarely cross a standard detection threshold as applied at the sensor level.

Thresholding sensor responses result in a loss of information. Keeping this information allows some TBD approaches to deal with the classical data association problem effectively in high clutter and low SNR situations. For example, in detection scenarios with simultaneous activation/illumination from different signal sources this feature allows the application of triangulation techniques, where in the case of contact tracking approaches essential information about weak targets would often be lost because these targets did not produce signals that cross the normal detection threshold. Extending this example to a multi-sensor network scenario, a TBD algorithm that can use unthresholded (or below threshold) data has the potential to show improved performance compared to an algorithm that relies on thresholded data. In low SNR situations, this can substantially increase performance particularly in the case of a dense multi-target scenario.

Naturally, TBD algorithms consume high computational processing power: An efficient realization and coding of the TBD scheme is mandatory.

Another issue that arises when using the TBD scheme is the quality of the sensor model: Practical experience with thresholded data shows that a coarser modelling of the likelihood function might be sufficient and often leads to robust algorithms. How much have these sensor models to be improved in order to allow the TBD algorithms to exploit the information provided with the unthresholded data?

TBD algorithms that are well known to the tracking community are the likelihood ratio detection and tracking (LRDT), maximum likelihood probabilistic data association (MLPDA), maximum likelihood probabilistic multihypothesis tracking (MLPMHT), Houghtransform based methods and dynamic programming techniques; also related are the probability hypothesis density (PHD), the histogram probabilistic multi- hypothesis tracking (H-PMHT) algorithms, and, of course, various particle filter approaches. Some of these algorithms are capable of tracking extended targets and performing signal estimation in multi-sensor measurements.

The aim of this special issue is to focus on recent developments in this expanding research area. The special issue will focus on one hand on the development and comparison of algorithmic approaches, and on the other hand on their currently ever-widening range of applications such as in active or passive surveillance scenarios (e.g. for object tracking and classification with image and video based sensors, or scenarios involving chemical, electromagnetic and acoustic sensors). Special interest lies in multi-sensor data fusion and/or multi-target tracking applications.

Authors should follow the EURASIP Journal on Advances in Signal Processing manuscript format described at the journal site http://www.hindawi.com/journals/asp/. Prospective authors should submit an electronic copy of their complete manuscript through the EURASIP JASP Manuscript Tracking System at http://www.hindawi.com/mts/, according to the following timetable:

Manuscript Due	April 1, 2007
Acceptance Notification	July 1, 2007
Final Manuscript Due	October 1, 2007
Publication Date	4th Quarter, 2007

GUEST EDITORS:

Yvo Boers, Surface Radar, Thales Nederland B.V. Haaksbergerstraat 49, 7554 PA Hengelo, The Netherlands; yvo.boers@nl.thalesgroup.com

Frank Ehlers, NATO Undersea Research Centre, Viale S. Bartolomeo 400, 19138 La Spezia, Italy; frankehlers@ieee.org

Wolfgang Koch, FGAN-FKIE, Neuenahrer Strasse 20, D53343 Wachtberg, Germany; w.koch@fgan.de

Tod Luginbuhl, Naval Undersea Warfare Center, 1176 Howell Street, Newport, RI 02841-1708, USA; t.e.luginbuhl@ieee.org

Lawrence D. Stone, Metron Inc., 11911 Freedom Drive, Suite 800, Reston, VA 20190, USA; stone@metsci.com

Roy L. Streit, Metron Inc., 11911 Freedom Drive, Suite 800, Reston, VA 20190, USA; streit@metsci.com

Hindawi Publishing Corporation
http://www.hindawi.com

Special Issue on

Advanced Signal Processing and Pattern Recognition Methods for Biometrics

Call for Papers

Biometric identification has established itself as a very important research area primarily due to the pronounced need for more reliable and secure authentication architectures in several civilian and commercial applications. The recent integration of biometrics in large-scale authentication systems such as border control operations has further underscored the importance of conducting systematic research in biometrics. Despite the tremendous progress made over the past few years, biometric systems still have to reckon with a number of problems, which illustrate the importance of developing new biometric processing algorithms as well as the consideration of novel data acquisition techniques. Undoubtedly, the simultaneous use of several biometrics would improve the accuracy of an identification system. For example the use of palmprints can boost the performance of hand geometry systems. Therefore, the development of biometric fusion schemes is an important area of study. Topics related to the correlation between biometric traits, diversity measures for comparing multiple algorithms, incorporation of multiple quality measures, and so forth need to be studied in more detail in the context of multibiometrics systems. Issues related to the individuality of traits and the scalability of biometric systems also require further research. The possibility of using biometric information to generate cryptographic keys is also an emerging area of study. Thus, there is a definite need for advanced signal processing, computer vision, and pattern recognition techniques to bring the current biometric systems to maturity and allow for their large-scale deployment.

This special issue aims to focus on emerging biometric technologies and comprehensively cover their system, processing, and application aspects. Submitted articles must not have been previously published and must not be currently submitted for publication elsewhere. Topics of interest include, but are not limited to, the following:

- Fusion of biometrics
- Analysis of facial/iris/palm/fingerprint/hand images
- Unobtrusive capturing and extraction of biometric information from images/video
- Biometric identification systems based on face/iris/palm/fingerprint/voice/gait/signature
- Emerging biometrics: ear, teeth, ground reaction force, ECG, retina, skin, DNA
- Biometric systems based on 3D information
- User-specific parameterization
- Biometric individuality
- Biometric cryptosystems
- Quality measure of biometrics data
- Sensor interoperability
- Performance evaluation and statistical analysis

Authors should follow the EURASIP JASP manuscript format described at http://www.hindawi.com/journals/asp/. Prospective authors should submit an electronic copy of their complete manuscript through the EURASIP JASP manuscript tracking system at http://www.hindawi.com/mts/, according to the following timetable:

Manuscript Due	May 1, 2007
Acceptance Notification	September 1, 2007
Final Manuscript Due	December 1, 2007
Publication Date	1st Quarter, 2008

GUEST EDITORS:

Nikolaos V. Boulgouris, Department of Electronic Engineering, Division of Engineering, King's College London, London WC2R 2LS, UK; nikolaos.boulgouris@kcl.ac.uk

Juwei Lu, EPSON Edge, EPSON Canada Ltd., Toronto, Ontario M1W 3Z5, Canada; juwei@ieee.org

Konstantinos N. Plataniotis, The Edward S. Rogers Sr. Department of Electrical and Computer Engineering, University of Toronto, Toronto, Ontario, Canada, M5S 3G4; kostas@dsp.utoronto.ca

