RESEARCH

Formal Analysis of 2D Image Processing Filters using Higher-order-logic Theorem Proving

Adnan Rashid[1], Sa'ed Abed[2*] and Osman Hasan[1]

*Correspondence: s.abed@ku.edu.kw
[2]Computer Engineering Department, College of Engineering and Petroleum, Kuwait University, Kuwait City, Kuwait
Full list of author information is available at the end of the article

Abstract

Two-dimensional (2D) image processing systems are concerned with the processing of the images represented as 2D arrays and are widely used in medicine, transportation and many other autonomous systems. The dynamics of these systems are generally modeled using 2D difference equations, which are mathematically analyzed using the 2D z-transform. It mainly involves a transformation of the difference equations based models of these systems to their corresponding algebraic equations, mapping the 2D arrays (2D discrete-time signals) over the (z_1,z_2)-domain. Finally, these (z_1,z_2)-domain representations are used to analyze various properties of these systems, such as transfer function and stability. Conventional techniques, such as paper-and-pencil proof methods and computer based simulation techniques for analyzing these filters cannot assert the accuracy of the analysis due to their inherent limitations like human error proneness, limited computational resources and approximations of the mathematical expressions and results. In this paper, as a complimentary technique, we propose to use formal methods, higher-order-logic (HOL) theorem proving, for formally analyzing the image processing filters. These methods can overcome the limitations of the conventional techniques and thus ascertain the accuracy of the analysis. In particular, we formalize the 2D z-transform based on the multivariate theories of calculus using the HOL Light theorem prover. Moreover, we formally analyze a generic (L_1, L_2)-order 2D Infinite Impulse Response (IIR) image processing filter. We illustrate the practical effectiveness of our proposed approach by formally analyzing a second-order image processing filter.

Keywords: Formal Analysis; 2D Image Processing Systems; 2D z-transform; Theorem Proving; HOL Light; Higher-order Logic

Introduction

Two-dimensional (2D) image processing systems [1,2] typically involve image filtering, editing, enhancement, compression and restoration of the images represented as 2D arrays (2D discrete-time signals). Image processing filters [2] are the fundamental components of the 2D image processing systems that are widely used for image filtering. These filters are categorized as high-pass, band-pass and low-pass filters based on the passage of the allowable range of frequencies. For example, a high-pass filter permits a range of frequencies greater than a certain threshold. Moreover, these filters are widely used in autonomous vehicles [3,4], and medicine [5]. For example, they are used to perform various image processing tasks for controlling the autonomous vehicles, such as noise reduction, color normalization, histogram equalization and edge detection to enhance the quality of the images captured using various devices such as Closed-circuit Television (CCTV) and webcams [6]. Similarly, they are widely used in medicine for performing various image pre and post processing tasks, such as image quality enhancement, noise removal and image smoothing [5].

The dynamics of these image processing systems are generally modeled using 2D difference equations. Next, the 2D z-transform is used to mathematically analyze these systems. It mainly involves a transformation of the difference equations based models of these systems to their corresponding algebraic equations, using the definition and various classical properties of the 2D z-transform, while mapping 2D arrays over the (z_1,z_2)-domain. Finally, these (z_1,z_2)-domain representations are used to analyze various properties of these image processing systems like transfer function and stability [2].

Conventionally, the image processing filters have been analyzed using paper-and-pencil proof techniques and computer based symbolic and numerical methods. However, in the former case, the analysis is error-prone due to the highly involved human manipulation, particularly, for analyzing the larger and complex image processing systems and thus we cannot ascertain an absolute accuracy of the analysis in this approach. Similarly, the later approaches suffer from some of their inherent limitations. For example, the symbolic methods involve a large number of unverified symbolic procedures residing in the root of the associated tools [7]. Similarly, the numerical techniques include a finite number of iterations due to the limited power of the computing machines. Moreover, they are based on the mathematical results that are approximated due to the finite precision arithmetic of computers. Therefore, these conventional approaches cannot be trusted when analyzing the image processing filters utilized in various safety-critical areas, such as autonomous driving and medicine.

