Development of Two-dimensional Depth-averaged Model with Higher Order Scheme

Jia Jun, Lim
River Engineering and Urban Drainage Research Centre (REDAC)
Universiti Sains Malaysia (USM)
Nibong Tebal, Malaysia
ljj15_red004@student.usm.my

How Tion, Puay
River Engineering and Urban Drainage Research Centre (REDAC)
Universiti Sains Malaysia (USM)
Nibong Tebal, Malaysia
redac_puay@.usm.my

Nor Azazi, Zakaria
River Engineering and Urban Drainage Research Centre (REDAC)
Universiti Sains Malaysia (USM)
Nibong Tebal, Malaysia
redac01@usm.my

***Abstract*—Two-dimensional depth-averaged model or shallow water equation model is commonly used in solving the flow problem involving free-surface. A two-dimensional depth-averaged model (DA-CIP model) incorporate with Constrained Interpolation Profile (CIP) scheme is developed to solve the depth-averaged equation. The CIP scheme was developed to solve the hyperbolic equation. It allowed the construction of the solution inside the grid cell that close to the solution and required less cell information. The DA-CIP model developed is verified using several benchmarking dam break flow problems. From the verification procedures, it can observe that the DA-CIP model is able to solve the depth-averaged equation correctly and the reproduction of the dam break flow event is desirably good.**

Keywords—Constrained Interpolation Profile (CIP), Depth-averaged equation, dam break flow problem

I. INTRODUCTION

Two-dimensional depth-averaged model or the shallow water model is one of the most widely used approaches in describing the free-surface flow [1]. This set of equation is deduced from the Navier-Stokes equations by integrating through the vertical depth of water column [2], [3]. The following assumptions are implied when using the depth-averaged model approach: the density of fluid is constant; the viscosity term is negligible; and the vertical accelerations of fluid are negligible[3]. According to Akoh [4], the challenges faced by researcher developing numerical method based on depth-averaged approach are the nonlinear shock solutions exist in the equations. The study further highlighted that high-resolution schemes with exact or approximate Riemann solvers are proved very successful in solving the depth-averaged equations. [5]. Numerous typical numerical solution algorithms for depth-averaged equations are summarized in [6].

The Constrained Interpolation Profile (CIP) method was developed by [7] to solve the hyperbolic equation. CIP scheme can be considered as a kind of semi-Lagrangian method, where a Lagrangian invariant solution was employed by CIP scheme. The Lagrangian invariant solution used was based on spatial points and was always casted in a non-conservative form [8]. Semi-Lagrangian method that traced back along the characteristics in time depended on an interpolation of the initial profile to determine the value at the upstream departure point [9]. The CIP method proposed here allowed the construct of a solution inside the grid cell that was close to the real solution, with some constraints [9]. These constraints such as time evolution of spatial gradient or spatially integrated conservative quantities were used to construct the interpolation profile [9]. Since adaptive grid system was not required in CIP method, the problems of grid distortion caused by structural breakup and topology change can therefore be eliminated [9]. The CIP scheme was a robust and less diffusive solver, resulting in a solution of third order accuracy in space [10].

This study demonstrate the development of a high resolution numerical model based on the two-dimensional depth-averaged governing equations incorporate with CIP scheme [7]. The CIP scheme is employed to solve the advection part in the momentum equation, where this scheme is of third order accuracy. The DA-CIP model is verified with the benchmarking one-dimensional dam break flow problem and two-dimensional partial dam break flow problem to test its accuracy and stability.

II. GOVERNING EQUATION

The two-dimensional DA-CIP model is developed based on the following governing equations in their non-conservative form.

Continuity equation

$$\frac{\partial h}{\partial t}+u\frac{\partial h}{\partial x}+v\frac{\partial h}{\partial y}=-h\left(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}\right) \quad (1)$$

Momentum equation

$$\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y} = -g\left(\frac{\partial h}{\partial x}+\frac{\partial z_b}{\partial x}\right)-\frac{gn^2u\sqrt{u^2+v^2}}{h^{1/3}} +\frac{\partial}{\partial x}\left(-\overline{u'^2}h\right)+\frac{\partial}{\partial y}\left(-\overline{u'v'}h\right) \quad (2a)$$

$$\frac{\partial v}{\partial t}+u\frac{\partial v}{\partial x}+v\frac{\partial v}{\partial y} = -g\left(\frac{\partial h}{\partial y}+\frac{\partial z_b}{\partial y}\right)-\frac{gn^2v\sqrt{u^2+v^2}}{h^{1/3}} +\frac{\partial}{\partial x}\left(-\overline{u'v'}h\right)+\frac{\partial}{\partial y}\left(-\overline{v'^2}h\right) \quad (2b)$$

where u and v are the components of depth-averaged velocity vectors in x and y direction, h is the flow depth, n is the Manning's Coefficient, z_b is the vertical elevation of the channel bed from a horizontal reference datum and g is the acceleration due to gravity.

III. Numerical Algorithm

In the case of inviscid fluid, the bed shear stresses and turbulence terms are neglected, the depth-averaged continuity and momentum equation become,

$$\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}=-g\left(\frac{\partial h}{\partial x}+\frac{\partial z_b}{\partial x}\right)=-g\frac{\partial h}{\partial x} \quad (4a)$$

$$\frac{\partial v}{\partial t}+u\frac{\partial v}{\partial x}+v\frac{\partial v}{\partial y}=-g\left(\frac{\partial h}{\partial y}+\frac{\partial z_b}{\partial y}\right)=-g\frac{\partial h}{\partial y} \quad (4b)$$

Where in the case of flat bed, $\frac{\partial z_b}{\partial x}=\frac{\partial z_b}{\partial y}=0$.