Arun Ross, Lane Department of Computer Science & Electrical Engineering, West Virginia University, Morgantown WV, 26506, USA; arun.ross@mail.wvu.edu

Special Issue on Signal Processing for Data Converters

Call for Papers

Data converters (ADCs and DACs) ultimately limit the performance of today's communication systems. New concepts for high-speed, high-resolution, and power-aware converters are therefore required, which also lead to an increased demand for high-speed and high-resolution sampling systems in the measurement industry. Present converter technologies operate on their limits, since the downscaling of IC technologies to deep submicron technologies makes their design increasingly difficult. Fortunately, downscaling of IC technologies allows for using additional chip area for digital signal processing algorithms with hardly any additional costs. Therefore, one can use more elaborate signal processing algorithms to improve the conversion quality, to realize new converter architectures and technologies, or to relax the requirements on the analog design. Pipelined ADCs constitute just one example of converter technology where signal processing algorithms are already extensively used. However, time-interleaved converters and their generalizations, including hybrid filter bank-based converters and parallel sigma-delta-based converters, are the next candidates for digitally enhanced converter technologies, where advanced signal processing is essential. Accurate models constitute one foundation of digital corrected data converters. Generating and verifying such models is a complex and time-consuming process that demands high-performance instrumentation in conjunction with sophisticated software defined measurements.

The aim of this special issue is to bring forward recent developments on signal processing methods for data converters. It includes design, analysis, and implementation of enhancement algorithms as well as signal processing aspects of new converter topologies and sampling strategies. Further, it includes design, analysis, and implementation of software defined measurements for characterization and modeling of data converters.

Topics of interest include (but are not limited to):

- Analysis, design, and implementation of digital algorithms for data converters
- Analysis and modeling of novel converter topologies and their signal processing aspects
- Digital calibration of data converters
- Error identification and correction in time-interleaved ADCs and their generalizations
- Signal processing for application-specific data converters (communication systems, measurement systems, etc.)
- New sampling strategies
- Sampling theory for data converters
- Signal processing algorithms for data converter testing
- Influence of technology scaling on data converters and their design
- Behavioral models for converter characterization
- Instrumentation and software defined measurements for converter characterization

Authors should follow the EURASIP JASP manuscript format at http://www.hindawi.com/journals/asp/. Prospective authors should submit an electronic copy of their complete manuscript through the EURASIP JASP Manuscript Tracking System at http://www.hindawi.com/mts/, according to the following timetable:

Manuscript Due	May 1, 2007
Acceptance Notification	September 1, 2007
Final Manuscript Due	December 1, 2007
Publication Date	1st Quarter 2008

GUEST EDITORS:

Christian Vogel, Signal Processing and Speech Communication Laboratory, Graz University of Technology, 8010 Graz, Austria; vogel@tugraz.at

Håkan Johansson, Department of Electrical Engineering, Linköping University, 581 83 Linköping, Sweden; hakanj@isy.liu.se

Boris Murmann, Department of Electrical Engineering, Stanford University, Stanford, CA 94305-4070, USA; murmann@stanford.edu

Hindawi Publishing Corporation
http://www.hindawi.com

Special Issue on Distributed Space-Time Systems

Call for Papers

Diversity is a powerful technique to mitigate channel fading and to improve robustness to cochannel interference in a wireless network. Space-time wireless systems traditionally use multiple colocated antennas at the transmitter and receiver along with appropriate signal design (also known as space-time coding) to realize spatial diversity in the link. Typically this diversity can augment any frequency and time diversity available to the receiver. Multiple antennas also offer the ability to use spatial multiplexing to dramatically increase the data rate.

A recent development in this area aims at dispensing with the need for colocated antennas. Popularly known as the cooperative diversity technique, this uses the antennas at multiple user terminals in a network in the form of a virtual antenna array to realize spatial diversity in a distributed fashion. Such techniques create new challenges in the design of wireless systems.

The purpose of this call for papers is to address some of these challenges such as new protocols for cooperative diversity, cross-layer design, cooperative multiplexing, space-time coding for distributed antennas, cooperative channel estimation and equalization, selecting the right users for participating in a cooperative network, modulation specific issues like OFDMA and CDMA, and distributed space-time processing for sensor networks.

Papers on the following and related topics are solicited for this special issue:

- New protocols for cooperative diversity systems
- Cross-layer protocol design
- Signal design for distributed space-time systems
- Cooperative channel estimation and equalization
- Cooperative MIMO systems
- Distributed space-time processing for sensor networks
- Power allocation in distributed space-time systems
- Fast algorithms and efficient architectures for virtual MIMO receivers
- Energy efficient relay network architectures

Authors should follow the EURASIP Journal on Advances in Signal Processing manuscript format described at the journal site http://www.hindawi.com/journals/asp/. Prospective authors should submit an electronic copy of their complete manuscript through the EURASIP JASP Manuscript Tracking System at http://www.hindawi.com/mts/, according to the following timetable:

Manuscript Due	May 1, 2007
Acceptance Notification	September 1, 2007
Final Manuscript Due	December 1, 2007
Publication Date	1st Quarter, 2008

GUEST EDITORS:

Ranjan K. Mallik, Department of Electrical Engineering, Indian Institute of Technology - Delhi, Hauz Khas, New Delhi 110016, India; rkmallik@ee.iitd.ernet.in

Arogyaswami J. Paulraj, Department of Electrical Engineering, Stanford University, 232 Packard, 350 Serra Mall, Stanford, CA 94305, USA; apaulraj@stanford.edu

Mrityunjoy Chakraborty, Department of Electronics and Electrical Communication Engineering, Indian Institute of Technology - Kharagpur, Kharagpur 721302, West Bengal, India; mrityun@ece.iitkgp.ernet.in

Keith Q. T. Zhang, Department of Electronic Engineering, City University of Hong Kong, Tat Chee Avenue, Kowloon, Hong Kong; eekzhang@cityu.edu.hk

George K. Karagiannidis, Department of Electrical and Computer Engineering, Aristotle University of Thessaloniki, 54 124 Thessaloniki, Greece; geokarag@auth.gr

Special Issue on

Cooperative Localization in Wireless Ad Hoc and Sensor Networks

Call for Papers

One of the major requirements for most applications based on wireless ad hoc and sensor networks is accurate node localization. In fact, sensed data without position information is often less useful.

