**Motivation:** Accelerate image processing tasks through efficient use of FPGAs. Combine already designed components at runtime to implement series of transformations (pipelines)

**Fast, Flexible Image Processing**
- Run this pipeline:
  - Median Filter → Image size of 40185 pixels → Histogram

**On this Environment:**
- Which component implementations to use?
- How to minimize overall latency?
- When to use FPGA?
- How to change the pipeline or interfaces dynamically?

**Goal:** If pipeline selection is left to the image analyst, can the other three steps be performed automatically at runtime?

**Four Shortcomings in Codesign**
- Using reconfigurable hardware incurs execution costs not present in software or ASIC-based systems
  - Hardware initialization
  - Communication
  - Reprogramming

**Efficient Use of FPGAs**
- Software algorithm’s runtime for small images less than the hardware costs
  - Profiling the hardware and software runtimes for different image sizes determines the crossover point
  - Deciding at runtime to execute in software or hardware is simple for one algorithm processing one image

**Image Processing Pipelines**
- Series of image processing algorithms applied to an image
  - Each algorithm has a software and hardware implementation
  - Finding the optimal pipeline assignment is complicated
    - Exponential number of implementations
    - Coupling costs differ for each pipeline assignment
  - Need a strategy to find a fast pipeline implementation at runtime

**Our Codesign Environment**
- **Pipeline Selection:** choosing and ordering components
- **Pipeline Assignment:** assigning pipelines to minimize overall latency with the efficient use of software and FPGA
- **Pipeline Compilation:** creating image processing pipelines dynamically
- **Pipeline Execution:** executing image processing pipelines dynamically

**Pipeline Assignment**
- Chooses an algorithm to solve PA based on pipeline size

**Pipeline Compilation**
- Builds executable pipeline from PA solution

**Pipeline Execution**
- Executes the pipelines

**Pipeline Assignment**
- Choose algorithm based on pipeline size

**Runtime Interfacing for Pipeline Synthesis**
- Connects the appropriate implementations so that the coupling costs are satisfied

**Four Challenges to Codesign**
- Unify implementation languages
- Partitioning design
- Interfacing hardware and software
- Abstract communication layer and runtime interface
- Choosing a target architecture

**Reconfigurable Systems**
- Applications are configured statically
- Design is not sensitive to user changes
- FPGA-based tools do not account for overhead costs
- Latency is underestimated
- Partition bound too early
- Interface changes too costly
- System code needs extensive rewrites

**SW/HW Runtime Procedural Partitioning Tool**
- Solves PA within either fixed or adaptive time limit based on user’s choice

**Optimization Method**
- Fixed
- Adaptive

**Dynamic Programming**
- 1-15
- 10-20
- 16-20

**Random Pipeline Test**
- Forty test pipelines of different lengths were run in the Dynamo system for the best latency solution
- Image size of 40185 pixels
- Average ARE: 23% with overhead, 70% without

**Future Work**
- Extend the pipeline assignment problem for FPGA devices
- in a network of workstations
- with embedded processors
- Extend the pipeline assignment problem’s objectives to include power minimization
- Extend the latency model to include an estimation of the error for better accuracy

**Publications**

**A Two Component Pipeline**
- **Median Filter**
- Image size of 40185 pixels

**Comparison of Hardware and Software Pipeline Performances**

**Packet Exchange Platform**
- **Runtime Environment’s Communication Agent**
- **Pipeline’s Communication Agent**

**Dynamo: A Runtime Codesign Environment**
- Dr. Miriam Leeser
- Dr. Laurie Smith King
- Heather Quinn