Combinatorial Optimization in Mapping Generalized Template Matching onto Reconfigurable Computers

Xuejun Liang
Department of Computer Science, Jackson State University, Jackson, MS, USA 39217

Qutaibah Malluhi
Computer Science and Engineering Department, Qatar University, Doha, Qatar
Generalized Template Matching (GTM)

- Template Matching
- Two-D Digital Filtering
- Morphologic Operation
- Motion Estimation

For image region $i \leftarrow 1$ to $r$
For template $j \leftarrow 1$ to $m(i)$
For (all) pixel $P$ in $IR_i$
Basic-Function($P, T_{i,j}$)

Pixel level parallelism
Template level parallelism
Target FPGA Board Architecture
GT M Mapping Procedure

GTM Application Specification

<table>
<thead>
<tr>
<th>Image Regions</th>
<th>Templates</th>
</tr>
</thead>
<tbody>
<tr>
<td>IR₁</td>
<td>{T₁,₁, T₁,₂}</td>
</tr>
<tr>
<td>IR₂</td>
<td>{T₂,₁, T₂,₂, T₂,₃}</td>
</tr>
<tr>
<td>IR₃</td>
<td>{T₃,₁, T₃,₂}</td>
</tr>
</tbody>
</table>

BF DFG (Loop Body)

Component Library
Adder, Multiplier,…

FPGA Board

M₁₁, M₁₂, M₂₁, M₂₂

FPGA1, FPGA2
Phase 1: Enumerate FPGA buffers and FPGA unit function Designs

1. <Diagram 1>

2. <Diagram 2>

3. <Diagram 3>

4. <Diagram 4>

5. <Diagram 5>

...
Phase 2: Select UF designs and FPGA buffers to compose RF designs. Select a group of RFs, bind them to FPGA chips and memory ports, and partition the workload among them so as to minimize the computation time.

Each RF uses different memory ports and works on different image regions.
A: P1: MAP Enumeration and Pruning

B: P1: Buffer Generation and Area Estimation

C: P1: UF Mapping and Area Estimation

D: P2: RF Mapping and Binding

E: Code Generation

To minimize the computation time by
1. Selecting a group of RFs
2. Binding them to FPGA chips & memory ports
3. Partitioning the workload among them
To minimize
\[ \max \{ \text{Time}(RF_{i,j}) \mid 1 \leq i \leq N_{\text{FPGA}}, \text{and} 1 \leq j \leq q(i) \} \]
subject to
\[
\begin{cases}
\sum_{1 \leq j \leq q(i)} \text{Area}(RF_{i,j}) \leq S_{\text{FPGA}}, & 1 \leq i \leq N_{\text{FPGA}} \\
\sum_{1 \leq j \leq q(i)} \text{Port}(RF_{i,j}) \leq N_{\text{MP}}, & 1 \leq i \leq N_{\text{FPGA}} \\
N_{\text{FPGA}} q(i) \sum_{i=1}^{r} \sum_{j=1}^{q(i)} \text{WL}(RF_{i,j}) = \text{workload} \\
\text{workload} = \sum_{s=1}^{r} d_s \times r_s
\end{cases}
\]
\[ \text{Time}(RF_{i,j}) = S(RF_{i,j}) \times \text{WL}(RF_{i,j}) \]
\[ S(RF) = \left( \frac{II}{PF} \right) \times CP \times C_{\text{IMG}} \]
RF Mapping and Binding (Cont.)

Reduce to one FPGA chip case by partitioning the workload evenly among FPGA chips

To minimize

\[
\max\ \{\text{Time}(RF_j) \mid 1 \leq j \leq q\}
\]

subject to

\[
\begin{align*}
\sum_{1 \leq j \leq q} \text{Area}(RF_j) & \leq S_{FPGA} \\
\sum_{1 \leq j \leq q} \text{Port}(RF_j) & \leq N_{MP} \\
\text{Port}(RF_j) & \geq 1, 1 \leq j \leq q \\
\sum_{j=1}^{q} \text{WL}(RF_j) & = \text{workload}
\end{align*}
\]
RF Mapping and Binding (Cont.)

Further reduce by partitioning the workload “evenly” among RFs

To maximize

$$\sum_{1 \leq j \leq q} \frac{1}{S(RF_j)}$$

subject to

$$\sum_{1 \leq j \leq q} \text{Area}(RF_j) \leq S_{FPGA}$$

$$\sum_{1 \leq j \leq q} \text{Port}(RF_j) \leq N_{MP}$$

$$WL(j) = \frac{\text{workload}}{S(RF_j) \times (\sum_{1 \leq j \leq q} 1/S(RF_j))}$$
RF Selection and Port Binding

To maximize
\[ \sum_{1 \leq j \leq q} \frac{1}{S(RF_j)} \]
subject to
\[ \begin{align*}
\sum_{1 \leq j \leq q} Area(RF_j) & \leq S_{FPGA} \\
\sum_{1 \leq j \leq q} Port(RF_j) & \leq N_{MP}
\end{align*} \]

Problem Size
\[ N^1 + N^2 + \ldots + N^{N_{MP}} = (N^{N_{MP}} - 1) \times N / (N - 1) \]

N is number of RF designs

For Naïve Approach

Assume \( N_{MP} = 4 \), \( W_{PORT} = 32 \), \( B_{DATA} = 8 \), \( N_{MW} = 1 \), \( \alpha_W = 1.0 \), \( R_{WIN} = 9 \), and \( N_{AP} = 20 \). Then \( N = 114 \) and \( \Rightarrow 1.7 \times 10^8 \)

1. Naïve Approach
2. Further reduce to two sub-problems
   - Generalized Integer Partition Problem
   - Generalized Knapsack Problem
RF Selection and Port Binding (Cont.)