Due to several factors (e.g., cost, size, power), only a small fraction of nodes obtain the position information of the anchor nodes. In this case, a node has to estimate its position without a direct interaction with anchor nodes and a cooperation between nodes is needed in a multihop fashion. In some applications, none of the nodes are aware of their absolute position (anchor-free) and only relative coordinates are estimated instead.

Most works reported in the literature have studied cooperative localization with the emphasis on algorithms. However, very few works give emphasis on the localization as estimation or on the investigation of fundamental performance limits as well as on experimental activities. In particular, the fundamental performance limits of multihop and anchor-free positioning in the presence of unreliable measurements are not yet well established. The knowledge of such limits can also help in the design and comparison of new low-complexity and distributed localization algorithms. Thus, measurement campaigns in the context of cooperative localization to validate the algorithms as well as to derive statistical models are very valuable.

The goal of this special issue is to bring together contributions from signal processing, communications and related communities, with particular focus on signal processing, new algorithm design methodologies, and fundamental limitations of cooperative localization systems. Papers on the following and related topics are solicited:

- anchor-based and anchor-free distributed and cooperative localization algorithms that can cope with unreliable range measurements
- derivation of fundamental limits in multihop and anchor-free localization scenarios
- new localization algorithms design methodologies based, for example, on statistical inference and factor graphs
- low-complexity and energy-efficient distributed localization algorithms
- distributed ranging and time synchronization techniques
- measurement campaigns and statistical channel modeling
- algorithm convergence issues
- UWB systems
- localization through multiple-antenna systems
- experimental results

Authors should follow the EURASIP JASP manuscript format at http://www.hindawi.com/journals/asp/. Prospective authors should submit an electronic copy of their complete manuscript through the EURASIP JASP Manuscript Tracking System at http://www.hindawi.com/mts/, according to the following timetable:

Manuscript Due	August 1, 2007
Acceptance Notification	December 1, 2007
Final Manuscript Due	March 1, 2008
Publication Date	2nd Quarter 2008

GUEST EDITORS:

Davide Dardari, WiLAB, Department of Electronics, Computer Science and Systems (DEIS), University of Bologna at Cesena, Italy; ddardari@ieee.org

Chia-Chin Chong, DoCoMo Communications Laboratories USA, Inc., Palo Alto, CA 94304-1201, USA; cchong@docomolabs-usa.com

Damien B. Jourdan, Athena Technologies, Inc., Warrenton, VA 20187, USA; jourdan@alum.mit.edu

Lorenzo Mucchi, Department of Electronics and Telecommunications, University of Florence, 50121 Florence, Italy; mucchi@lenst.det.unifi.it

Special Issue on Multimedia over Wireless Networks

Call for Papers

Scope

In recent years there has been a tremendous increase in demand for multimedia delivered over wireless networks. The design and capabilities of the mobile devices and the services being offered reflect the increase in multimedia usage in the wireless setting. Applications that are in the process of becoming essential to users include video telephony, gaming, or TV broadcasting. This trend creates great opportunities for identifying new wireless multimedia applications, and for developing advanced systems and algorithms to support these applications. Given the nature of the channel and of the mobile devices, issues such as scalability, error resiliency, and energy efficiency are of great importance in applications involving multimedia transmission over wireless networks.

The papers in this issue will focus on state-of-the-art research on all aspects of wireless multimedia communications. Papers showing significant contributions are solicited on topics including but are not limited to:

- Error resilience and error concealment algorithms
- Rate control for wireless multimedia coding
- Scalable coding and transmission
- Joint source-channel coding
- Joint optimization of power consumption and rate-distortion performance
- Wireless multimedia traffic modeling
- Wireless multimedia streaming
- Wireless multimedia coding
- QoS for wireless multimedia applications
- Distributed multimedia coding

Authors should follow the EURASIP Journal on Wireless Communications and Networking manuscript format described at http://www.hindawi.com/journals/wcn/. Prospective authors should submit an electronic copy of their complete manuscript through the EURASIP Journal on Wireless Communications and Networking's Manuscript Tracking System at http://www.hindawi.com/mts/, according to the following timetable:

Manuscript Due	March 1, 2007
Acceptance Notification	July 1, 2007
Final Manuscript Due	October 1, 2007
Publication Date	4th Quarter, 2007

GUEST EDITORS:

Dan Lelescu, R&D lab, Imaging Division, Micron Technology, Inc., 2125 O'Nel Drive, San Jose, CA 9513, USA; danlelescu@yahoo.com

Peter Schelkens, Deptartment of Electronics and Information Processing (ETRO), Vrije Universiteit Brussel (VUB), Pleinlaan 2, 1050 Brussel, Belgium; Peter.Schelkens@vub.ac.be

Kameswara Rao Namuduri, Deptartment of Electrical and Computer Engineering, Wichita State University, 1845 Fairmount, Wichita, 67260 KS, USA; kamesh.namuduri@wichita.edu

EURASIP JOURNAL ON IMAGE AND VIDEO PROCESSING

Special Issue on Anthropocentric Video Analysis: Tools and Applications

Call for Papers

Humans are a basic entity in most videos. Lately, there has been increased interest in devising automated video analysis algorithms that aim to extract, efficiently describe, and organize information regarding the state or state transition of individuals (identity, emotional state, activity, position and pose, etc), interactions between individuals (dialogue, gestures, engagement into collaborative or competitive activities like sports), physical characteristics of humans (anthropometric characteristics, 3D head/body models), and so forth. Such information can be utilized in a multitude of important applications that include, but are not limited to:

- Human computer interaction, ubiquitous computing
- Video characterization, classification, and semantic annotation
- Video indexing and retrieval
- Temporal video segmentation (shot and scene boundary detection) and summarization
- Intelligent video surveillance, access control, and other security related applications

High quality and original contributions on the following (nonexhaustive) list of topics related to anthropocentric video analysis and its applications are solicited:

- Detection and tracking of humans or human body parts
- Action recognition and human behavior analysis
- Emotional state recognition
- Anthropocentric video characterization, semantic annotation, indexing, retrieval, temporal segmentation and summarization
- Efficient description schemes for anthropocentric video information
- Dialogue detection, LiP activity detection, visual speech recognition
- Hand gesture recognition
- 3D modeling of humans
- Person verification and recognition

Authors should follow the EURASIP JIVP manuscript format at http://www.hindawi.com/journals/ivp/. Prospective authors should submit an electronic copy of their complete manuscripts through the EURASIP JIVP manuscript tracking system at http://www.hindawi.com/mts/, according to the following timetable:

Manuscript Due	February 1, 2007
Acceptance Notification	June 1, 2007
Final Manuscript Due	September 1, 2007
Publication Date	4th Quarter, 2007

GUEST EDITORS:

Nikos Nikolaidis, Department of Informatics, Aristotle University of Thessaloniki, Greece; nikolaid@aiia.csd.auth.gr

Maja Pantic, Department of Computing, Imperial College London, UK; m.pantic@imperial.ac.uk

Ioannis Pitas, Department of Informatics, Aristotle University of Thessaloniki, Greece; pitas@aiia.csd.auth.gr

IEEE ICME 2007 Call for Papers
2007 International Conference on Multimedia & Expo (ICME)

July 2-5, 2007
Beijing International Convention Center, Beijing, China

Sponsored by: Circuits and Systems Society, Communications Society, Computer Society, and Signal Processing Society.

