Intel® Integrated Performance Primitives (Intel® IPP)

Shinn Lee
IPP Chief Architect
OpenCV Syposium, Beijing, June 2007
What Is A Primitive?

• A low level building block that
  – Abstracts low-level implementation details
  – Performs a single operation
  – Is a component piece of a larger solution
What Are IPP?

- Huge collection of useful multimedia functions
  - Mathematics, signal, speech, audio, video, image, graphics
- Low-level and simplex
- Highly-optimized, processor-specific code
- Common across multiple platforms
  - desktop, and server processors
  - Windows*, Linux operation systems
- Widely used, well-received product
Intel® Integrated Performance Primitives (Intel® IPP) — Overview and Benefits

Free Code Samples

Cross-platform Application Programming Interface (API)

Processor-Optimized Implementation

Application Source Code

Intel IPP Usage Code Samples
- Sample video/audio/speech codecs
- Image processing and JPEG
- Signal processing
- Data compression
- .NET and Java integration

API calls

Intel IPP Library C/C++ API
- Cryptography
- Image processing
- Image color conversion
- JPEG / JPEG2000
- Computer Vision
- Video coding
- Audio coding
- Data Compression
- Signal processing
- Matrix mathematics
- Vector mathematics
- String processing
- Speech coding
- Speech recognition

Static/Dynamic Link

Intel IPP Processor-Optimized Binaries
- Intel® Core™ 2 Family Processors
- Intel® Core™ Duo and Core™ Solo Processors
- Intel® Pentium® D Dual-Core Processors
- Intel® Xeon® 64-bit Dual-Core Processors
- Intel® Pentium® M and Pentium® 4 Processors
- Intel® Itanium® 64-bit Processor Family
- Intel® Xeon® DP and MP Processors
- Intel® XScale™ Technology-based Processors

Benefits

Rapid Application Development

Cross-platform Compatibility and Code Re-Use

Outstanding Performance

• Data Compression
• Signal processing
• String processing
• Speech coding
• Speech recognition

Copyright © 2006, Intel Corporation. All rights reserved.
* Other names and brands may be claimed as the property of others. Intel may make changes to specifications, product descriptions and plans at any time, without notice.
The 15 Domains in IPP

1. **Static Pictures**
   - Image Processing
   - Computer Vision
   - JPEG (Image coding)
   - Ray-Tracing/Rendering*

2. **Motion Pictures**
   - Video Coding
   - Color Conversion

3. **Sound and Speech**
   - Audio Coding
   - Speech Coding
   - Speech Recognition

4. **Computer Data**
   - Data Compression
   - Cryptography
   - Text string processing

5. **Mathematics**
   - Digital Signal Processing
   - Small matrices
   - Vectors

* new in Intel IPP 5.2 (Q2-2007)
## IPP 5.x Domains and Functions

<table>
<thead>
<tr>
<th>Domain</th>
<th>IA</th>
<th>IXP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image Processing</td>
<td>ippi 2570</td>
<td>ippi 1574</td>
</tr>
<tr>
<td>Signal Processing</td>
<td>ipps 1865</td>
<td>ipps 783</td>
</tr>
<tr>
<td>Speech Recognition</td>
<td>ippsr 618</td>
<td>ippsr 618</td>
</tr>
<tr>
<td>Computer Vision</td>
<td>ippcv 417</td>
<td>ippcv 417</td>
</tr>
<tr>
<td>Color Conversion</td>
<td>ippcc 410</td>
<td>ippcc 410</td>
</tr>
<tr>
<td>Cryptography</td>
<td>ippcp 307</td>
<td>ippcp 307</td>
</tr>
<tr>
<td>Speech Coding</td>
<td>ippsc 297</td>
<td>ippsc 247</td>
</tr>
<tr>
<td>Video Coding</td>
<td>ippvc 219</td>
<td>ippvc 219</td>
</tr>
<tr>
<td>Image Coding</td>
<td>ippj 201</td>
<td>ippj 201</td>
</tr>
<tr>
<td>Audio Coding</td>
<td>ippac 152</td>
<td>ippac 152</td>
</tr>
<tr>
<td>Data Compression</td>
<td>ippdc 73</td>
<td>ippdc 73</td>
</tr>
<tr>
<td>Char Processing</td>
<td>ippch 72</td>
<td>ippch 72</td>
</tr>
<tr>
<td>Small Matrix</td>
<td>ippm 669</td>
<td>ippm 0</td>
</tr>
<tr>
<td>Vector Math</td>
<td>ippvm 136</td>
<td>ippvm 0</td>
</tr>
</tbody>
</table>

**Totals:**
- **IA:** 8006
- **IXP:** 5082

*Other names and brands may be claimed as the property of others. Intel may make changes to specifications, product descriptions and plans at any time, without notice.*
Intel IPP: Codec and Data Processing Standards Support

