Development Status of the Vector, Signal, and Image Processing Library (VSIPL)

M. Richards[1] ,D. Campbell[1] (presenter), R. Judd[2], J. Lebak[3], and R. Pancoast[4]

[1]Georgia Tech Research Institute, Atlanta, GA

[2]U.S. Navy SPAWAR Systems Center, San Diego, CA

[3]MIT Lincoln Laboratory, Lexington, MA

[4]Lockheed-Martin NESS, Moorestown, NJ

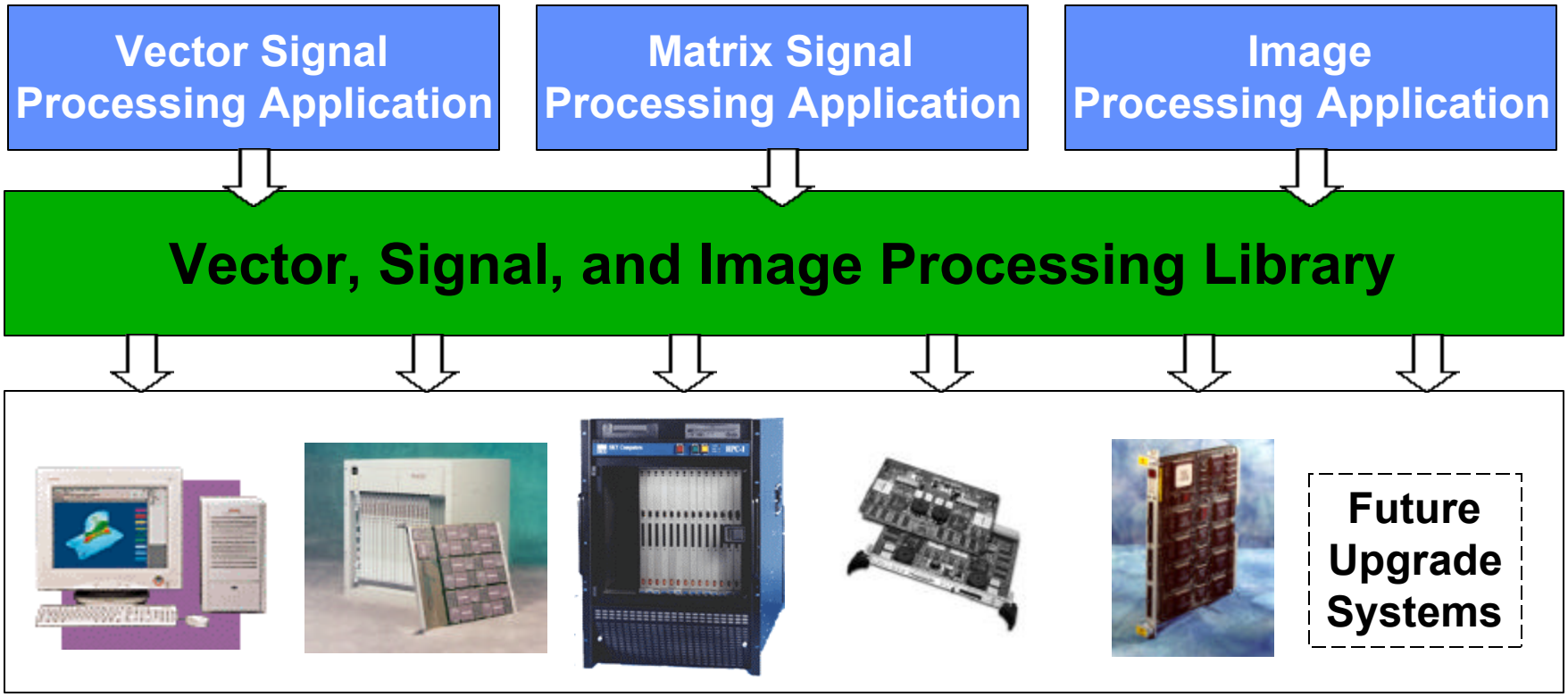
[1, 3] These authors sponsored by the U.S. Navy under GSA contract GS10TF02EBM0528 and Air Force Contract F19628-00-C-0002. Opinions, interpretations, conclusions, and recommendations are those of the authors and are not necessarily endorsed by the United States Air Force, General Services Administration, or United States Navy.

