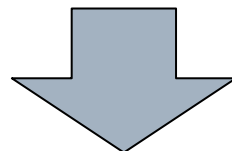


背景

- 携帯端末向け動画コンテンツの増加
 - 携帯端末向け放送(ワンセグ, マルチメディア放送)
 - インターネット配信(動画投稿, オンデマンド)
 - デジタル家電との連携(お出かけ転送, メディアサーバ)



- 携帯端末における高度な動画処理への要求
 - 多数の動画の並列復号(コンテンツ検索)
 - 高解像度動画の復号(フルセグ視聴など)

しかし、携帯端末の処理能力や資源には制約が大きく
これらの実現は困難

目的

携帯端末上で複数動画像の並列復号や
高解像度動画像の復号を可能とする
低演算量復号器を実装

携帯電話, MID, 音楽プレイヤー, 携帯型ゲーム機...



対象とする動画像符号化方式

□ 携帯端末向け動画像

- ワンセグ放送, インターネット配信

→ H.264 が普及

- 演算量の大きい符号化ツールを採用し, 高圧縮率を実現
- 復号に必要な演算量も大きい



□ 高解像度動画像

- デジタルテレビ(フルセグ)放送, パッケージソフト

→ MPEG-2 が普及

- H.264 よりも低演算量な符号化ツールを使用
- しかし, 画像サイズやフレームレートの差から復号演算量は携帯端末向け動画像の数十倍となる



最適化と簡略化による復号高速化

H.264 復号器の高速化

□ SIMD 演算による処理性能最適化

- OpenMAX(メディア処理 API)の最適化ライブラリを使用
- エントロピー復号, 逆量子化, 逆整数変換, イントラ予測, 動き補償, デブロッキングフィルタを最適化ライブラリ関数に置き換え

□ 処理簡略化による演算量削減

(1) デブロッキングフィルタ(DF)省略

- H.264 復号処理の中でも特に演算量の多いデブロッキングフィルタの処理を省略

(2) 輝度成分の動き補償(MC)簡略化

- 輝度成分の予測画像生成時に, 演算量の多い 6 タップフィルタではなく, バイリニアフィルタを使用