In solving the governing equations, the time-splitting method [5] is used. the governing equations are solved in two stages, namely the advection stage and non-advection stage,

Advection stage:

From continuity equation:
$$\frac{\partial h}{\partial t}+u\frac{\partial h}{\partial x}+v\frac{\partial h}{\partial y}=0 \quad (5a)$$

From momentum equation:
$$\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}=0 \quad (5b)$$

$$\frac{\partial v}{\partial t}+u\frac{\partial v}{\partial x}+v\frac{\partial v}{\partial y}=0 \quad (5c)$$

Non-advection stage:

From continuity equation:
$$\frac{\partial h}{\partial t}=-h\left(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial x}\right) \quad (6a)$$

From momentum equation:
$$\frac{\partial u}{\partial t}=-g\frac{\partial h}{\partial x} \quad (6b)$$

$$\frac{\partial v}{\partial t}=-g\frac{\partial h}{\partial y} \quad (6c)$$

The numerical algorithm is summarized in Fig. 1. The variables and parameters used in the model are declared first followed by the set up the initial conditions and boundary conditions.

In the advection stage, the advection terms of the continuity and momentum equations (5a, 5b, 5c) are solved using the CIP scheme. Temporary values of h*, u*and v* are obtained after the advection terms are solved.

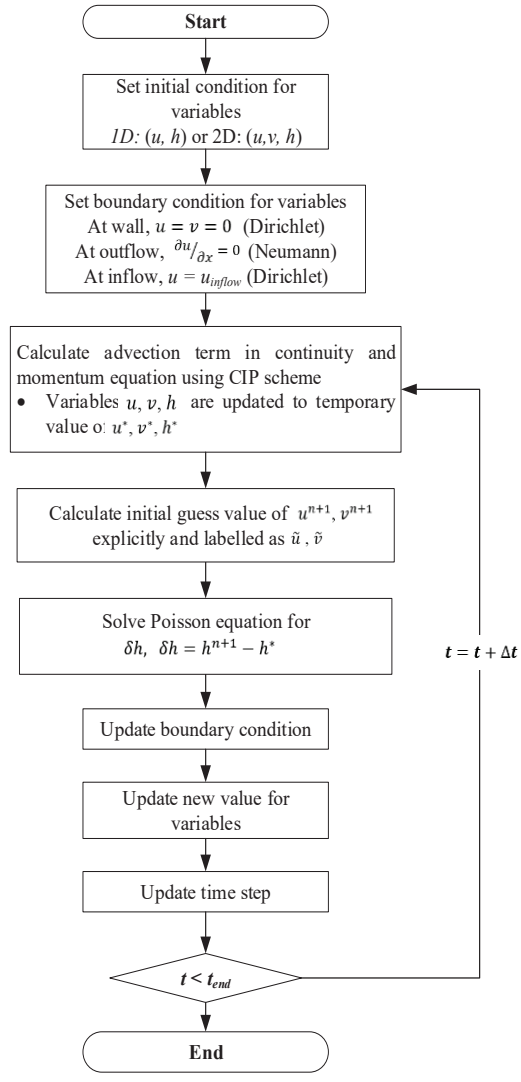

Fig. 1. Solution algorithm

In the non-advection stage, the non-advection terms (6a, 6b, 6c) are solved using the finite difference method and discretization is based on the staggered mesh system. The initial guess of the values u^{n+1} and v^{n+1} are calculated explicitly and labelled as $\tilde{u}$ and $\tilde{v}$.

By substituting the initial guess equation into the non-advection term of continuity equation, the following Poisson equation is obtained by substituting the

$$\frac{\partial^2(\delta h)}{\partial x^2}+\frac{\partial^2(\delta h)}{\partial y^2}=\frac{1}{gh^*(\Delta t)^2}\delta h+\frac{1}{g\Delta t}\left(\frac{\partial \tilde{u}}{\partial x}+\frac{\partial \tilde{v}}{\partial y}\right) \quad (11)$$

The Poisson equation is solved using the Successive Over Relaxation (SOR) method. The values of h, u and v are updated based on the δh obtained from solving the Poisson Equation. The updated value is ready for the next time step.

IV. VERIFICATION OF NUMERICAL MODEL

With the aid of the analytical solution, the accuracy and stability of a numerical model can be determined. If the performance of the verification processes is not satisfied, the model will undergo debugging procedures and fine-tuned accordingly. It is worth to know that it is more convenient and time-effective when debugging with one-dimensional form of a numerical model or simple two-dimensional problem like the partial dam break flow problem. Thus, these verification procedures are crucial when developing a numerical model. After the verification processes, the DA-CIP model is then ready for a more complex problem.

A. One-dimensional Dam Break Flow Problem

In order to investigate the performances of the numerical model, an idealised scenario of dam break flow problem in a rectangular channel was simulated. Ji *et al.* [11] mentioned that the dam break flow problem is a widely used benchmarking problem in investigating the accuracy and stability of the numerical model. The wide acceptance of the dam break flow problem as a verification test can be seen in previous research [11]–[15].

Researchers often test the accuracy of their numerical model with dam break flow problem before it was applied to the phenomenon addressed. The dam break flow problem includes a largely distorted free surface which allows the study of the capability of the numerical model to reproduce the discontinuous solution [11], [16]. Furthermore, the simplicity in setting up the initial and boundary conditions makes the verification process more straightforward for free surface flow [11], [13].

In this study, a one-dimensional dam break flow problem over a dry bed condition is being simulated to test the accuracy and numerical stability of the DA-CIP model. An idealized case of dam break flow scenario is considered when the vertical wall dividing two different water depths in a channel is suddenly removed. After the event of dam break, a negative wave propagates upstream and a dam break wave travels downstream.

The DA-CIP model is used to simulate the dam break flow problem of a domain range from 0 m $\leq x \leq$ 15.0 m with the following initial conditions:

$$h\,(x,\ t=0) = \begin{cases} 2.0 \text{ m},\ 0 \text{ m} \leq x \leq 3.0 \text{ m} \\ 0.0 \text{ m},\ x > 3.0 \end{cases} ;$$

$$u\,(x,\ t=0) = 0.0 \text{ m/s}$$

According to Chanson [17], Ritter's solution is read as

$$h = \frac{1}{9g}\left(2\sqrt{gh_0} - \frac{x}{t}\right)^2 \qquad (12)$$

where h is the flow depth, g is the gravitational acceleration, h_0 is the flow depth in the reservoir at the instantaneous stage of the dam collapse, x is the distance in x-coordinates and t is time.