Acknowledgements

U.S. Navy PMS411 & TASP COE

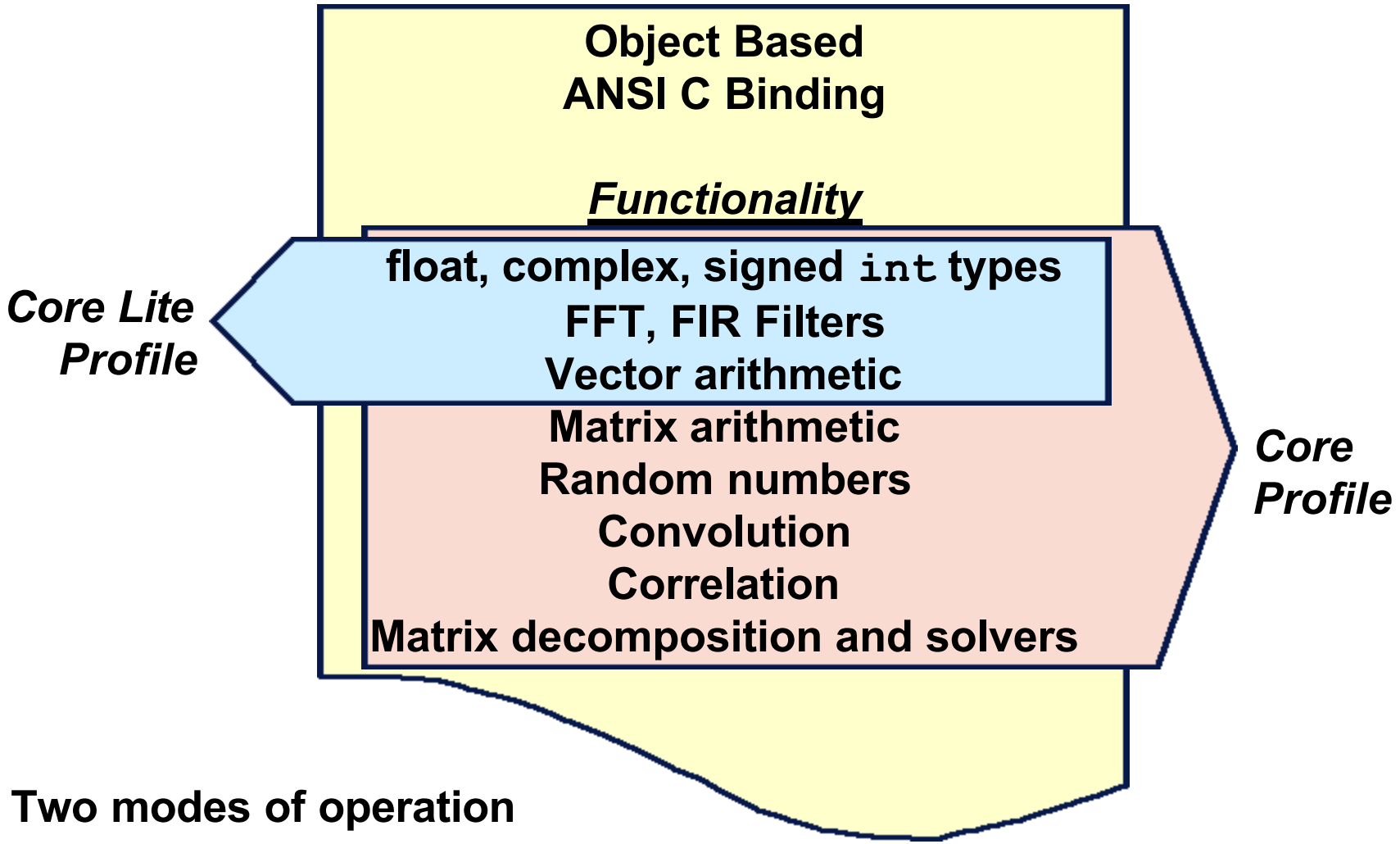
Defense Advanced Research Projects Agency

VSIPL Goals



- Portable to workstations, embedded systems, FPGAs with minimal performance cost
- Applicable to simple and complex applications
- Easier upgrade cycle
- Reduced development time and cost

VSIPL API Properties



- Two modes of operation
  - Development mode with extensive error checking
  - Performance mode with minimal error checking