Generalized Integer Partition

\[ i_1 + i_2 + ... + i_q \leq N_{MP} \]
\[ 1 \leq i_1 \leq i_2 \leq ... \leq i_q \]
\[ 1 \leq q \]
\[ i_j = Port(RF_{ij}) \]

Generalized Knapsack Problem

To maximize
\[ \sum_{1 \leq j \leq q} 1/S(RF_{ij}) \]
subject to
\[ \sum_{1 \leq j \leq q} \text{Area}(RF_{ij}) \leq S_{FPGA} \]

1. Solving GIP
2. For each solution of GIP, solving GKP
   A. Exhaustive Search, or
   B. Multi-Dimensional Binary Search (MDBS)
The solution to the problem is a set of $q$-vectors denoted by $S(q,n)$. For example, if $n=8$ and $q=4$, then

$$S(q,n) = \{(1,1,1,5),(1,1,2,4),(1,1,3,3),(1,2,2,3),(2,2,2,2)\}$$

Define the an operator: $\Theta$: $I \times (q$-vectors)$ \mapsto (q+1)$-vectors as

$$a \Theta (a_1, a_2, \ldots, a_q) = (a, a_1 + a - 1, a_2 + a - 1, \ldots, a_q + a - 1)$$

Then $S(q,n)$ can be computed by the following recursive equation.

$$S(q,n) = \begin{cases} 
(n) & \text{if } k = 1 \\
(1,1,\ldots,1) & \text{if } k = n \\
\bigcup_{b=1}^{k} b\Theta S(q-1,(n-b)-(b-1)*(q-1)) & \text{otherwise}
\end{cases}$$
Integer Partition Problem (Cont.)

A Dynamic Programming Algorithm to Compute $S(q,m)$ ($1 \leq q \leq m \leq n$)

For $i = 1$ to $n$
$S(i, i) = (1,1,...,1)$

For $j = 2$ to $n$
$S(1, j) = (j)$

For $i = 2$ to $n - 1$
For $j = i + 1$ to $n$
$S(i, j) = \Phi$

For $k = 1$ to $\left\lfloor \frac{j}{i} \right\rfloor$
$S(i, j) = S(i, j) \cup k \Theta S(i - 1, (j - k) - (k - 1)x(i - 1))$

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>n-1</th>
<th>n</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>S(1,1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>S(1,2)</td>
<td>S(2,2)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>n-1</td>
<td>S(1,n-1)</td>
<td>S(2,n-1)</td>
<td>S(n-1,n-1)</td>
<td></td>
</tr>
<tr>
<td>n</td>
<td>S(1,n)</td>
<td>S(2,n)</td>
<td>S(n-1,n)</td>
<td>S(n,n)</td>
</tr>
</tbody>
</table>
MDBS: Preprocessing

**Grouping:** Each $RF_{i,j}$ in $Cad(i)$ uses $i$ memory ports

$$Cad(i) = \{RF_{i,j} \mid j = 1,2,...,m(i)\}, \ i = 1,2,...N_{MP}$$

**Sorting:**

$$S(RF_{i,1}) < S(RF_{i,2}) < ... < S(RF_{i,m(i)}), \ i = 1,2,...N_{MP}$$

**Assumption:**

$$Area(RF_{i,1}) > Area(RF_{i,2}) > ... > Area(RF_{i,m(i)}), \ i = 1,2,...N_{MP}$$
MDBS: Algorithm

For a solution to \((i_1, i_2, \ldots, i_q)\) GIP, compute

- sum of the largest RF areas \((SLA)\)
  \[
  SLA = \sum_{1 \leq j \leq q} \text{Area}(RF_{i_j,1})
  \]

- sum of the smallest RF areas \((SSA)\)
  \[
  SSA = \sum_{1 \leq j \leq q} \text{Area}(RF_{i_j,m(j)})
  \]

If \(SLA < S_{FPGA}\), \(\{RF_{i_j,1} \mid 1 \leq j \leq q\}\) is a feasible solution.

If \(SSA > S_{FPGA}\), there is no solution.

