

時空間連続 DP のハードウェア実装に向けた演算精度の検討

石川健太[†] 佐賀美直人[‡] 宮崎敏明[‡]

会津大学コンピュータ理工学部[†] 会津大学大学院コンピュータ理工学研究科[‡]

1. はじめに

時空間連続 DP (Time-Space Continuous Dynamic Programming: TSCDP) は、空中に手で描いた文字の認識など、幅広い応用が考えられる。しかしながら、TSCDP は演算量が多いため、ソフトウェアによる実装では、実時間処理が難しい。我々は、TSCDP のハードウェア化を進めているが、回路規模を抑えるために、演算器やレジスタのビット幅を出来るだけ小さくしたい。本稿では、様々な実入力動画像を用いて、TSCDP 内で使用されている変数のダイナミックレンジを評価し、ハードウェア化に必要な演算精度を明らかにする。

2. 時空間連続 DP

TSCDP[2]は、入力動画像から予めマッチングする領域を指定（これをセグメンテーションという）することなく、参照動画像の類似部分を認識するアルゴリズムである。これは、従来の連続 DP 法[1]を、動画像のような時空間パターンが扱えるように拡張した DP であり、時間方向の拡大および縮小に対応したものは下記のように定式化できる。

$$\begin{aligned} S(x, y, \tau, t) &= \infty, d(x, y, \tau, t) = \infty \text{ if } (x, y) \notin [M, N], t \leq 0, \tau \notin [1, T] \\ S(x, y, 1, t) &= 3d(x, y, 1, t); \\ \text{for } 2 \leq \tau \leq T; \\ S(x, y, \tau, t) &= \end{aligned}$$

$$\min \begin{cases} S(x - v_x(\tau), y - v_y(\tau), \tau - 1, t - 2) \\ \quad + 2d(x, y, \tau, t - 1) + d(x, y, \tau, t); \quad \textbf{(1)} \\ S(x - v_x(\tau), y - v_y(\tau), \tau - 1, t - 1) + 3d(x, y, \tau, t); \quad \textbf{(2)} \\ S(x - v_x(\tau) - v_x(\tau - 1), y - v_y(\tau) - v_y(\tau - 1), \tau - 2, t - 1) \\ \quad + 3d(x - v_x(\tau), y - v_y(\tau), \tau - 1, t) + 3d(x, y, \tau, t); \quad \textbf{(3)} \end{cases}$$

式 1 : TSCDP の時間正規化漸化式および初期条件

ここで、 S は評価関数、 d は局所距離であり入力動画像と参照動画像の指定座標におけるユークリッド距離で定義される。 x, y は入力動画像の座標であり、 M, N は入力動画像および参照動画像の

Arithmetic Precision of Time-space Continuous Dynamic Programming toward Its Dedicated Hardware

[†]Kenta Ishikawa, [‡]Sagami Naoto, [‡]Toshiaki Miyazaki.

[†]School of Computer Science and Engineering, The University of Aizu

[‡]Graduate School of Computer Science and Engineering, The University of Aizu

縦横の最大値である。また、 t は入力動画像の何フレーム目か、 τ は参照動画像の何フレーム目かを表し、 T は τ の最大値である。さらに、出力である参照動画像をスポッティングする時空間点 (DP マッチングにおけるバックトレースの開始点) は次式で表される。

$$(x^*(t), y^*(t)) = \arg \min_{(x, y) \in \{local\ area\}} \left\{ \frac{S(x, y, T, t)}{3T} \leq h \right\}$$

式 2 : スポッティングする時空間点

ここで、 h は閾値であり、予め与える定数である。上記式 2 では、 (x, y) に関する最小値の計算を *local area* (画像の局所領域) で行っている。これにより、画面上に複数の類似部分が存在する場合、複数の局所領域における最小値でそれぞれがスポッティングされるため、同時に複数の文字を認識可能である。

3. ハードウェアアーキテクチャ

図 1 に筆者らが設計を進めている TSCDP 専用ハードウェアの概要図を示す。"Calculation" は、式 1 で示した漸化式計算を行うブロックであり、本 Calculation 部内の破線ブロック (1), (2), (3) における加算・乗算は、それぞれ式 1 の漸化式右辺の 3 つの式に対応している。また、局所距離 d を計算するために、破線ブロック右側に 3 つの加算器を設けている。入力データは、カメラから直接入ることを想定しており、実行結果は VGA に出力することを想定している。本アーキテクチャは、漸化式計算を直接ハードウェアで行う単純な構成としているが、高速化のためには、本 Calculation 部を複数設けたマルチプロセッサ化も視野に入れている。