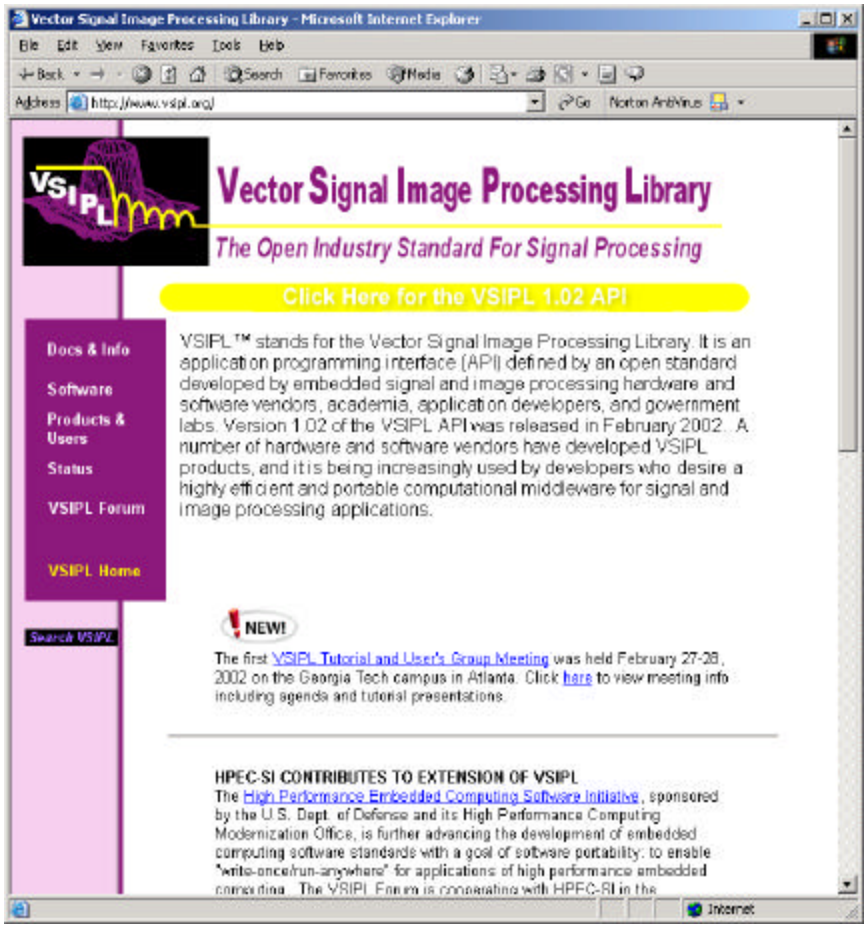
Current VSIPL Forum Products

- Standard API for Vector/Signal Processing
  - Version 1.02 released February 26, 2002
    - minor corrections and updates to VSIPL 1.01
  - Version 1.1 in final edit, expected 4Q 2002
- TASP VSIPL demonstration library
  - Developed by Randy Judd of USN SSC-SD
  - ANSI C production mode implementation
  - Core and Core lite profiles
  - "Core Plus" implementation including additional functionality
- Portable C Test Suite 1.03
  - Developed by Dan Campbell of GTRI
  - Tests compliance with Core Lite Profile of VSIPL 1.01 API
  - Does not test performance (speed or memory)

All may be downloaded from VSIPL web site
<http://www.vsipl.org>

Major Resources Available at vsipl.org

- VSIPL 1.02 API document
- Feb 2002 VSIPL Tutorial/User's Group presentations
- Supporting documents
  - VSIPL basic requirements
  - VSIPL profile definitions
  - and more …
- VSIPL Reference Implementation Software
  - three builds: Core Lite, Core, "Core Plus"
  - VSIPL Compliance Test Suite 1.03
- Links to VSIPL Product Vendors
- VSIPL Forum Information