- **Video Codecs**
  - H.264
  - H.263
  - H.261
  - MPEG-4
  - MPEG-2
  - Motion JPEG
  - DV

- **Audio Codecs**
  - MP3
  - AAC
  - AC3

- **Image Codecs**
  - JPEG
  - JPEG2000

- **Speech Codecs**
  - AMR-WB
  - G.711 / I / II
  - G.722.1 G.722.2 (GSM-AMR)
  - G.723.1 / A
  - G.726 / A
  - G.728 G/I/H
  - GSM-FIR
  - GSM 06.90-06.94
  - GSM 06.10-06.12
  - GSM 06.31-06.32

- **Echo Cancellation**
  - G.168-2000
  - G.167

- **Speech Recognition**
  - Aurora
  - Advanced Aurora
  - Gaussian Mixture

- **Data Compression**
  - Huffman encoding/decoding
  - RLE encoding/decoding
  - MoveToFront (MTF)
  - Burrows-Wheeler Transformations (BWT)
  - General Interval Transform (GIT)
  - Lempel-Ziv-Storer-Szymanski (LZSS) functions

- **Cryptography**
  - Rijndael, DAARijndael
  - DES, DAA-DES
  - Triple DES, DAA-TDES
  - Twofish, DAA-Twofish
  - Blowfish, DAABlowfish
  - SHA1, SHA256/384/512, HMAC-SHA1
  - MD5, HMAC-MD5
  - Digital Signature Algorithm (DSA)
UMC: IPP based codec

• What is it?
  – A set of unified C++ interface for media building blocks:
  – Designed for portability, allow cross platform (uArch, OS) support

• Benefits of using UMC
  – Simplifies media application development by providing consistent interface to all codecs
  – Simplifies adding new codec support
  – Simplifies porting application to new platform and easy to integrate with DirectShow, GStreamer

• Introduced as sample in 5.0
  – Combine all media samples into one package
UMC Architecture

Application Layer

Decoding/encoding/transcoding applications, DShow filters, plug-ins

UMC Base Classes (Abstract Base Classes)

Audio Codec* Video Codec* Splitter

UMC Derivative Classes (Codec’s)

MP3 Encoder AAC Decoder AC3 Decoder H.264 Decoder MPEG4 Decoder DV Decoder AVI Splitter MPEG4 Splitter

OS-Dependent Renders

DirectX Render SDL Render File Reader Socket Reader

VM Library – OS Abstraction Layer

IPP Libraries

* illustration purpose only, more codec support in 50 gold.
## Codec Support in OS/Platform

<table>
<thead>
<tr>
<th>Codec components</th>
<th>IA-32</th>
<th>Intel® EM64T</th>
<th>Itanium® processors</th>
<th>Intel® IXP 4XX Product Line of Network Processors</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>L</td>
<td>W</td>
<td>L</td>
<td>W</td>
</tr>
<tr>
<td>AAC audio decoder (integer based)</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>AAC audio decoder (floating point based)</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>AC3 audio decoder</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>MP3 audio decoder (integer based)</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>MP3 audio decoder (float point based)</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>AAC audio encoder (integer based)</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>AAC audio encoder (floating point based)</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>MP3 audio encoder (integer based)</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>MP3 audio encoder (floating point based)</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
</tbody>
</table>

“●” - Validated as of IPP 5.0  
“○” - Not supported yet  
“N/ A” - Not available due to absence of corresponding APIs for this platform  
L - Linux*  
W - Windows*
Continued....

<table>
<thead>
<tr>
<th>Codec components</th>
<th>IA-32</th>
<th>Intel® EM64T</th>
<th>Itanium® processors</th>
<th>Intel® IXP 4XX Product Line of Network Processors</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>L</td>
<td>W</td>
<td>L</td>
<td>W</td>
</tr>
<tr>
<td>DV video decoder</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>MPEG 2 video decoder</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>MPEG 4 video decoder</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>MJPEG video decoder</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>H.261 video decoder</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>H.263 video decoder</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>H.264 video decoder</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>MPEG-2 video encoder</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>MPEG 4 video encoder</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>H.264 video encoder</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>H.261 video encoder</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>H.263 video encoder</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>AVI splitter</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>MP4 splitter</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>MPEG-2 PS/TS splitter</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
</tbody>
</table>

Start integrating UMC-based media codec for your development!