Otherwise, compute sum of the median RF areas \((SMA)\)

\[
SMA = \sum_{1 \leq j \leq q} \text{Area}(RF_{i_j,m(j)/2})
\]
If $SMA > S_{FPGA}$, then $Cad(i_j) \ (1 \leq j \leq q)$ can be divided into two parts:

$Cad(i_j,0) = \{RF_{i_j,k} \mid k = 1,2,\ldots,m(i) / 2\}$ \quad $Cad(i_j,1) = \{RF_{i_j,k} \mid k = m(i) / 2 + 1,\ldots,m(i)\}$

Picking $q$ RF from these $2 \times q$ sets gives $2^q$ combinations. Then the searching problem can be broken up as $2^q-1$ sub-problems with these combinations except one combination $Cad(i_1,0),\ldots, Cad(i_q,0)$.

If $SMA \leq S_{FPGA}$, then $Cad(i_j) \ (1 \leq j \leq q)$ can be divided into two parts:

$Cad(i_j,0) = \{RF_{i_j,k} \mid k = 1,2,\ldots,m(i) / 2 - 1\}$ \quad $Cad(i_j,1) = \{RF_{i_j,k} \mid k = m(i) / 2,\ldots,m(i)\}$

Picking $q$ RF from these $2 \times q$ sets gives $2^q$ combinations. Then the searching problem can be broken up as $2^q-1$ sub-problems with these combinations except one combination $Cad(i_1,1),\ldots, Cad(i_q,1)$.
MDBS: Example

Assume $S_{FPGA}=37$ and (1, 2) is a solution to GIP. Assume there are 6 RFs in Cad(1) and 5 RFs in Cad(2). Their areas are listed in the following table.

<table>
<thead>
<tr>
<th>j</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Area(RF_{1,j})</td>
<td>24</td>
<td>20</td>
<td>16</td>
<td>14</td>
<td>13</td>
<td>12</td>
</tr>
<tr>
<td>Area(RF_{2,j})</td>
<td>35</td>
<td>29</td>
<td>26</td>
<td>22</td>
<td>18</td>
<td>NA</td>
</tr>
</tbody>
</table>

$SLA = 24+35 = 59 > S_{FPGA}$ and $SSA=12+18 = 30 < S_{FPGA}$. As $SMA = 16+26 = 42 > S_{FPGA}$, three sub-problems are needed to consider.
**MDBS: Example (Cont.)**

For sub-problem (1), as $SLA = 14+22 = 36 < S_{FPGA}$, $(RF_{1,4}, RF_{2,4})$ is a feasible solution.

For sub-problem (2), as $SSA = 12+26 = 38 > S_{FPGA}$, there is no solution.

For sub-problem (3), $SLA = 46$ and $SSA = 34$. As $SMA = 42 > S_{FPGA}$, it splits into three sub-problems again.

<table>
<thead>
<tr>
<th>Sub(1): j</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Area(RF$_{1,j}$)</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>14</td>
<td>13</td>
<td>12</td>
</tr>
<tr>
<td>Area(RF$_{2,j}$)</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>22</td>
<td>18</td>
<td>NA</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sub(2): j</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Area(RF$_{1,j}$)</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>14</td>
<td>13</td>
<td>12</td>
</tr>
<tr>
<td>Area(RF$_{2,j}$)</td>
<td>35</td>
<td>29</td>
<td>26</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sub(3): j</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Area(RF$_{1,j}$)</td>
<td>24</td>
<td>20</td>
<td>16</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>Area(RF$_{2,j}$)</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>22</td>
<td>18</td>
<td>NA</td>
</tr>
</tbody>
</table>
Sub-problem (3-1) provides another feasible solution (RF\(_{1,3}\), RF\(_{2,5}\)),

Sub-problems (3-2) has no solution.

Sub-problems (3-3) has no solution.

The final solution for (1, 2) then will be one of the two feasible solutions that has maximum sum of speeds.
# Efficiency of MDBS

<table>
<thead>
<tr>
<th>Complexity Assume $m(i) = m$, $N_{MP} = n$</th>
<th>Example: $N_{MP} = 4$, $W_{PORT} = 32$, $B_{DATA} = 8$, $N_{MW} = 1$, $\alpha_W = 1.0$, $R_{WIN} = 3$ and $N_{AP} = 9$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Naïve $m^n \times n^n$</td>
<td>2,880,952 148.05</td>
</tr>
<tr>
<td>Exhaustive Search $m^n$</td>
<td>9,982 0.526</td>
</tr>
<tr>
<td>Multi-Dimensional Binary Search</td>
<td>1,138 0.1288</td>
</tr>
<tr>
<td></td>
<td>$\begin{aligned} \theta(m^{\log_2(2^n - 1)}) &amp; \text{ if } n &gt; 1 \ \theta(\log_2 m) &amp; \text{ if } n = 1 \end{aligned}$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Comparison</th>
<th>Space</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Naïve / Exhaustive Search</td>
<td>$2880952/9982 = 288$</td>
<td>$148.05/0.526 = 281$</td>
</tr>
<tr>
<td>Exhaustive Search / MDBS</td>
<td>$9982/1138 = 9$</td>
<td>$0.526/0.1288 = 3$</td>
</tr>
</tbody>
</table>
Series 1: Multi-Dimensional Binary Search
Series 2: Exhaustive Search