http://www.vsipl.org

Changes in VSIPL 1.1

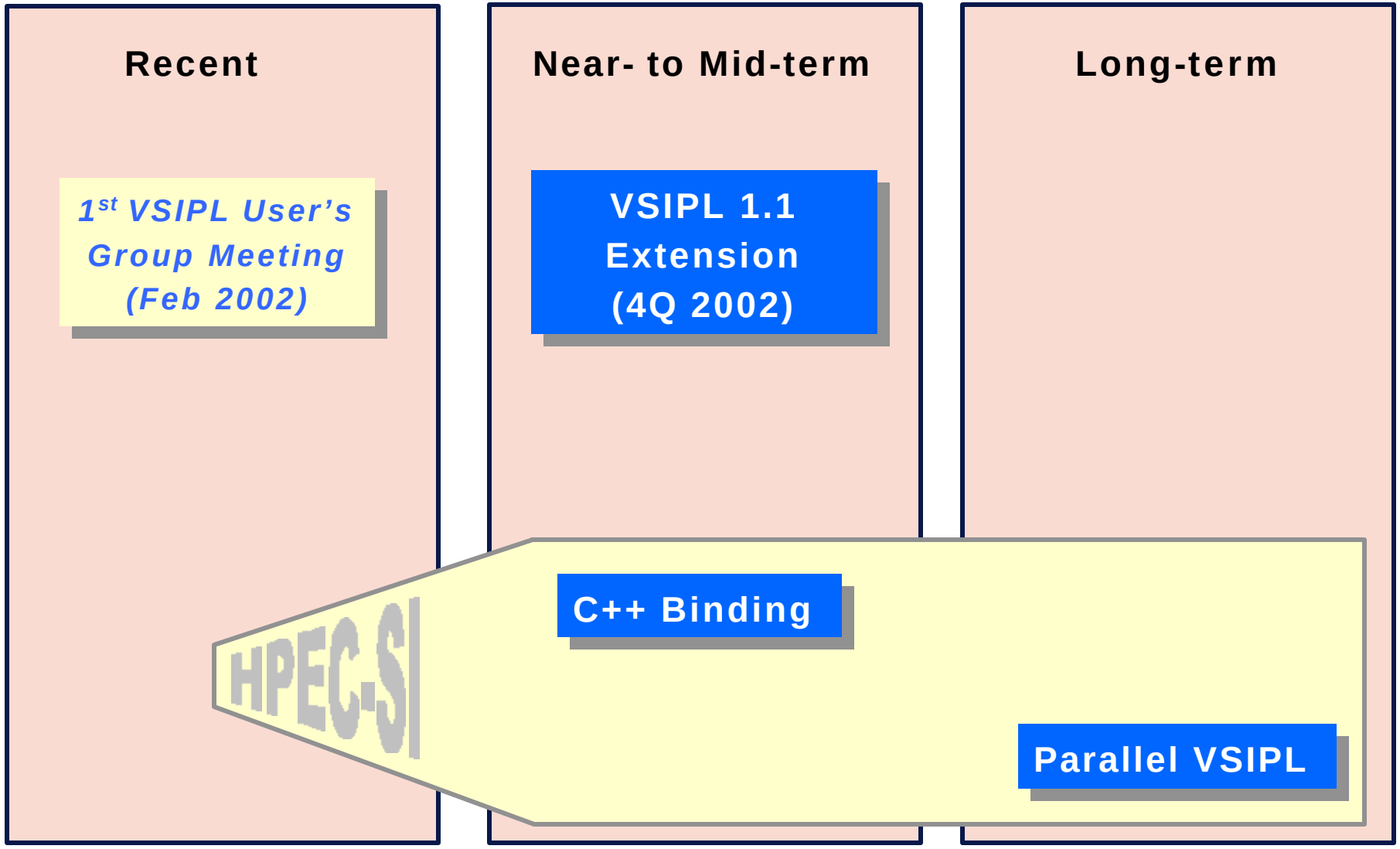
- **Correction of various errata**
- **New functions**
  - **Singular value decomposition,** $A = USV^H$
    - **includes functionality to extract subspaces corresponding to the highest or lowest singular values**
    - **supports pre- and post multiplication of a matrix by** U **or** V
  - **Windowed FFT**
    - **Defines window as part of the FFT object**
    - **Integrates data taper and FFT calculation**
  - **New I/O functions to**
    - **Operate on VSIPL vendor-dependent objects (*e.g.*, FFT, QR, LUD)**
      - **differ from block objects because the data associated with them is implementation-dependent**
    - **Allow objects to be communicated, saved to files, *etc.***

Current Commercial Implementations (Aug. 2002)