Copyright © 2006, Intel Corporation. All rights reserved. * Other names and brands may be claimed as the property of others. Intel may make changes to specifications, product descriptions and plans at any time, without notice.
**Intel® IPP Crypto is...**

- Variety of crypto functions – block ciphers, hash-functions, public-key cryptography

<table>
<thead>
<tr>
<th>Symmetric ciphers</th>
<th>DES, 3DES, Blowfish, TwoFish, Rijndael, RC5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stream ciphers</td>
<td>RC4</td>
</tr>
<tr>
<td>Hashing</td>
<td>SHA1, SHA256, SHA224, SHA512, SHA384, MD5, HMAC, DAA</td>
</tr>
<tr>
<td>Asymmetric ciphers and Schemes</td>
<td>RSA, RSAES-OAEP</td>
</tr>
<tr>
<td>Key agreement</td>
<td>DLSVDP-DH, ECSVDP-DH, ECSVDP-DHC</td>
</tr>
<tr>
<td>Digital Signature Generation/Verificati on and Schemes</td>
<td>DLSP-DSA, DLVP-SSA, ECSP-DSA, ECVP-DSA, ECSP-NR, ECVP-NR, RSASSA-PSS</td>
</tr>
<tr>
<td>Miscellaneous</td>
<td>Big Number, Finite Fields and Polynomial Arithmetic</td>
</tr>
</tbody>
</table>

**397 crypto functions in IPP 5.2 gold**
Threading Optimization in IPP Crypto Library

- Optimized for Intel® IA-32, Intel® x64, Itanium®-based platform and Intel XScale microarchitectures;
- Special tuning for multi-core Intel CPUs has been done wherever is possible

<table>
<thead>
<tr>
<th>algorithm/function</th>
<th>single-thread</th>
<th>multi-thread (2x2Duo 3GHz)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decrypt Rijndael-128-CBC</td>
<td>~20 cpe</td>
<td>~7 cpe</td>
</tr>
<tr>
<td>Decrypt Rijndael-192-CBC</td>
<td>~26 cpe</td>
<td>~7.5 cpe</td>
</tr>
<tr>
<td>Decrypt Rijndael-256-CBC</td>
<td>~28 cpe</td>
<td>~8 cpe</td>
</tr>
<tr>
<td>Decrypt RSA-1024</td>
<td>1380 op/s</td>
<td>2380 op/s</td>
</tr>
<tr>
<td>Decrypt RSA-2048</td>
<td>190 op/s</td>
<td>375 op/s</td>
</tr>
<tr>
<td>Decrypt RSA-4096</td>
<td>27 op/s</td>
<td>54 op/s</td>
</tr>
</tbody>
</table>

98 of 397 crypto functions are threaded in IPP
Directions for Future Development

- More Code Samples
- Examples
  - zlib, libzip2
  - Medical Imaging lib
  - jpeg, libjpeg
  - Clng, VSIP
  - Open Dynamics Eng
  - (game physics)
  - OpenSSL

- Core general algorithms
  - DCT / DFT / Wavelet transform
  - Entropy coding (Huffman, etc.)
  - Matrix/Vector operations
  - LZ77, LZSS
  - Burrows-Wheeler transform

- Standard models/formats
  - DICOM wrapper
  - JPEG, JPEG2000, PNG, JBIG
  - Perlin noise
  - SSL
  - H.264, MPEG-4, MPEG-2

- Domain Functions (APIs)
- Domain Functions (APIs)

- User Application customisations
- Application
- Middleware/SDKs
- OS plug-ins
- Operating System
- Drivers
- BIOS/Firmware

Maintain the Primitives
Grow the Higher-Level APIs

Intel IPP primitive library
Higher-Level APIs

Examples in the Market Today:

- **Data Compression:**
  - zlib, libbzip2
- **Medical/Document Imaging:**
  - Pegasus Imaging
  - LeadTools
  - Accusoft
  - Snowbound Software
  - libPNG
- **Encrypted communications:**
  - OpenSSL
- **Computer Vision**
  - OpenCV
- **Signal Processing**
  - VSIPL++

Proposed Strategy:

- Use Code Samples to enter higher-level API market
- “Promote” code samples into binary-only, new libraries
  - higher performance than code samples

Question:

- Will ISVs object to not having source code?
Summary

• IPP is a broad library for many market segments
  – 15 domains total, covering ~9000 functions
  – Opportunities in compressing/uncompressing data:
    – “human data”: images, video, audio, speech
    – “computer data”: lossless compression, encryption, text

• IPP 5.2 Gold released now
  – push on multi-core and 64-bit Core2
  – push on Enterprise data compression/transmission
  – push on Healthcare imaging

• Future IPP versions
  – shift to higher-level APIs
Reference and Contacts:

- **External:**
  - Intel® IPP Product Web Site
    (releases info, Features, Samples, performance data, licensing, etc)
  - Intel® IPP Support Web Site
    (FAQs, Top critical issues, etc)
  - Intel® Premier Support
    (Technical Support, software update, etc)
  - Intel® IPP Forum
    (Technical discussion with other developers)

- **Contacts:**
  - Technical Consulting Engineers:
    Ying Hu
  - Business Development Manager
    Wei Lin
  - Product Marketing Manager
    Siang-Chun The

- **H264 doc:**
  - ITU-T Standard
  - H264 short overview