General Co-Chairs
Xinhua Zhuang, *U. Missouri-Columbia, USA*
Wen Gao, *Peking University, China*

Technical Program Chair
Yun Q. Shi, *NJIT, USA*

Technical Program Vice-Chairs
Mark Liao *(Circ. & Sys. Society) Acad. Sinica*
Yu-Hen Hu *(Comm. Society) U. Wisconsin, USA*
Philip Sheu *(Comp. Society) UC Irvine, USA*
Joern Ostermann *(Sig. Pr. Soc.) LUH, Germany*

Conference Manager
Hong Man, *Stevens Inst. Tech., USA*

Special Session Chairs
John Aa. Sorenson, *ECC, Denmark*
Shipeng Li, *Microsoft Research Asia, China*

Tutorial Chairs
Ming-Ting Sun, *University of Washington, USA*
Oscar Au, *HKUST, China*

Finance Chairs
Belle Tseng, *NEC Lab America, USA*
Shiqiang Yang, *Tsinghua University, China*

Publicity Chairs
Bo Li, *Beihang University, China*
Ed Wong, *Brooklyn Poly. Univ., USA*

Registration Chair
Yun He, *Tsinghua University, China*
Hong Man, *Stevens Inst. Tech., USA*

Electronic Media Chairs
Zicheng Liu, *Microsoft Research, USA*
Chiren Shyu, *U. Missouri-Columbia, USA*

Publications Chairs
Wenjun Zeng, *U. Missouri-Columbia, USA*
Yuanchun Shi, *Tsinghua University, China*

Demo-Exhibit Chairs
Jian Lu, *Vobile Inc., USA*
Feng Wu, *Microsoft Research Asia, China*

Local Arrangement Chairs
Hanqing Lu, *IA of CAS, China*
Xilin Chen, *ICT of CAS, China*

North America Liaison
Heather Yu, *Panasonic, USA*
Yong Rui, *Microsoft, China*

Europe Liaison
Murat Kunt, *EPFL, Switzerland*
Jean-Luc Dugelay, *EUROCOM, France*

IEEE International Conference on Multimedia & Expo is a major annual international conference with the objective of bringing together researchers, developers, and practitioners from academia and industry working in all areas of multimedia. ICME serves as a forum for the dissemination of state-of-the-art research, development, and implementations of multimedia systems, technologies and applications. ICME is co-sponsored by four IEEE societies including the Circuits and Systems Society, the Communications Society, the Computer Society, and the Signal Processing Society. The conference will feature world-class plenary speakers, exhibits, special sessions, tutorials, and paper presentations.

Prospective authors are invited to submit a four-page paper in double-column format including authors' names, affiliations, and a short abstract. Only electronic submissions will be accepted. Topics include but are not limited to:

- Audio, image, video processing
- Virtual reality and 3-D imaging
- Signal processing for media integration
- Multimedia communications and networking
- Multimedia security and content protection
- Multimedia human-machine interface and interaction
- Multimedia databases
- Multimedia computing systems and appliances
- Hardware and software for multimedia systems
- Multimedia standards and related issues
- Multimedia applications
- Multimedia and social media on the Internet

A number of awards will be presented to the Best Papers and Best Student Papers at the conference. Participation for special sessions and tutorial proposals are encouraged.

SCHEDULE

- Special Session Proposals Due: **December 1, 2006**
- Tutorial Proposals Due: **December 1, 2006**
- Regular Paper Submissions Due: **January 5, 2007**
- Notification of Acceptance: **March 19, 2007**
- Camera-Ready Papers Due: **April 16, 2007**

Check the conference website http://www.icme2007.org for updates.

International Advisory Board

Sadaoki Furui, *Tokyo Inst. Tech., Japan* (Chair)
Ming Liou, *HKUST, China* (Co-Chair)
Peter Pirsch, *LUH, Germany* (Co-Chair)
Jan Biemond, *Delft Univ. Tech., Netherlands*
Shih-Fu Chang, *Columbia Univ., USA*
Rama Chellappa, *University of Maryland, USA*
Chang-Wen Chen, *Florida Inst. Tech., USA*
Liang-Gee Chen, *National Taiwan University*
Robert M. Haralick, *City Univ. of New York, USA*
T. S. Huang, *UIUC, USA*
Anil Jain, *Michigan State University, USA*
Ramesh Jain, *UC Irvine, USA*
Chung-Sheng Li, *IBM Watson Research, USA*
Xing-Gang Lin, *Tsinghua Univ., China*
K. J. Ray Liu, *University of Maryland, USA*
Songde Ma, *Ministry of Science and Technology, China*
Timothy K. Shih, *Tamkang University*
T. Sikora, *Technical Univ. Berlin, Germany*
Ming-Ting Sun, *Univ. Washington, USA*
Qi Tian, *Institute for Inforcomm Research, Singapore*
B. W. Wah, *UIUC, USA*
Hong-Jiang Zhang, *Microsoft, China*
Ya-Qin Zhang, *Microsoft, China*

3DTV CONFERENCE 2007

THE TRUE VISION - CAPTURE, TRANSMISSION AND DISPLAY OF 3D VIDEO

May 7-9, 2007, KICC Conference Center, Kos Island, Greece

First Call For Papers

Creating exact 3D moving images as ghost-like replicas of 3D objects has been an ultimate goal in video science. Capturing 3D scenery, processing the captured data for transmission, and displaying the result for 3D viewing are the main functional components. These components encompass a wide range of disciplines: imaging and computer graphics, signal processing, telecommunications, electronics, optics and physics are needed.

The objective of the **3DTV-Conference** is to bring together researchers and developers from academia and industry with diverse experience and activity in distinct, yet complementary, areas so that full scale 3D video capabilities are seemlessly integrated.