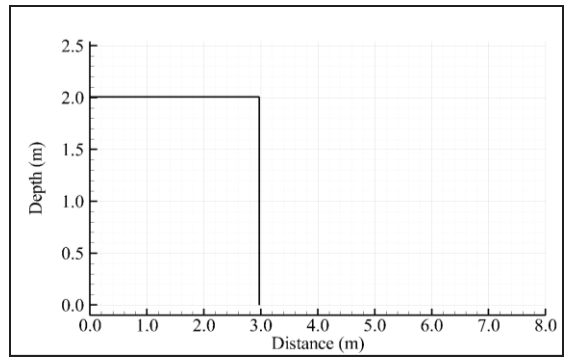

Fig. 2. Initial Condition for the one-dimensional dam break scenario. The figure showed the initial water depth of the reservoir at 2.0m before the dam break event.

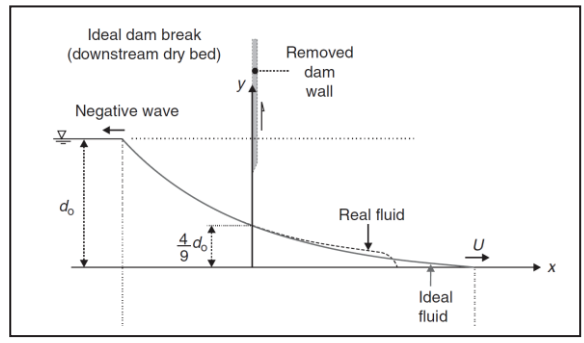

Fig. 3. Sketch of dam break wave in a dry horizontal channel with infinite length of reservoir. The water surface profile can be approximated by Ritter's solution [17].

Fig. 3 show the sketch of dam break wave adapted from Ritter's solution. The solution was formulated based assuming a smooth rectangular channel, an infinitely long reservoir and a horizontal surface [17]. Observing Fig. 3, d_o is identical to h_o , and the flow depth at the location of dam wall will always equal to $\frac{4}{9}d_0$ due to the properties of the dam break wave, where its profile is parabolic with tangent to the channel bottom at its wave front.

Fig. 4 - 10 show the comparison of the simulated solution with Ritter's solution[17], [18]. The results from DA-CIP models are indicated with filled circle while Ritter's solution is representing with a solid line in the figure. The dashed line shows the initial condition mentioned. The numerical result has good agreement with the Ritter's solution.

The numerical results (denoted with filled circle) appear to be close with the analytical solution (denoted as solid line) for time step 0.0 s to 1.0 s (Fig. 4 - 9). This suggests that the numerical results agreed closely with the Ritter's solution, in which the numerical model able to reproduce this dam break flow problem accurately.

By observing Fig. 8, the negative wave reached the wall of the reservoir at t=0.92 s. This is coinciding with the analytical result where according to [17], the

negative waves reached the wall at t=0.92 s for such configuration.

From Fig. 10, it can be observed that the numerical results and analytical solution do not agree well at the region circled in the figure. As mentioned earlier, infinitely long reservoir is assumed in Ritter's solution, therefore the negative waves will travel further upstream infinitely. While in this study, the upstream reservoir wall is set at x=0.0 m, therefore, the numerical results can only be validated with the Ritter's solution before the negative waves hit the upstream reservoir wall.

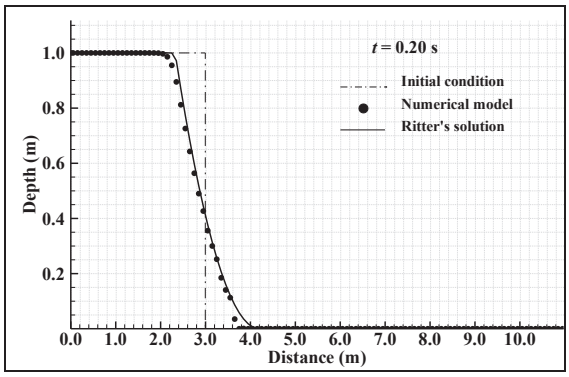

Fig. 4. The numerical and analytical solution for dam break flow over dry bed condition at time, t=0.20 s.

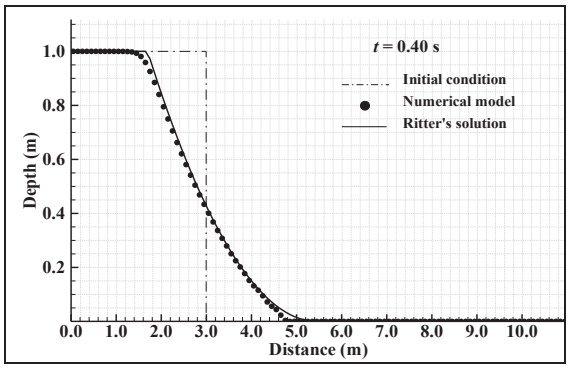

Fig. 5. The numerical and analytical solution for dam break flow over dry bed condition at time, t=0.40 s.

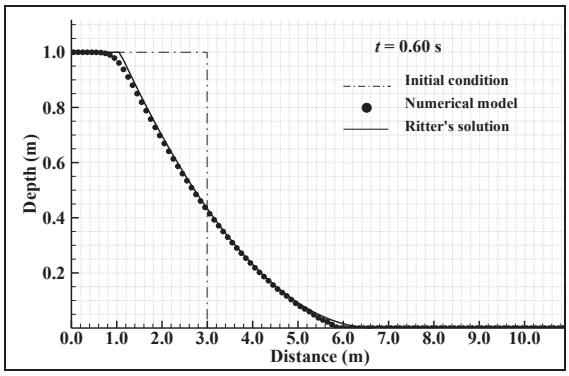

Fig. 6. The numerical and analytical solution for dam break flow over dry bed condition at time, t=0.60 s.