Formal methods [8] are system analysis techniques that are based on developing a mathematical model of the system using logic and verifying its various properties using deductive reasoning. Higher-order-logic (HOL) theorem proving [9, 10] is a widely utilized formal method for analyzing many safety-critical systems. In this paper, we propose a HOL theorem proving based framework for analyzing the image processing filters. In particular, we formalize the 2D z-transform based on the multivariate theories of calculus using the HOL Light theorem prover [11]. The main motivation of selecting HOL Light for the proposed formalization is the presence of the fundamental libraries of multivariate calculus[1], vectors[2] and matrices[3], which are required to formally analyze the 2D image processing systems.

Contributions of the Paper

The novel contributions of the paper are:

- Formalization of 2D z-transform and its Region of Convergence (ROC).
- Formal verification of various classical properties of 2D z-transform, such as linearity, shifting in time-domain, scaling in (z_1, z_2)-domain and complex conjugation.
- Formal analysis of a generic (L_1, L_2)-order 2D IIR image processing filter.
- Formal analysis of a second-order image processing filter

[1] https://github.com/jrh13/hol-light/blob/master/Multivariate

[2] https://github.com/jrh13/hol-light/blob/master/Multivariate/vectors.ml

[3] https://github.com/jrh13/hol-light/blob/master/Multivariate/vectors.ml

Preliminaries

This section provides an introduction to the HOL Light theorem prover and the formalization of some fundamental concepts from the multivariate calculus libraries of HOL Light that facilitate the understanding of the rest of the paper.

HOL Light Theorem Prover

HOL Light [12] is a proof assistant for developing proofs of the mathematical concepts written as theorems in higher-order logic. HOL Light is implemented in the strongly-typed functional programming language ML [13]. A theorem is a statement that is formalized as an axiom or can be implied from already verified theorems using inference rules. Soundness is assured in a theorem proving environment as every new theorem is verified using the primitive inference rules or any other previously verified theorems. HOL Light provides an extensive support of theories, such as Boolean algebra, arithmetic, real numbers, vectors and matrices, which are extensively used in our formalization. Indeed, one of the motivations for selecting the HOL Light theorem prover for the proposed framework is the availability of extensive libraries of vectors and matrices.

Multivariable Calculus Theories in HOL Light

This section presents an introduction to some fundamental concepts formalized in HOL Light, such as summability, infinite summation and vector summation, and some HOL Light notations that help understanding the rest of the paper.

An N-dimensional vector in HOL Light is formalized as a $\mathbb{R}^N$ column matrix capturing individual elements as real numbers. All vector operations are then considered as matrix manipulations. Most of the theorems in multivariable calculus theories of HOL Light are proved for functions with an arbitrary data-type of $\mathbb{R}^M \rightarrow \mathbb{R}^N$. Similarly, complex numbers ($\mathbb{C}$) can be described as $\mathbb{R}^2$ instead of defining a new datatype. The HOL Light symbol &: $\mathbb{N} \rightarrow \mathbb{R}$ represents an injection of natural numbers to real numbers. Similarly, the symbol Cx: $\mathbb{R} \rightarrow \mathbb{C}$ typecasts real numbers to complex numbers. The symbols Re: $\mathbb{C} \rightarrow \mathbb{R}$ and Im: $\mathbb{C} \rightarrow \mathbb{R}$ represent the real and imaginary components of a complex number, respectively. The HOL Light symbol % captures the scalar multiplication of a vector or matrix. Similarly, a matrix-vector multiplication is modelled as $**$ in HOL Light.