Topics of Interest

3D Capture and Processing
- 3D time-varying scene capture technology
- Multi-camera recording
- 3D photography algorithms
- Synchronization and calibration of camera arrays
- 3D view registration
- Multi-view geometry and calibration
- Holographic camera techniques
- 3D motion analysis and tracking
- Surface modeling for 3-D scenes
- Multi-view image and 3D data processing

3D Transmission
- Systems, architecture and transmission aspects of 3D
- 3D streaming
- Error-related issues and handling of 3d video
- Hologram compression
- Multi-view video coding
- 3D mesh compression
- Multiple description coding for 3D
- Signal processing for diffraction and holographic 3DTV

3D Visualization
- 3D mesh representation
- Texture and point representation
- Object-based representation and segmentation
- Volume representation
- 3D motion animation
- Dense stereo and 3D reconstruction
- Stereoscopic display techniques
- Holographic display technology
- Reduced parallax systems and integral imaging
- Underlying optics and VLSI technology
- Projection and display technology for 3D videos
- Human factors

3D Applications
- 3D imaging in virtual heritage and virtual archaeology
- 3D Teleimmersion and remote collaboration
- Augmented reality and virtual environments
- 3D television, cinema, games and entertainment
- Medical and biomedical applications
- 3D Content-based retrieval and recognition
- 3D Watermarking

Paper Submission

Prospective contributors are invited to submit full papers electronically using the on-line submission interface, following the instructions available at http://www.3dtv-con.org. Papers should be in Adobe PDF format, written in English, with no more than four pages including figures, using a font size of 11. Conference proceedings will be published online by the IEEE.

Important Dates

1 December 2006	Special sessions and tutorials proposals
15 December 2006	Regular Paper submission
9 February 2007	Notification of acceptance
2 March 2007	Submission of camera-ready papers

3DTV NoE | ITI-CERTH | Information Society Technologies | Institute of Electrical and Electronics Engineers | European Association for Signal and Image Processing | MPEG-IF



General Chairs
Georgios Triantafyllidis
Centre for Research and Technology Hellas, GR
Levent Onural,
Bilkent University, TR

Technical Chairs
Aljoscha Smolic,
Fraunhofer Gesellschaft, DE
Jörn Ostermann
Leibniz University of Hannover, DE
John Watson
University of Aberdeen, UK
Reha Civanlar
Koc University, TR
Thomas Sikora,
Technical University of Berlin, DE

Finance Chair
Nikos Grammalidis
Centre for Research and Technology Hellas, GR

Publication Chair
Xenophon Zabulis
Centre for Research and Technology Hellas, GR

Publicity Chair
Atanas Gotchev
Tampere University of Technology, FI

Industry Liaison
Ismo Rakkolainen
FogScreen, FI

American Liaison
Yiannis Aloimonos
University of Maryland, US

Far East Liaison
Jinwoong Kim
Electronics and Telecommunications Research Institute (ETRI), KR

Special Sessions and Tutorials Chair
Aydin Alatan
Middle East Technical University, TR

Webmaster
Georgios Litos
Centre for Research and Technology Hellas, GR

5th International Symposium on

Image and Signal Processing and Analysis ISPA 2007

September 27-29, 2007, Istanbul, Turkey

http://www.isispa.org

Call for Papers

The 5th International Symposium on Image and Signal Processing and Analysis (ISPA 2007) will take place in Istanbul, Turkey, from September 27-29, 2007. The scientific program of the symposium consists of invited lectures, regular papers, and posters. The aim of the symposium is to foster interaction of researchers and exchange of new ideas. Prospective authors are invited to submit their manuscripts reporting original work, as well as proposals for special sessions.

Co-Operations and Co-Sponsorships

- European Association for Signal Processing (EURASIP)
- IEEE Region 8*

Symposium Topics

A. Image and Video Processing
B. Image and Video Analysis
C. Image Formation and Reproduction
D. Signal Processing
E. Signal Analysis
F. Applications

For a detailed list of subtopics please visit ISPA 2007 web site.

Important Dates

Submission of full paper: February 15, 2007
Notification of acceptance/rejection: April 15, 2007
Submission of camera-ready papers and registration: May 15, 2007

Symposium Venue

Located in the center of the Old World, Istanbul is one of the world's great cities famous for its historical monuments and scenic beauties. It is the only city in the world which spreads over two continents: it lies at a point where Asia and Europe are separated by a narrow strait - the Bosphorus. Istanbul has been the cradle for many civilizations for over 2500 years and has a very rich history. It has been the capital of three great empires, the Roman, Byzantine and Ottoman empires, and for more than 1,600 years over 120 emperors and sultans ruled the world from here. Istanbul is the heart of Turkey with respect to entertainment, culture, education, shopping, imports and exports, tourism and the arts. The symposium will be organized in the congress center of the Bogazici University.

Paper Submission Procedure

Papers including title, author list and affiliations, figures, results, and references should not exceed six A4 pages. Detailed instructions for electronic submission are available on the ISPA web site. All papers will be subject to a peer-review process with at least two reviewers. All accepted papers will be published in the symposium proceedings in book form and on CD-ROM, which will be available through IEEE Publications Center and in IEEEXplore digital library.

Call for Special Session Proposals

Prospective organizers of special sessions are invited to send proposals to Special Session Co-Chairs, according to instructions provided on the ISPA web site. The aim of a special session is to provide an overview of the state-of-the-art and current research directions in specific fields of image and signal processing and analysis.

Best Student Paper Award

Best Student Paper Award in the amount of 300 EUR will be given to a student author. The student's name must appear first on the paper and the paper must be presented at the symposium to be eligible for the award.