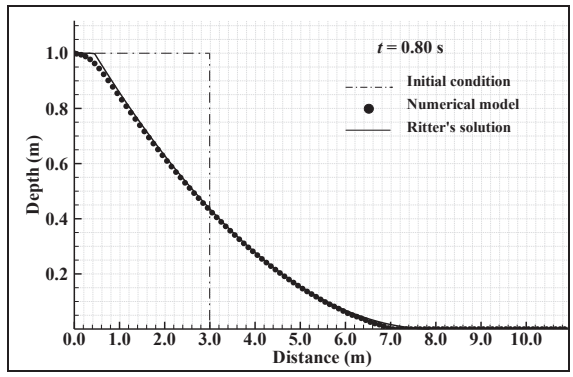

Fig. 7. The numerical and analytical solution for dam break flow over dry bed condition at time, t=0.80 s.

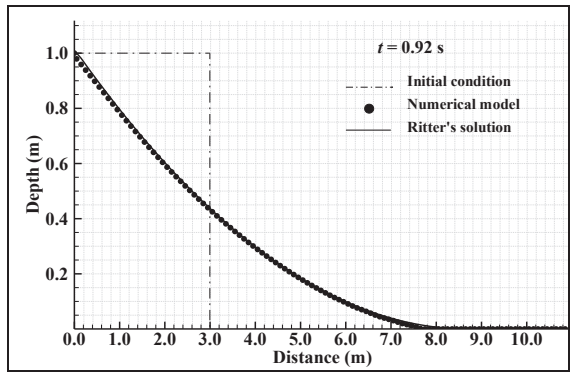

Fig. 8. The numerical and analytical solution for dam break flow over dry bed condition at time, t=0.92 s.

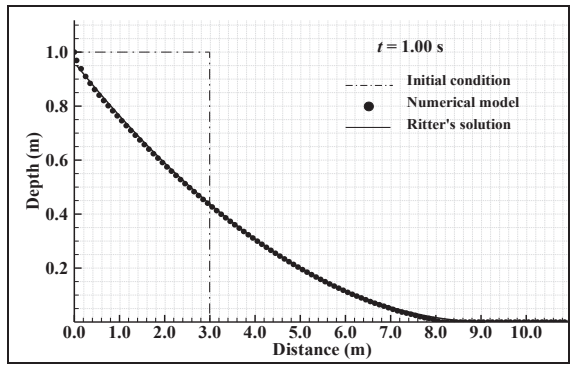

Fig. 9. The numerical and analytical solution for dam break flow over dry bed condition at time, t=1.00 s.

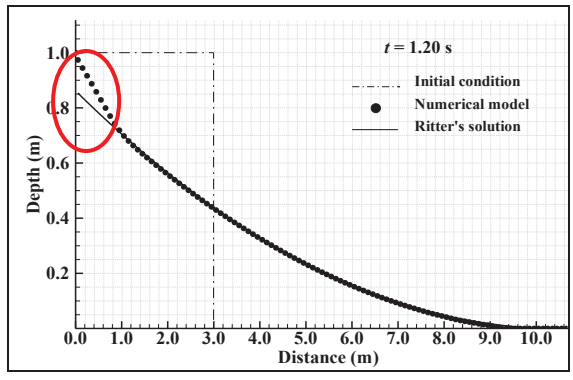

Fig. 10. The numerical and analytical solution for dam break flow over dry bed condition at time, t=1.20 s.

B. Two-dimensional Partial Dam Break Flow Problem

The depth-average model is simulated with two-dimensional partial dam break problem with the following initial condition:

$$h(x, t=0) = \begin{cases} 10.0 \text{ m}, & 0 \text{ m} \le x \le 100.0 \text{ m} \\ 5.0 \text{ m}, & 100.0 \text{ m} < x < 200.0 \text{ m} \end{cases};$$

$u(x, t=0) = 0.0$ m/s; $v(x, t=0) = 0.0$ m/s

The basin is of 200.0 m long and 100.0 m wide, surrounded by solid walls with a dam separating a reservoir from the floodplain. At initial, the water depth in the reservoir was 10.0 m and the water depth at floodplain was 5.0 m.

The model is simulated with a partial dam collapsed condition over a 75.0 m wide breach at location one-third of the dam.

Observing Fig. 11 - 16, the DA-CIP model is capable of reproducing the features of dam break such as the drop water level in the reservoir as well as the formation of front waves towards boundaries. Fig. 17 shows the simulation results from Akoh's study [4], where the present DA-CIP model is able to reproduce simulation result that coupled with previous research.

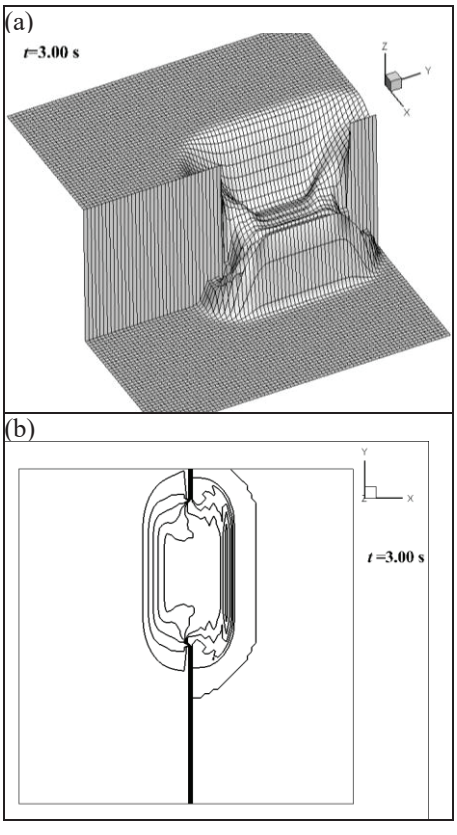

Fig. 11. Flow profile at t=3.00s (a) perspective view and (b) flow depth contour.