The generalized summation over an arbitrary function fn: A $\rightarrow \mathbb{R}^N$ is formalized in HOL Light as follows:

Definition 1 Generalized Summation of Vector
$\vdash_{def}$ ∀st fn. vecsum st fn = (lambda k. summ st (λx. fn x$k))

where vecsum accepts a set st: A → bool over which the summation occurs and a function fn of data-type A $\rightarrow \mathbb{R}^N$ and returns a generalized vector summation over the set st. Here, the HOL Light function summ provides a finite summation for a fn over real numbers. For example, a mathematical expression $\sum_{k=0}^{n} f(k)$ is described in HOL Light as vecsum (0..n) fn.

Definition 2 Summs

$\vdash_{def}$ ∀st fn lt. (fn summs lt) st ⇔

((λn. vecsum (st INTER (0..n)) fn) → lt) sequentially

The HOL Light function summs accepts a set of natural numbers st: N → bool, a function fn: $\mathbb{N} \rightarrow \mathbb{R}^N$ and a limit value lt: $\mathbb{R}^N$ and returns the traditional mathematical expression $\sum_{k=0}^{\infty} f(k) = L$. Here, INTER captures the intersection of two sets. Similarly, sequentially represents a net providing a sequential growth of a function f, i.e., $f(k), f(k+1), f(k+2), ...$, etc. This is mainly used in modeling the concept of an infinite summation.

We provide the formalization of the summability of a function fn: $\mathbb{N} \rightarrow \mathbb{R}^N$ over st: N → bool, which ensures that there exist some limit value L: $\mathbb{R}^N$, such that $\sum_{k=0}^{\infty} f(k) = L$ in HOL Light as:

Definition 3 Summability of a Function

$\vdash_{def}$ ∀fn st. summable fn st ⇔ (∃lt. (fn summs lt) st)

The limit of a function fn: A → $\mathbb{R}^N$ is formalized as:

Definition 4 Limit of a Function

$\vdash_{def}$ ∀net f. limt net fn = (∈lt. (fn → lt) net)

where the function limt takes a net with components of data-type A and a function fn, and returns a limit value lt: $\mathbb{R}^N$ to which fn converges at the given net. It is formalized using the Hilbert choice operator ∈. Similarly, the concept *tends to* (→) is formalized in HOL Light as:

Definition 5 Tends to

$\vdash_{def}$ ∀fn lt net. (fn → lt) net ⇔

∀e. &0 < e ⇒ eventually (λx. dist (fn x, lt) < e) net

Now, we provide a formalization of an infinite summation, which is used in the formal definition of the 2D z-transform presented in Section *Formalization of 2D z-Transform*.

Definition 6 Infinite Summation of a Function

$\vdash_{def}$ ∀fn st. inftsumm st fn = (∈lt. (fn summs lt) st)

where the HOL Light function inftsumm accepts st: num → bool specifying the starting point and a function fn of data-type $\mathbb{N} \rightarrow \mathbb{R}^N$, and returns a limit value lt: $\mathbb{R}^N$ to which the infinite summation of fn converges from the given st.

Next, we formally verify an equivalence of the infinite summation (Definition 6) to its alternate form in terms of sequential limit as the following HOL Light theorem:

Theorem 1 Relationship Between Infinite Summation and the Sequential Limit

$\vdash_{thm}$ ∀st fn. inftsumm st fn = limt sequentially (λk. vecsum (st INTER (0..k)) fn)

Methods

Figure 1 depicts our proposed method for analyzing the image processing filters using HOL theorem proving. The user provides the 2D difference equations that models the dynamics of the image processing system, which needs to be analyzed. This 2D difference equation is modeled in higher-order logic using the multivariate calculus theories of HOL Light. In the next step, we formalize the 2D z-transform that is required for mathematically analyzing the image processing systems. It mainly transforms the difference equations based models of these systems to their corresponding algebraic equations, using the definition and various classical properties, such as, linearity, shifting and scaling, of the 2D z-transform, while mapping 2D arrays over the (z_1,z_2)-domain. Finally, these (z_1,z_2)-domain representations are used to analyze various properties of these systems, such as transfer function and the solution of the corresponding difference equations.