	Vendor	*Implementation**
CSPI	**CSPI**	**Core Lite** *also resell VSI-Pro for Core*
DNA COMPUTING SOLUTIONS	**DNA Computing Solutions** *supported under VxWorks and Linux*	**Core**
MCCI	**MCCI Autocoding Toolset**	**Core Lite**
Computer Systems, Inc. MERCURY	**Mercury Computer Systems**	**Core Lite + some 2D**
MPI Software Technology	**MPI Software Technology, Inc.** *Licensed by Thales, Radstone, Concurrent, CSPI; Supports G4 under VxWorks, LynxOS, Linux, MacOS, Linux PPC, and Windows*	**Core**
SKYCOMPUTERS	**Sky Computers**	**"Core Plus", multiple data types**
SYNERGY microsystems	**Synergy Microsystems**	**Core Lite**
TRANSTECH DSP	**Transtech DSP**	**Core**

** Most vendors also accommodate specific customer requirements*

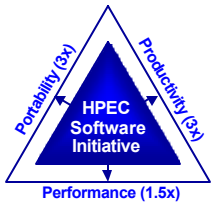
VSIPL Activities



High Performance Embedded Computing Software Initiative

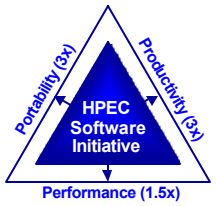
- HPEC-SI seeks to bridge the gap between high level tools and embedded hardware by building and extending on existing open standards such as VSIPL, MPI, DRI, *etc.*
- HPEC-SI extensions will extend VSIPL into embedded niches not currently addressed
  - C++ binding
  - parallel data distribution and computational algorithms

Figure courtesy of MIT Lincoln Laboratory



VSIPL++

- C++ binding offers benefits over C binding
  - much more compact code
  - drastic reduction in number of function prototypes
  - enables use of template and generic programming techniques to gain performance improvements similar to early binding
- HPEC-SI program serving as forum for defining VSIPL++ concepts and 0.1 spec
  - CodeSourcery implementing detailed specification and reference library
- Goal is to have a 0.1 draft specification and prototype software in Fall 2002



Parallel VSIPL

- Standard VSIPL machine model is a single threaded uniprocessor
- Efficient parallel algorithms require
  - coordinated data distribution and parallel algorithms strategies
  - user control of data distribution
  - scalability of algorithm to different machine sizes and layouts

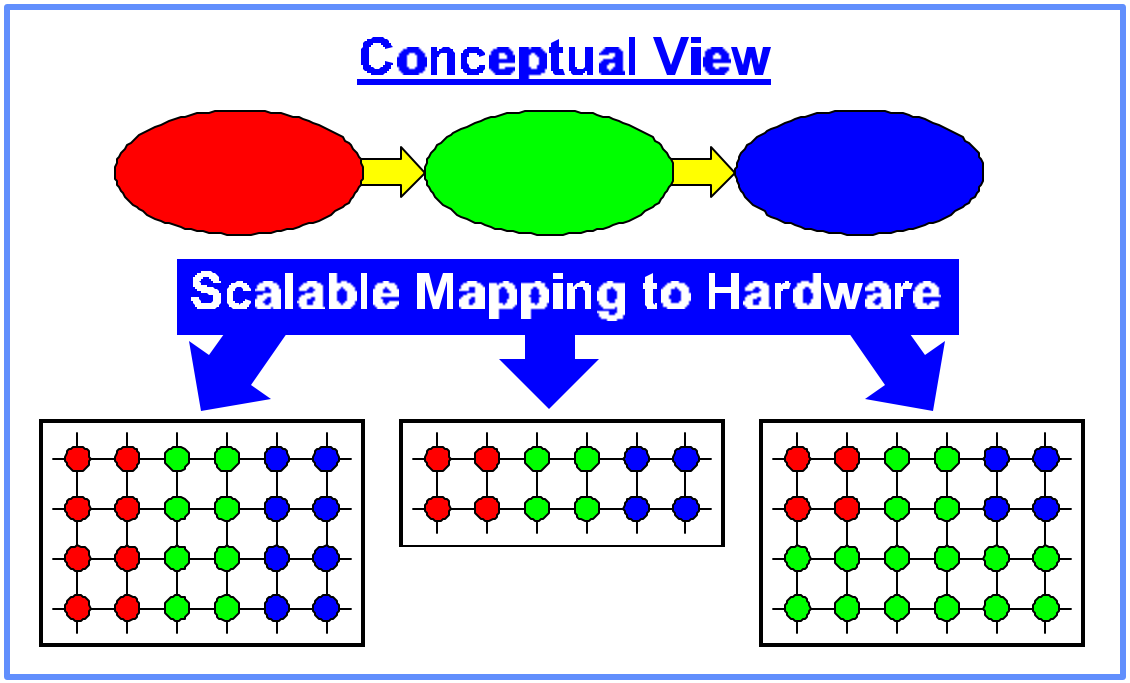
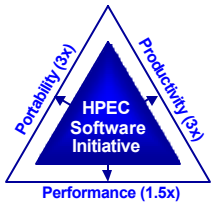