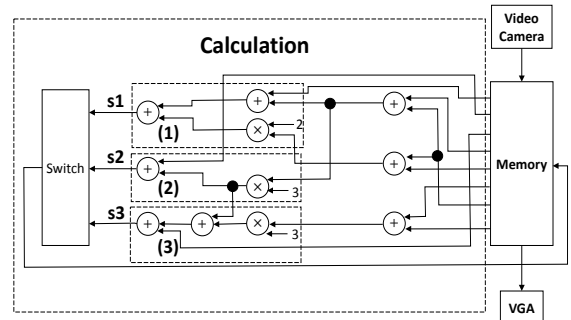


図 1 : TSCDP 専用ハードウェアの概要

4. 評価

本項では、図1の Calculation 部内で用いる演算器の精度およびレジスタ類のビット幅を決定するために、漸化式計算に関わる変数のダイナミックレンジを、実入力動画像を用いて確認した。具体的には、入力動画像として、空中に手で“A”，“B”，“C”をそれぞれ描く 80x60 ピクセルの動画像を用意した。また、各変数の入力動画像サイズの影響を調べるために、各動画像から、40x30 ピクセルおよび 20x15 ピクセルの動画像も作製した。用意した全ての動画像に TSCDP 処理を実行し、図1における信号 s1, s2, s3 に対応するプログラム内の double 型変数 s1, s2, s3 の値をダンプした。これらのダンプにより得られたヒストグラムの一部を図2および図3に示す。

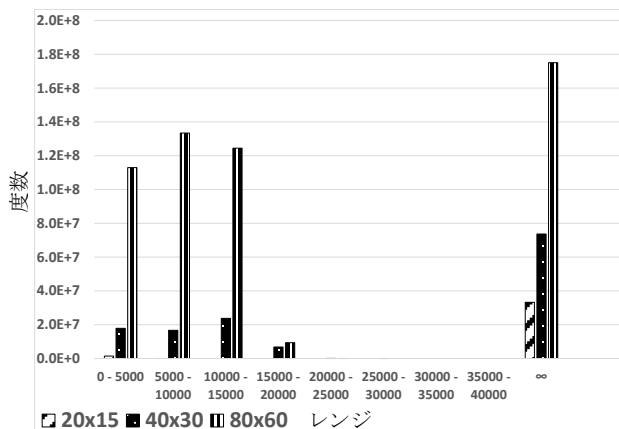


図2：“B”を描く入力動画像を処理した場合の各解像度における変数 s1 のヒストグラム

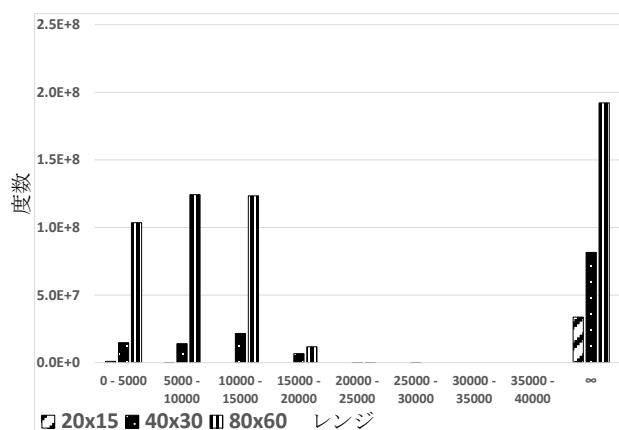


図3：“B”を描く入力動画像を処理した場合の各解像度における変数 s3 のヒストグラム

5. 評価結果

全ての評価の中で最もダイナミックレンジが

大きくなった“B”を描く入力動画像を処理した場合のダイナミックレンジおよび標準偏差を表1に示す。表中の値は整数で表記しているが、実際には各変数は計算中に小数点以下の値をとる。また、各標準偏差値が大きいことから、各変数 s1, s2, s3 の値はダイナミックレンジ内で万遍なく変化していることが分かる。表1のダイナミックレンジがそのまま変数の可変領域であるとすると、解像度 20x15 では、 $2^{13} = 8192 > 6506$ であることから、対応する信号を保持するには 13 ビット確保できれば充分であると分かる。同様に、解像度 40x30, 80x60 では、 $2^{15} = 32768 > 29729$ であることから、対応する信号を保持するには 15 ビット確保できれば充分である。さらに、プログラム内の式2に対応する処理部分（閾値 h との比較）についても評価したところ、小数点以下の精度を必要とし、整数で実現するためには 15 ビット左シフト（32768 倍）しなければならないことが分かった。上記の結果から、評価で用いた入力動画像の解像度において、TSCDP を実行するには各変数について $15 + 15 = 30$ ビットあれば充分であることが分かった。

表1：“B”を描く入力動画像を処理した場合のダイナミックレンジおよび標準偏差

解像度	20x15	40x30	80x60
変数 s1	100 – 6502 (895)	76 – 29397 (4851)	1 – 24936 (4175)
変数 s2	120 – 6506 (895)	75 – 29397 (4851)	1 – 24936 (4175)
変数 s3	114 – 6003 (851)	79 – 29729 (4887)	66 – 24444 (4214)

()内は標準偏差

6. おわりに

様々な実入力動画像を用いて、TSCDP の漸化式計算に関わる変数のダイナミックレンジを評価し、TSCDP 専用ハードウェアに必要な演算精度およびレジスタ類に必要なビット幅を明らかにした。これを踏まえ、具体的なハードウェアの実装を行い、TSCDP の実時間処理を目指す。

文献

- [1] R. Oka, “Spotting Method for Classification of Real World Data,” The Computer Journal, Vol. 41, No. 8, pp. 559–565, 1998
- [2] 岡隆一, 松崎隆, “人の動作のスポッティング認識に基づくエージェントのための動きモデル自動獲得,” HAI シンポジウム 2012, pp. 1–4, 2012.