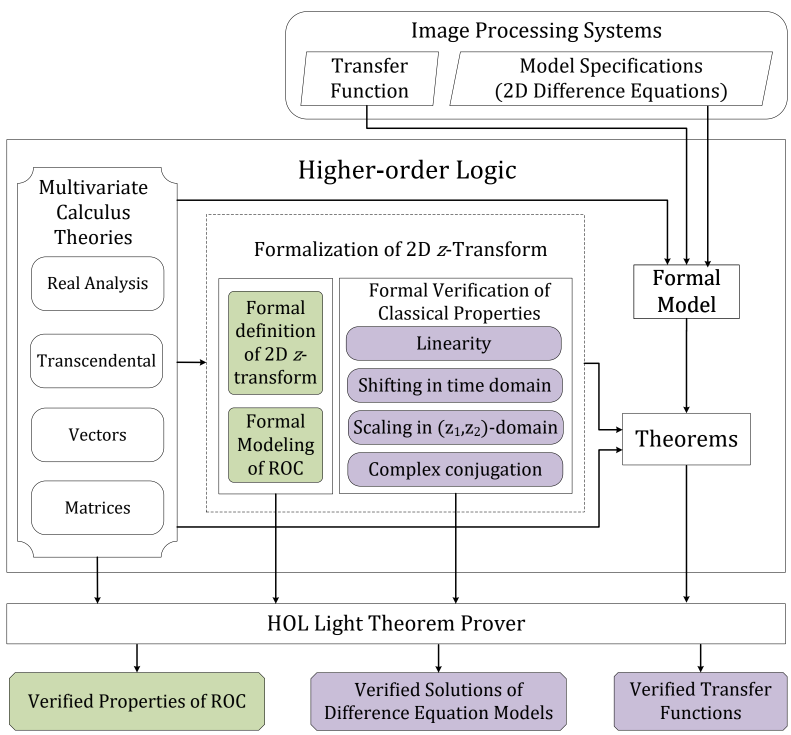

Figure 1: Proposed Framework

Results

Formalization of the 2D z-Transform

The 2D z-transform of a 2D discrete-time function (2D array) $f(n_1, n_2)$ is mathematically expressed as follows [2]:

$$F(z_1, z_2) = \sum_{n_1=0}^{\infty} \sum_{n_2=0}^{\infty} f(n_1, n_2) {z_1}^{-n_1} {z_2}^{-n_2} \qquad (1)$$

where f is a function of type $\mathbb{N} \to \mathbb{N} \to \mathbb{C}$, and z_1 and z_2 are complex variables. The limits from 0 to ∞ make Equation (1) as a mathematical representation of a unilateral 2D z-transform. We have opted for this representation based on the same motivation that was considered for one-dimensional z-transform [14] and the Laplace transform [15].

We formalize the 2D z-transform (Equation (1)) in HOL Light as follows:

Definition 7 2D z-Transform
$\vdash_{def}$ ∀f z1 z2. z_transform_2d f z1 z2 = inftsumm (from 0)
(λn1. inftsumm (from 0) (λn2. f n1 n2 / (z1 pow n1 ∗ z2 pow n2)))

where z_transform_2d accepts a function of type $\mathbb{N} \to \mathbb{N} \to \mathbb{C}$ and two complex variables z1: $\mathbb{C}$ and z2: $\mathbb{C}$, and returns a complex number, which represents the 2D z-transform of f: $\mathbb{N} \to \mathbb{N} \to \mathbb{C}$ according to Equation (1).