Figure courtesy of Dennis Cottel, USN SSC-SD

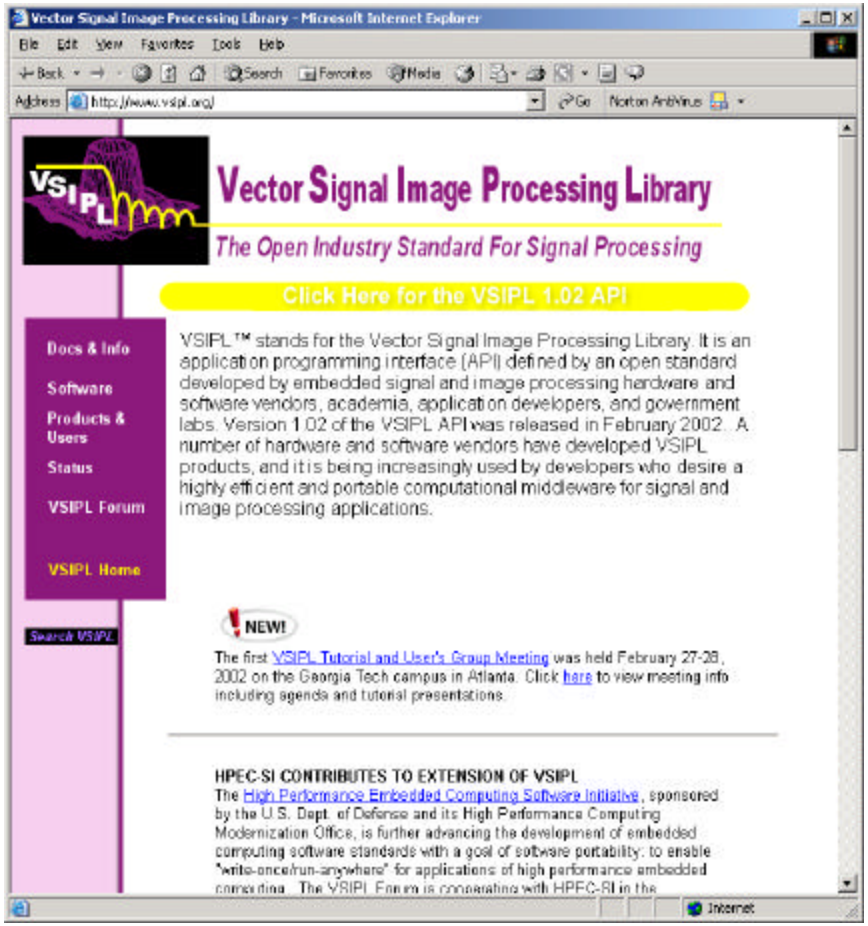
Parallel VSIPL Status

- HPEC-SI is researching approaches to development of parallel VSIPL
  - key issue is memory management strategy (blocks and views)
- Candidate components to build on include:
  - Data Reorganization Interface (DRI)
  - MIT/LL-Lockheed Martin Parallel Vector Library (PVL)
  - USN SSC-SD Scalable Programming Environment (SPE)
  - Commercial products such as GEDAE, MCCI Autocoder, Raytheon Sage
- Goal is to have a 0.1 draft specification and prototype software in Spring – Summer 2003



Summary

- VSIPL 1.02 is available
- VSIPL 1.1 in final edit, due 4Q CY2002
- Implementations are here
- VSIPL development is continuing
  - HPEC-SI leading extension to VSIPL++ and “Parallel VSIPL”



www.vsipl.org