* request pending

Organizing Institutions
Bogazici University, Turkey
Sabanci University, Turkey
University of Zagreb, Croatia

Honorary Chair
Hrvoje Babic
University of Zagreb, Croatia

General Co-Chairs
Aytul Ercil
aytulercil@sabanciuniv.edu
Sabanci University, Turkey

Sven Loncaric
sven.loncaric@fer.hr
University of Zagreb, Croatia

Program Co-Chairs
Maria Petrou
maria.petrou@imperial.ac.uk
Imperial College London, UK

Tapio Saramaki
tapio.saramaki@tut.fi
Tampere Univ. of Tech., Finland

Special Sessions Co-Chairs
Bulent Sankur
bulent.sankur@boun.edu.tr
Bogazici University, Turkey

Robert Bregovic
bregovic@cs.tut.fi
Tampere Univ. of Tech., Finland

Publicity Chair
Enis Cetin
cetin@bilkent.edu.tr
Bilkent University, Turkey

Local Arrangements Chair
Aysın Ertuzun
email address
ertuz@boun.edu.tr
Bogazici University

Publications Chair
Marko Subasic
marko.subasic@fer.hr
University of Zagreb, Croatia

Registration Chair
Gulbin Akgun
gulbin@sabanciuniv.edu
Sabancı University, Turkey

Information Tech. Chair
Vjekoslav Levacic
vjekoslav.levacic@fer.hr
University of Zagreb, Croatia

Symposium Secretariat
Gulbin Akgun
Sabanci University, Faculty of Engineering and Natural Sciences
Tel: 216 483 9543
Fax: 216 483 9550
secretariat@isispa.org

ISSPA 2007

International Symposium on Signal Processing and its Applications

in conjunction **with the International Conference on Information Sciences, Signal Processing and their Applications**

12 – 15 February 2007, Millennium Hotel, Sharjah, U.A.E.

ISSPA General Chair & Steering Committee Chair
B. Boashash
University of Sharjah, UAE
The University of Queensland, Australia

Conference Chair
M. Bettayeb
University of Sharjah, UAE

Conference Vice-Chair
S. Al-Araji
Etisalat University College, UAE

Technical Program Co-Chairs
K. Assaleh
American University of Sharjah, UAE
M. IbnKahla
Queens University, Canada
I. Tabus
Tampere University of Technology, Finland

Plenary Sessions
S. Mitra
University of California, Santa Barbara, USA

Special Sessions
M. Cheriet, Chair
University of Quebec, Canada
M. Barkat, Co-Chair
American University of Sharjah, UAE
L. Karam, Co-Chair
Arizona State University, USA

Tutorials
M. El-Tarhuni
American University of Sharjah, UAE

Industry Liaison
H. Al-Ahmad
Etisalat University College, UAE

Publications
M. Al-Qutayri
Etisalat University College, UAE

Publicity
M. Al-Mualla
Etisalat University College, UAE

Sponsorship & Exhibits
K. Al-Midfa
Etisalat University College, UAE

Student Sessions
A. Elwakil
University of Sharjah, UAE

Finance & Registration
C. B. Yahya and A. Darwish
University of Sharjah, UAE

Local Arrangements
I. Kamel
University of Sharjah, UAE

Social Events
I. Shahin
University of Sharjah, UAE
H. A. Al-Hammady
Etisalat University College, UAE

Web and IT
B. Soudan
University of Sharjah, UAE

International Liaisons
S. Anderson, Australia
DSTO, Australia
S. Furui, Asia and Pacific
Tokyo Institute of Technology, Japan
M. Gabbouj, Europe
Tampere University of Technology, Finland
M. Jaidane, Africa
ENIT, Tunisia
Y. Zhang, America
Villanova University, USA

Call For Participation

ISSPA 2007 marks the 20th anniversary of launching the first ISSPA in 1987 in Brisbane, Australia. Since its inception, ISSPA has provided, through a series of 8 symposia, a high quality forum for engineers and scientists engaged in research and development of Signal and Image Processing theory and applications. Effective 2007, ISSPA will extend its scope to add the new track of information sciences. Hence, the intention that the previous full name of ISSPA is replaced after 2007 by the following new full name:
International Conference on Information Sciences, Signal Processing and their Applications. ISSPA is an IEEE indexed conference.

ISSPA 2007 is organized by the University of Sharjah, College of Engineering, Etisalat University College and the American University of Sharjah.

The regular technical program will run for three days along with an exhibition of signal processing and information sciences products. In addition, tutorial sessions will be held on the first day of the symposium. Presentations will be given in the following topics:

1. Filter Design Theory and Methods	11. Multimedia Signal Processing	21. Signal Processing for Bioinformatics
2. Multirate Filtering & Wavelets	12. Nonlinear signal processing	22. Signal Processing for Geoinformatics
3. Adaptive Signal Processing	13. Biomedical Signal and Image Processing	23. Biometric Systems and Security
4. Time-Frequency/Time-Scale Analysis	14. Image and Video Processing	24. Machine Vision
5. Statistical Signal & Array Processing	15. Image Segmentation and Scene Analysis	25. Data visualization
6. Radar & Sonar Processing	16. VLSI for Signal and Image Processing	26. Data mining
7. Speech Processing & Recognition	17. Cryptology, Steganography, and Digital Watermarking	27. Sensor Networks and Sensor Fusion
8. Fractals and Chaos Signal Processing	18. Image indexing & retrieval	28. Signal Processing and Information Sciences Education
9. Signal Processing in Communications	19. Soft Computing & Pattern Recognition	29. Others
10. Signal processing in Networking	20. Natural Language Processing	30. Special Sessions

Prospective authors were invited to submit ***full length*** *(four pages) papers* ***via the conference website*** *for presentation in any of the areas listed above (showing area in submission). Submission of proposals for student session, tutorials and sessions on special topics were sent to the conference secretary. All articles submitted to ISSPA 2007 are* ***peer-reviewed*** *using a* ***blind review process*** *by at least two independent reviewers.*

For more details see

www.isspa2007.org/

Important Deadlines:

Full Paper Submission:
October 14, 2006
Tutorials/Special Sessions Submission:
October 14, 2006
Notification of Acceptance:
December 3, 2006
Final Accepted Paper Submission:
December 19, 2006

Conference Secretary

A-K. Hamid
University of Sharjah, UAE
Tel : +971 6 5050932
Fax :+971 6 5050872
E-mail: akhamid@sharjah.ac.ae

IEEE

EURASIP

15th European Signal Processing Conference

EUSIPCO 2007

September 3-7, 2007, Poznań, Poland

Just in the centre of Europe!

European Association for Signal, Speech and Image Processing

EURASIP

CALL FOR PAPERS

The 2007 European Signal Processing Conference (EUSIPCO-2007) is the fifteenth in a series of conferences promoted by EURASIP, the European Association for Signal, Speech, and Image Processing. The conference will be organized by Poznań University of Technology, Faculty of Electronics and Telecommunications Chair of Multimedia Telecommunications and Microelectronics and PTETiS Society in Conference Center at Poznań International Fair.

As usual, EUSIPCO-2007 areas of interest will cover all aspects of signal processing theory and applications as listed below. Proposals for special sessions and tutorials are strongly encouraged. Accepted papers will be published in the proceedings of EUSIPCO-2007. Acceptance will be based on quality, relevance and originality.