An essential issue with the applicability of the 2D z-transform of $f(n_1, n_2)$ is the existence of $F(z_1, z_2)$ that occurs due to the presence of the infinite summations in Equation (1). Thus, we need to identify conditions for the existence of the 2D z-transform. A set of all those values of z_1 and z_2 for which the infinite summations are converging and $F(z_1, z_2)$ is finite (or summable) is known as the ROC. It is mathematically expressed as follows:

$$ROC = z_1, z_2 \in \mathbb{C} : \exists k. \sum_{n_1=0}^{\infty} \sum_{n_2=0}^{\infty} f(n_1, n_2) {z_1}^{-n_1} {z_2}^{-n_2} = k \tag{2}$$

We formalize the ROC of the 2D z-transform as follows:

Definition 8 Region of Convergence (ROC)
$\vdash_{def}$ ∀f n1. ROC_2d f n1 =
{(z1, z2) | ¬(z1 = Cx(&0)) ∧ ¬(z2 = Cx(&0)) ∧
z_tr_summable f z1 z2 n1 ∧ z_tr_td_summable f z1 z2}

where, ROC_2d accepts a function f: $\mathbb{N} \to \mathbb{N} \to \mathbb{C}$ and n1 capturing the starting point of the outer summation of the 2D z-transform (Equation (1)), and returns a set of non-zero values of variables z1 and z2 for which the 2D z-transform of f exists. It is necessary to specify the associated ROC_2d to compute the 2D z-transform. Moreover, the functions z_tr_summable and z_tr_td_summable capture the summability of the function f for the inner and the outer (double) summations, respectively, and are formalized in HOL Light as follows:

Definition 9 Summability of Function for Inner Summation
$\vdash_{def}$ ∀f z n1. z_tr_summable f z1 z2 n1 =
(∀n1. summable (from 0) (λn2. f n1 n2 / (z1 pow n1 ∗ z2 pow n2)))

Definition 10 Summability of Function for Outer (Double) Summation
$\vdash_{def}$ ∀f z1 z2. z_tr_td_summable f z1 z2 = summable (from 0)
(λn1. inftsumm (from 0) (λn2. f n1 n2 / (z1 pow n1 ∗ z2 pow n2)))

Moreover, we verify two fundamental properties of ROC, such as the linearity of the ROC and scaling of the ROC, which are quite helpful for formally verifying the classical properties of the 2D z-transform in Section *Formal Verification of the Classical Properties of the 2D z-Transform*.

Theorem 2 Linearity of ROC

$\vdash_{thm}$ ∀z1 z2 a b f g n1.
 [A1]: (z1, z2) IN ROC_2d f n1 ∧ [A2]: (z1, z2) IN ROC_2d g n1
 ⇒ (z1, z2) IN ROC_2d (λn1 n2. a ∗ f n1 n2) n1 INTER
 ROC_2d (λn1 n2. b ∗ g n1 n2) n1

Theorem 3 Scaling of ROC

$\vdash_{thm}$ ∀z1 z2 a f n1. [A]: (z1, z2) IN ROC_2d f n1
 ⇒ (z1, z2) IN ROC_2d (λn1 n2. f n1 n2 / a) n1

Theorem 2 ensures that if (z1, z2) is inside ROC_2d f n1 and ROC_2d g n1 for functions f and g then it is also inside the intersection of both ROCs for the scaled version of these functions. Similarly, Theorem 3 provides the scaling property with respect to the division by a complex number a.

Formal Verification of the Classical Properties of the 2D z-Transform

We use Definitions 7 and 8, and Theorems 2 and 3 for verifying some of the classical properties of the 2D z-transform in HOL Light. This verification plays a vital role in reducing the effort required for analyzing image processing systems, as described later in Sections *Formal Verification of a (L_1, L_2)-Order 2D Infinite Impulse Response (IIR) Image Processing Filter* and *Formal Verification of a Second-Order 2D Image Processing Filter*.