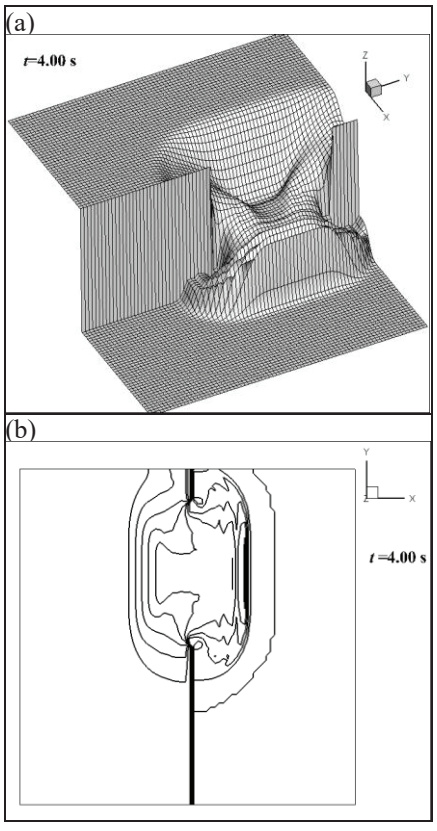

Fig. 12. Flow profile at t=4.00s (a) perspective view and (b) flow depth contour.

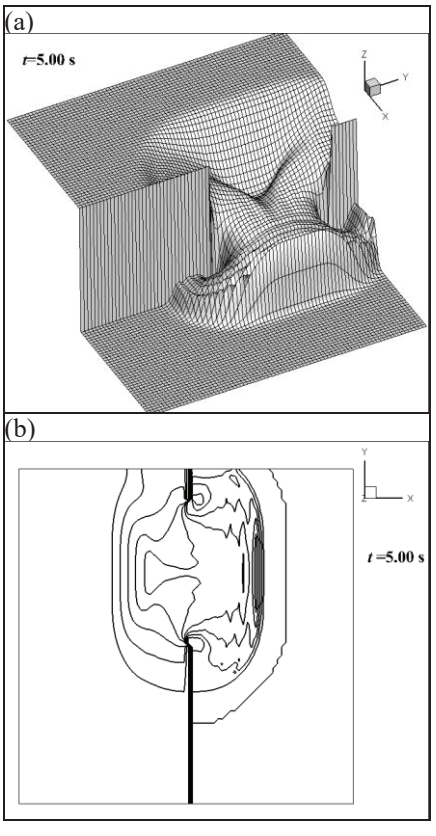

Fig. 13. Flow profile at t=5.00s (a) perspective view and (b) flow depth contour.

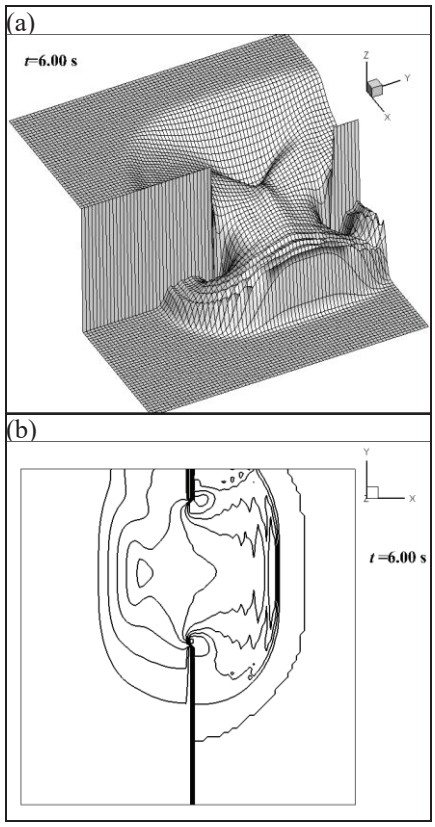

Fig. 14. Flow profile at t=6.00s (a) perspective view and (b) flow depth contour.

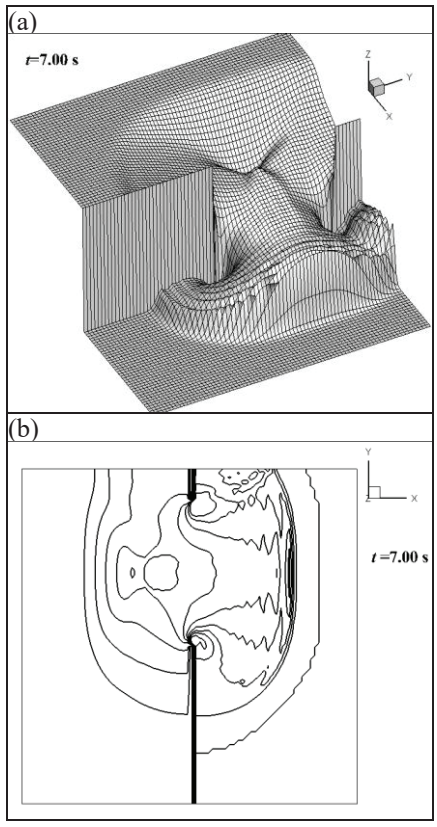

Fig. 15. Flow profile at t=7.00s (a) perspective view and (b) flow depth contour.

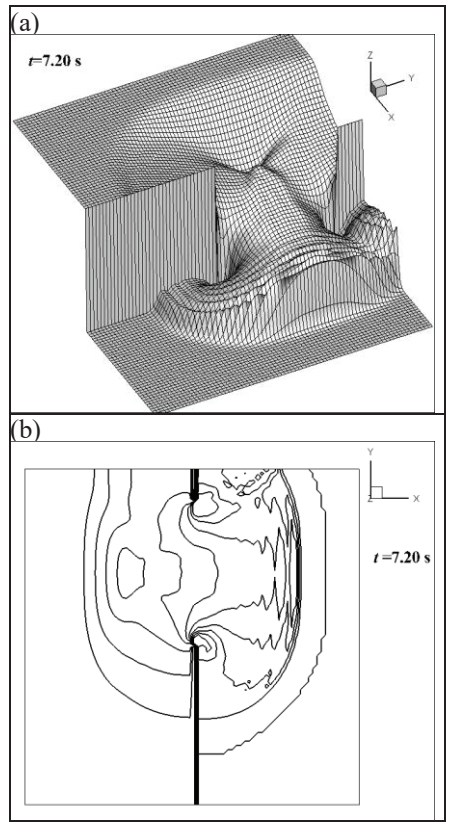

Fig. 16. Flow profile at t=7.20s (a) perspective view and (b) flow depth contour.