The conference topics include:

- Audio and Electroacoustics
- Design and Implementation of Signal Processing Systems
- DSP Applications and Embedded Systems
- Emerging Technologies in Signal Processing
- Signal Processing for Communications
- Image and Multidimensional Signal Processing
- Medical Imaging
- Image and Video Analysis
- Multimedia Signal Processing
- Speech Processing and Coding
- Image, Video and Audio Compression
- Nonlinear Signal Processing
- Sensor Array and Multichannel Processing
- Signal Detection and Estimation
- Signal Processing Applications (Biology, Geophysics, Seismic, Radar, Sonar, Remote Sensing, Astronomy, Bio-informatics, Positioning, etc.)
- Signal Processing Algorithms and their Implementations in Communication Systems
- Hardware Solutions for Signal Processing
- Education on Signal Processing

Submission

Procedures to submit a paper, proposals for sessions/tutorials, can be found at www.eusipco2007.org. Submitted papers must be final, full papers, no more than five pages long all inclusive and strictly conforming to the format specified on the EUSIPCO web site.

Best Student Paper Awards

Student authors who appear as first authors in a paper may enter the student paper contest.

Important Dates (updated)

Proposals for Special Sessions and Tutorials:	December 11, 2006
Electronic submission of Full papers (4 pages A4):	February 5, 2007
Notification of Acceptance:	May 11, 2007
Submission of Camera-Ready Papers and Registration:	June 10, 2007

www.eusipco2007.org

Chairman
Marek Domański
domanski@et.put.poznan.pl
Poznań University of Technology, Poland

Technical Program Chairmen
Ryszard Stasiński
rstasins@et.put.poznan.pl
Maciej Bartkowiak
mbartkow@et.put.poznan.pl
Poznań University of Technology, Poland

Special Sessions
Maciej Niedźwiecki
maciekn@eti.pg.gda.pl
Gdańsk University of Technology, Poland

Tutorials
Tomasz Zieliński
tzielin@uci.agh.edu.pl
AGH University of Science and Technology, Poland

Local Arrangements
Sławomir Maćkowiak
smack@et.put.poznan.pl
Poznań University of Technology, Poland

Overseas Liason
Andrzej Cichocki
cia@brain.riken.jp
Brain Science Institute RIKEN, Japan

Wen Gao
wgao@pku.edu.cn
Peking University, China

Yo-Sung Ho
hoyo@gist.ac.kr
Gwangju Institute of Science and Technology, Korea

Janusz Konrad
jkonrad@bu.edu
Boston University, USA

Masayuki Tanimoto
tanimoto@nuee.nagoya-u.ac.jp
Nagoya University, Japan

Scientific Secretariat
office@eusipco2007.org
Poznań University of Technology
Chair of Multimedia Telecommunications and Microelectronics
Polanka 3, 60-965 Poznań
POLAND

POLITECHNIKA POZNAŃSKA

PTETiS

About EURASIP:

The European Association for Signal, Speech and Image Processing (www.eurasip.org) was founded on 1 September 1978 to: "improve communication between groups and individuals that work within the multidisciplinary, fast growing field of Signal Processing in Europe and elsewhere, and to exchange and disseminate information in the field all over the world." The association exists to promote the efforts of researchers by providing a learned and professional platform for dissemination and discussion of all aspects of signal processing. EURASIP is a non profit organization which is governed by its Administrative Committee (AdCom).

EURASIP Areas of Interest:

- Continuous and discrete time signal theory
- Applications of signal processing
- Systems and technology
- Speech communication
- Image processing and communication.

EURASIP sponsors and co-sponsors a number of conferences within Europe and the rest of the world each year. The main event is EUSIPCO (European Signal Processing Conference), which is now recognized as one of the premier signal processing conferences, attracting delegates and papers from all over the world. The venues of consecutive conferences are: Lausanne, Switzerland (1980); Erlangen, Germany (1983); Hague, the Netherlands (1986); Grenoble, France (1988); Barcelona, Spain (1990); Brussels, Belgium (1992); Edinburgh, UK (1994); Trieste, Italy (1996); Rhodes, Greece (1998); Tampere, Finland (2000); Toulouse, France (2002); Vienna, Austria (2004); Antalya, Turkey (2005); Florence, Italy (2006).
The 2007 event will be held in Poznań, Poland.

About Poznań

Poznań, a capital of Wielkopolska province, is the fifth biggest city in Poland with population of 580 000. It is halfway between Berlin and Warsaw and it is older than each one of them. Poznań is easily accessible, since it is located in central Europe and it is easy to get there both from Western and Eastern part of the continent and also the rest of the world.
Poznań-Ławica International Airport is situated only 6 km from the conference venue. There are a lot of direct flights to many of European cities. The conference site is in the city centre, in a walking distance from the main railway station, as well as a variety of hotels of various standards.
Poznań is a dynamic economic, academic, scientifical and cultural centre. Thanks to its excellent economic performance and International Fair the city is often called the economic capital of Poland. It is an excellent place for organizing conferences because it is also a centre of academic life. There are 22 universities and other institutions of higher education with over 120 000 students. Among the universities there is Poznań University of Technology one of the biggest and most recognized technical universities in Poland. Thanks to such a considerable number of students the city has got a creative and unforgettable atmosphere. An abundance of monuments and interesting places from all époques creates pleasant surroundings for social meetings after conference sessions.

www.eusipco2007.org

Chair of organizing committee
Jenny Benois-Pineau
University of Bordeaux/LABRI
jenny.benois@labri.fr

Technical Program Chair
Eric Pauwels
Centrum voor Wiskunde en Informatica
Eric.Pauwels@cwi.nl

Organizing Committee
Régine André-Obrecht (IRIT)
Patrick Bouthemy (IRISA)
Chabane Djeraba (IRIN)
Philippe Joly (IRIT)
Riccardo Leonardi (U. Brescia)
Bernard Merialdo (Eurecom)
Heri Nicolas (LABRI)
Georges Quénot (CLIPS-IMAG)

Local Committee
Publicity: *V. Bogati*
Publications : *J. Pinquier*
Webmaster : *D. Auber*
Cultural program : *J. –Ph. Domenger*
Financial management : *J.-L. Lassartesses*

Special issue
Ebroul Izquierdo (QMUL)