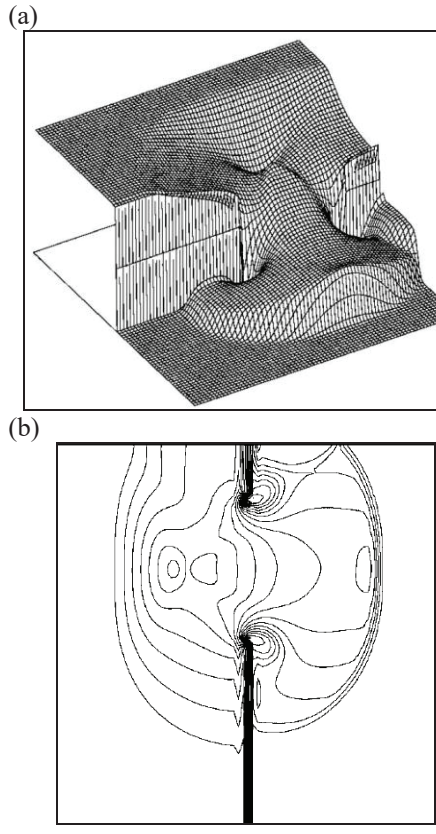

Fig. 17. Flow profile at t=7.20s from Akoh studies [4] (a) perspective view and (b) flow depth contour.

V. Conclusion

The two-dimensional DA-CIP model developed is successfully incorporated with CIP scheme. The DA-CIP model developed is able to solve the depth-averaged equation correctly. From the verification with one-dimensional dam break flow problem, the DA-CIP model is able to reproduce the scenario correctly. The speed of negative wave propagating upstream is accurately reproduced. Observing verification with two-dimensional partial dam break flow scenario, the is no unphysical features formed in the solution. The solution is considerably good when compare with previous literature studies.

References

[1] S. Dazzi, R. Vacondio, A. Dal Palù, and P. Mignosa, "A local time stepping algorithm for GPU-accelerated 2D shallow water models," *Adv. Water Resour.*, vol. 111, pp. 274–288, 2018.

[2] F. Alcrudo and P. Garcia-Navarro, "A high-resolution Godunov-type scheme in finite volumes for the 2D shallow-water equations," *Int. J. Numer. Methods Fluids*, vol. 16, no. 6, pp. 489–505, Mar. 1993.

[3] N. Benmansour, D. Ouazar, and L. C. Wrobel, "numerical simulation of shallow water wave propagation using a boundary element wave equation model," *Int. J. Numer. Methods Fluids*, vol. 24, no. 1, pp. 1–15, Jan. 1997.

[4] R. Akoh, S. Ii, and F. Xiao, "A CIP/multi-moment finite volume method for shallowwater equations with source terms," *Int. J. Numer. Methods Fluids*, vol. 56, no. October 2007, pp. 2245–2270, 2007.

[5] C. Lofthouse and A. Robert, "Riffle-pool sequences and meander morphology," *Geomorphology*, vol. 99, no. 1–4, pp. 214–223, 2008.

[6] E. F. Toro, *Shock-Capturing Methods for Free-Surface Shallow Flows*. New York: John Wiley, 2001.

[7] T. Yabe, "A universal cubic interpolation solver for compressible and incompressible fluids," *Shock Waves*, vol. 1, no. 3, pp. 187–195, 1991.

[8] T. Yabe, R. Tanaka, T. Nakamura, and F. Xiao, "An Exactly Conservative Semi-Lagrangian Scheme (CIP–CSL) in One Dimension," *Mon. Weather Rev.*, vol. 129, no. 1992, pp. 332–344, 2001.

[9] T. Yabe, F. Xiao, and T. Utsumi, "The Constrained Interpolation Profile Method for Multiphase Analysis," *J. Comput. Phys.*, vol. 169, no. 2, pp. 556–593, 2001.

[10] T. Utsumi, T. Kunugi, and T. Aoki, "Stability and accuracy of the Cubic Interpolated Propagation scheme," *Comput. Phys. Commun.*, vol. 101, no. 1–2, pp. 9–20, 1997.

[11] Q. L. Ji, X. Z. Zhao, and S. Dong, "Numerical study of violent impact flow using a CIP-based model," *J. Appl. Math.*, vol. 2013, 2013.

[12] V. Caleffi, A. Valiani, and A. Zanni, "Finite volume method for simulating extreme flood events in natural channels Finite volume method for simulating extreme flood events in natural channels Simulation des évènements hydrologiques extremes dans les cours d ' eau naturels par une methode en v," *J. Hydraul. Res.*, vol. 41, no. 2, pp. 167–177, 2003.

[13] Y. He, A. E. Bayly, A. Hassanpour, F. Muller, K. Wu, and D. Yang, "A GPU-based coupled SPH-DEM method for particle-fluid flow with free surfaces," *Powder Technol.*, vol. 338, pp. 548–562, 2018.

[14] L. Schippa and S. Pavan, "Numerical modelling of catastrophic events produced by mud or debris flows," *Int. J. Saf. Secur. Eng.*, vol. 1, no. 4, pp. 403–423, 2011.

[15] C. Yang, B. Lin, C. Jiang, and Y. Liu, "Predicting near-field dam-break flow and impact force using a 3D model," *J. Hydraul. Res.*, vol. 48, no. 6, pp. 784–792, 2010.

[16] V. Caleffi, A. Valiani, and A. Zanni, "Finite volume method for simulating extreme flood events in natural channels," *J. Hydraul. Res.*, vol. 41, no. 2, pp. 167–177, 2003.

[17] H. Chanson, *The hydraulics of open channel flow: An introduction*. Elsevier Ltd, 2004.

[18] O. Castro-Orgaz and H. Chanson, "Ritter's dry-bed dam-break flows: positive and negative wave dynamics," *Environ. Fluid Mech.*, vol. 17, no. 4, pp. 665–694, 2017.

IMPROVED IMAGE COMPRESSION USING LOSSLESS HUFFMAN ENCODING (I2COM)

Nirajan Bist
Department of Computer and Electronics Engineering
Kantipur Engineering College
Lalitpur, Nepal
neonbest12@gmail.com

Suraj Joshi
Department of Computer and Electronics Engineering
Kantipur Engineering College
Lalitpur, Nepal
surajjoshi0123@gmail.com

Abhishek Karki
Department of Computer and Electronics Engineering
Kantipur Engineering College
Lalitpur, Nepal
abhishekkarki17@gmail.com

Bharat Raj Joshi
Department of Computer and Electronics Engineering
Kantipur Engineering College
Lalitpur, Nepal
br14joshi@gmail.com

***Abstract*—With the development of information technology, image has become the mainstream of information transmission. Even if the storing capacity of storage devices has increased it is not sufficient for enormous pool of those images. The purpose of this work is to demonstrate robustness of simplistic implementation of Huffman encoding in lossless image compression. Some of the formats like JPEG 2000, PNG, etc. are available for lossless compressed image formats are highly used in practice currently but I2COM provides better compression ratio than these available formats in case of small sized images. We have compressed image data along with mapping table which increased the compression Ratio about 30% as compared to compression without compressing the mapping table. I2COM is lightweight image compressor which takes uncompressed BMP image file and compresses it using Huffman Encoding among various available lossless compression technique. Compression is based on various redundant information in file and I2COM uses the frequency of color as a redundant information. Resulting compression ratio (ratio of original size to compressed size) can be normally expected up to 7:1 or more depending upon the number of colors used in the image.**

***Index Terms*—Lossless Image Compression, Huffman Encoding, Compression Ratio**

I. Introduction

Compression of images is a very useful application for saving the storage data. Main motive of compression is to diminish superfluity or redundant information of the image data in order to make storage or transmission more efficient [1]. The goal of the compression is to keep the image quality intact and minimum processing resources. Images may contain spatial redundancy, coding redundancy, grayscale redundancy and frequency redundancy [2]. This project focus on frequency redundancy of colors to reduce the image file size. The compression may be Lossy or Lossless on the fact that the original image reconstruction ability after compression. Lossy compression involves quantization and entropy coding whereas Lossless compression involves entropy coding only [3]. Huffman encoding serves as lossless compression technique that exploits sequences statistics by assigning frequently used symbols fewer bits than rare symbols [4]. Appropriate headers are to be added to the compressed file. The field of image compression continues to grow at a rapid pace. As we look to the future, the need to store and transmit images will only continue to increase faster than the available capability to process all the data. Image compression involves reducing the size of image data files, while retaining necessary information. Retaining necessary information depends upon the application.

II. Methodology

A. Background

Huffman coding is a method for lossless data compression and it is independent of data type. Coding process begins by collecting all the probabilities for particular color in descending order. The process starts from bottom i.e leaf nodes and finally making a Huffman tree. At every leaf of tree, color nodes are present with each node have color frequency and color value in RGB format. The process follows in steps. In each step, two color nodes having smallest probabilities are extracted from the sorted list of color nodes. The color nodes are linked by a new node with its probability as sum of probability of those two nodes and is added to the tree. The process goes on until only one node remains which becomes root node for Huffman tree. In this paper the Huffman technique will be accomplished or applied after breaking the image into small parts

(rows and columns) and apply this technique to each part.

The problem can be divided into three major parts: Image Acquisition: In our work, image is initially read from user to be decompressed through User Interface (UI) and then it is converted into numpy array with the aid of Python Imaging Library (PIL) module in Python.

Image Compression: Frequency table of the unique colors used in the image is formed using Priority Queue. Finally table along with mapped binary values of individual image pixels are written in compressed file.

Decompression: First table is loaded in a dictionary, and reverse mapping of the image pixels is performed to get actual color of individual pixel which gives a numpy array. Finally we write the image from the array to a file using Python Imaging Library (PIL) module.

B. Algorithm

Algorithm 1 Encoding Algorithm

```
Read image to be compressed.
Initialize Priority Queue (P) empty.
for all pixel in image do
    if pixel color not in P then
        Form node(N) with frequency = 1
        Put N into P
    else
        Increase color frequency by 1
    end if
end for
Prepare Huffman Tree using the Priority Queue
Create Mapping Dictionary containing color codes
for each color from the tree
Save Mapping Table to file.
for all pixel in image do
    Write corresponding codes of the pixel color in
    file using dictionary
end for
```

III. Image Compression

Huffman coding can be demonstrated clearly by compressing a raster image. Suppose we have a 5x5 raster image with 24-bit color(8-bit for each channel). The uncompressed image will take 5 x 5 x 24 = 600 bits of storage.

First, we count up how many times each color occurs in the image. Then we sort the colors in order of decreasing frequency. We end up with a row that looks like Fig. 1.(b)

Now we put the colors together by building a tree such that the colors farthest from the root are the least frequent. The colors are joined in pairs, with a node forming the connection. A node can connect either to another node or to a color. In our example, the tree might look like Fig. 1.(c)

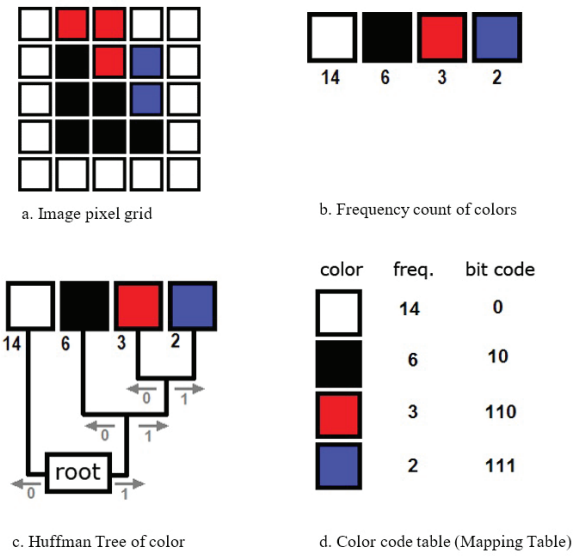

Fig. 1. Example of image compression process [5]

Our result is known as a Huffman tree. It can be used for encoding and decoding. Each color is encoded as follows. We create codes by moving from the root of the tree to each color. If we turn right at a node, we write a 1, and if we turn left 0. This process yields a Huffman code table in which each symbol is assigned a bit code such that the most frequently occurring symbol has the shortest code, while the least common symbol is given the longest code.