Program Committee
Régine André-Obrecht (IRIT)
Yannis Avritis (NTUA)
Michel Barlaud (I3S)
Jenny Benois-Pineau (LABRI)
Catherine Berrut (CLIPS-IMAG)
Alberto del Bimbo (U Florence)
Nozha Boujema (INRIA)
Patrick Bouthemy (IRISA/INRIA)
Matthieu Cord (LIP6)
Edward Delp (Purdue University)
Marcin Detyniecki (LIP6)
Chabane Djeraba (LIFL)
Moncef Gabbouj (TUT)
Patrick Gros (IRISA)
Alan Hanjalic (TU Delft)
Benoit Huet (Eurecom)
Ebroul Izquierdo (QMUL)
Philippe Joly (IRIT)
Stefanos Kollias (NTUA)
Yiannis Kompatsiaris (ITI)
Riccardo Leonardi (U Brescia)
Stepahne Marchand-Maillet (UniGe)
Ferran Marques (UPC)
José María Martínez (UAM)
Bernard Merialdo (EURECOM)
Jan Nesvadba (Philips)
Andreas Nuernberger (U Magdebourg)
Francois Pachet (Sony)
Fernando Pereira (IST)
Georges Quénot (CLIPS-IMAG)
Andreas Rauber (TU Wien)
Gael Richard (ENST)
Philippe Salembier (UPC)
Henri Sanson (FT R&D)
Steffen Staab (U Koblenz)
Alan Smeaton (DCU)
Shin'Ichi Satoh (NII)
Murat Tekalp (KU)

IEEE

Fifth International Workshop on

Content-Based Multimedia Indexing, CBMI-2007

June 25-27, 2007
Bordeaux, France

EURASIP

CBMI 2007 CALL FOR PAPERS

Following the four successful previous events of CBMI (Toulouse 1999, Brescia 2001, Rennes 2003 and Riga 2005), the LABRI/ *University of Bordeaux* will organize the next CBMI event. CBMI'07 aims at bringing together the various communities involved in the different aspects of Content-Based Multimedia Indexing. The scientific program of CBMI'07 will include the presentation of invited plenary talks, special sessions as well as regular sessions with contributed research papers.

Authors are encouraged to submit extended papers to the Special Issue of Signal Processing: Image Communication journal, EURASIP on CBMI. Topics of interest for submissions include, but are not limited to:

- Multimedia indexing and retrieval (image, audio, video, text)
- Multimedia content extraction
- Matching and similarity search
- Construction of high level indices
- Multi-modal and cross-modal indexing
- Content-based search techniques
- Multimedia data mining
- Presentation tools
- Meta-data compression and transformation
- Handling of very large scale multimedia database
- Organisation, summarisation and browsing of multimedia documents
- Applications
- Evaluation and metrics

PAPER SUBMISSION

Prospective authors are invited to submit full papers of not more than eight (8) pages including results, figures and references. Papers will be accepted only by electronic submission through the conference web site: http://cbmi07.labri.fr/. Style files (Latex and Word) are provided for the convenience of the authors.

Submission of full paper (to be received by):	**January 25, 2007**
Notification of acceptance:	**March 10, 2007**
Submission of camera-ready papers:	**April 10, 2007**

WORKSHOP VENUE

CBMI'07 will be held in Bordeaux (France) on June 25-27, 2007

For further information: http://cbmi07.labri.fr/

MUSCLE
Multimedia Understanding through Semantics, Computation and Learning

CNRS
CENTRE NATIONAL DE LA RECHERCHE SCIENTIFIQUE

COST 292
Semantic Multimodal Analysis of Digital Media

INRIA

15th International Conference on Digital Signal Processing

July 1-4, 2007
Cardiff
Wales, UK

Call for Papers

The 15th International Conference on Digital Signal Processing (DSP 2007), the most longstanding conference in the area of DSP, organised in cooperation with the IEEE, will be held in Cardiff the capital of Wales, UK, July 1-4, 2007. DSP 2007 belongs to a series of events which had its genesis in London in 1968 and continued to Florence, Nicosia, Lemessos, and Santorini. The last meeting took place overlooking the cauldron in the bay of Fira, Santorini, in 2002. This now tranquil location was once the scene of a massive eruption which led directly to the extinction of one of Europe's oldest civilisations, the Minoans; in 2007 delegates will be brought right up to date in an area of rebirth, Cardiff Bay, the heart of Europe's youngest capital. The conference will contain a number of Special sessions organised by internationally recognised experts. The programme will also include presentations of new results in lecture and poster sessions, together with plenary sessions delivered by eminent scientists. Accepted papers will appear in IEEE Xplore.

Special Sessions

TBA

Indicative Topics of Interest

Adaptive signal processing
Array processing, radar and sonar
Biomedical signal and image processing
Bioinformatics and genomic signal processing
Blind equalization
Blind source separation
Collaborative networking
Computer vision and pattern recognition
Data fusion
Design and implementation of signal processing systems
Detection and estimation theory
Distributed Signal Processing
Image and multidimensional signal processing
Information forensics and security
Joint source-channel coding
Machine learning for signal processing
Multimedia signal processing
Multimudal signal processing
Multivariate statistical analysis
Musical signal processing
Nonlinear signal processing
Progressive data transmission
Sensor array and multichannel systems
Speech and language processing
Time-frequency and time-scale analysis

Expected dates (to be confirmed):

Electronic paper submission	February 19, 2007
Acceptance notification	April 2, 2007
Camera-ready papers	April 9, 2007
Conference	July 1-4, 2007

www.cardiff.ac.uk/dsp2007/

DSP2007

Honorary Chair
A. G. Constantinides
Imperial College, UK

Technical Chair
J. A. Chambers
Cardiff University, UK

Organising Chair
S. Sanei
Cardiff University, UK

Publications Chair
J. McWhirter
QinetiQ, UK

Publicity Chair
Y. A. Hicks
Cardiff University, UK

Organising Committee
S. Lambotharan, UK
M. Al-Akaidi, UK
D. Marshall, UK

International Technical Committee
V. Cappellini, Italy
A. Cichocki, Japan
C. F. N. Cowan, UK
Z. Cvetkovic, UK
M. Davies, UK
A. Gershman, Germany
G. Giannakis, USA
F. Harris, USA
C. Jutten, France
R. Leyman, Singapore
M.D. Macleod, UK
D. Mandic, UK
S. K. Mitra, USA
M. Petrou, UK
I. Pitas, Greece
V. Poor, USA
I.K. Proudler, UK
T. Reed, USA
A. Sayed, USA
A. N. Skodras, Greece
A. Zoubir, Germany