Because each color has a unique bit code that is not a prefix of any other, the colors can be replaced by their bit codes in the image file. The most frequently occurring color, white, will be represented with just a single bit rather than 8 bits. Black will take two bits. Red and blue will take three. After these replacements are made, the 600-bit image will be compressed to 14 x 1 + 6 x 2 + 3 x 3 + 2 x 3 = 41 bits.Further for decompression we need to save huffman color mapping table, and image size.

A. Comparative Measures

The compression ratio is defined as the ratio of original file size to the compressed file size.

$$C_R = \frac{s1}{s2} \tag{1}$$

Where s1 is the data rate of original image and s2 is that of the encoded bit-stream.

TABLE I
Comparative Compression Ratio Table

Image Name	Compression Ratio(CR)		
	I2COM	*JPEG 2000*	*PNG*
threecolor.bmp	12.01	10.12	10.32
leaves.bmp	11.34	10.11	9.27
ovalcircle.bmp	12.09	10.33	9.24
car.bmp	6.59	8.74	7.84
roads.bmp	3.71	7.64	6.52

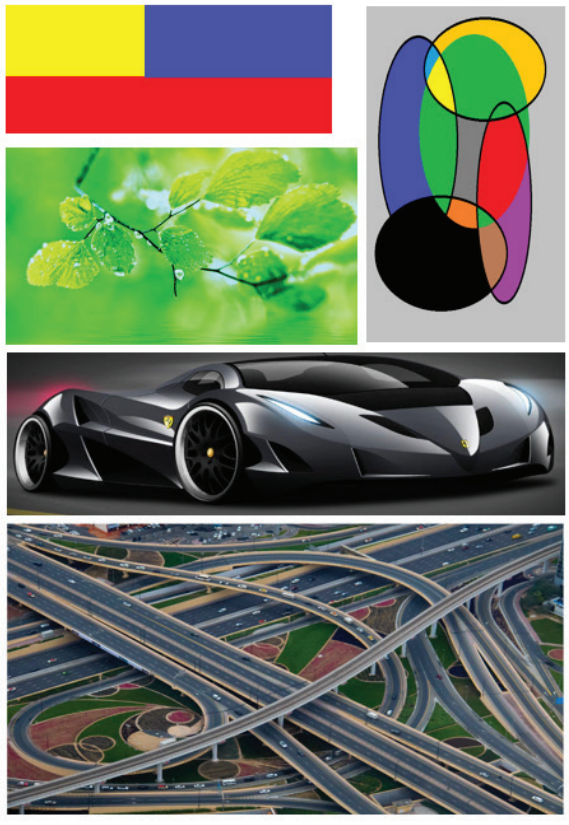

Fig. 2. Images used in comparison

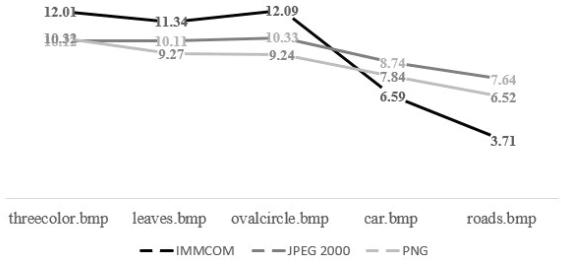

Fig. 3. Comparison Chart for Compression Ratio

TABLE II
UNCOMPRESSED BMP FILE INFORMATION TABLE

Image Name	*Size(KB)*	*Width(px)*	*Height(px)*
threecolor.bmp	187	270	236
leaves.bmp	4110	1600	900
ovalcircle.bmp	910	513	758
car.bmp	5774	1908	1044
roads.bmp	8593	2048	1260

IV. Conclusion and Future Enhancements

The method of splitting an image into different rows and columns will not only provides better results in image compression but also helpful for security purpose of the image transmission. Huffman Encoding method for the image compression has given fair compression ratio average 7:1. This has advantage over complex iterative and progressive algorithms that it is straight forward algorithm to focus on color frequency redundancy. It has turned out to be simple but robust method of image compression. We have compressed image data along with mapping table which increased the compression ratio about 30% as compared to compression without compressing the mapping table. Thus we got some improvement in the compression size in expense of computational speed.

Further we can see that the images compressed with I2COM technique is more efficient in compressing the small sized images than than larger images in comparison to two other lossless compression techniques JPEG 2000 and PNG.

This project can be extended to support lossless compression of videos as well. Other algorithms like Lempel-Ziv-Welch ,a highly advance universal lossless compression, coordinated with this project may help in compressing the larger images efficiently.

Acknowledgment

The authors acknowledge Er. Bishal Thapa and Er. Dipesh Shrestha for co-operating with us and providing us valuable guidance in completing this project .

References

[1] A. K. Jain, Image data compression: A review", in Proc. IEEE, vo169, pp. 349-389, 1981.
[2] G.C Chang Y.D Lin (2010) An Efficient Lossless ECG Compression Method Using Delta Coding and Optimal Selective Huffman Coding IFMBE proceedings 2010, Volume 31, Part 6, 1327-1330, DOI: 10.1007/978-3-642-14515-538.
[3] . Dr. T. Bhaskara Reddy, Miss.Hema suresh yaragunti, Dr.S.kiran, Mrs.T.Anuradha A novel approach of lossless image compression using hashing and Huffman coding ,International Journal of Engineering research and technology ,vol.2 issue 3,march-2013.
[4] M. Yang and N. Bourbakis. An Overview of Lossless Digital Image Compression Techniques, IEEE, pp. 1099-1102. 2005
[5] Huffman Coding - Base of JPEG Image Compression. [online] Available at: https://www.print-driver.com/stories/huffman-coding-jpeg