Linearity of the 2D z-Transform: The linearity of the 2D z-transform is mainly used in decomposing complex (larger) systems to subsystems or combining smaller systems to larger ones having different scaling inputs. It can be mathematically expressed as:

If $\mathcal{Z}[f(n1, n2)] = F(z_1, z_2)$ and $\mathcal{Z}[g(n_1, n_2)] = G(z_1, z_2)$ then the following holds:

$$\mathcal{Z}[\alpha * f(n_1, n_2) \pm \beta * g(n_1, n_2)] = \alpha * F(z_1, z_2) \pm \beta * G(z_1, z_2) \quad (3)$$

The 2D z-transform of a linear combination of 2D sequences (or arrays) is equal to the linear combination of the 2D z-transform of the individual arrays. We verify linearity property in HOL Light as:

Theorem 4 Linearity of the 2D z-Transform

$\vdash_{def}$ ∀f g z1 z2 a b n1.
 [A1]: (z1, z2) IN ROC_2d f n1 ∧ [A2]: (z1, z2) IN ROC_2d g n1
 ⇒ z_transform_2d (λn1 n2. a ∗ f n1 n2 ± b ∗ g n1 n2) z1 z2 =
 a ∗ z_transform_2d f z1 z2 ± b ∗ z_transform_2d g z1 z2

where a: $\mathbb{C}$ and b: $\mathbb{C}$ are arbitrary complex constants. Assumptions A1 and A2 capture the regions of the convergence of functions f and g, respectively. The proof of the above theorem is mainly based on Theorem 2 and the linearity of the infinite summation alongwith some complex arithmetic reasoning.

Shifting Property of the 2D z-Transform: The shifting property of the 2D z-transform is mostly used for analyzing the 2D Linear Shift Invariant (LSI) systems. In particular, it is used to solve the difference equations capturing the dynamics of these systems. The shifting property expresses the transform of the shifted signal $f(n_1 - m_1, n_2 - m_2)$ in terms of its 2D z-transform $F(z_1, z_2)$.

If $\mathcal{Z}[f(n_1, n_2)] = F(z_1, z_2)$ and assuming $f(-n_1, n_2) = 0$, $f(n_1, -n_2) = 0$ and $f(-n_1, -n_2) = 0$, $\forall n_1 = 1, 2, ..., m_1$ and $\forall n_2 = 1, 2, ..., m_2$, i.e., $f(n_1, n_2)$ is non-zero in the first quadrant only, then the shifting of a 2D array is mathematically expressed as follows:

$$\mathcal{Z}[f(n_1 - m_1, n_2 - m_2)] = z_1^{-m_1} * z_2^{-m_2} * F(z_1, z_2) \tag{4}$$

We formally verify the above property in HOL Light as:

Theorem 5 Shifting in Time Domain
$\vdash_{thm}$ ∀f z1 z2 m1 m2 n1.
  [A1]: (z1, z2) IN ROC_2d f n1 ∧ [A2]: in_fst_quad_2d f
   ⇒ z_transform_2d (λn1 n2. f (n1 - m1 n2 - m2)) z1 z2 =
    z_transform_2d f z1 z2 / (z1 pow m1 ∗ z2 pow m2)

where the function in_fst_quad_2d ensures that the function f is non-zero in the first quadrant only and is formalized in a relational form, i.e., f (n1 - m1, n2 - m2), ∀m1 m2. m1 < n1, m2 < n2. The verification of Theorem 5 is mainly based on the properties of complex numbers alongwith two properties regarding the negative offset of series and infinite summation. More details about the proof process of this theorem can be found in our proof script[4].

Scaling in (z_1, z_2)-domain Property of the 2D z-Transform: The scaling property of the 2D z-transform results in shrinking or expansion of the (z_1, z_2)-domain, i.e., 4D complex (z_1, z_2)-plane. If $\mathcal{Z}[f(n_1, n_2)] = F(z_1, z_2)$, then two different types of scaling are defined as:

$$\mathcal{Z}[{h_1}^{n_1} {h_2}^{n_2} f(n_1, n_2)] = F\,\left(\frac{z_1}{h_1}\right)\,\left(\frac{z_2}{h_2}\right) \tag{5}$$

$$\mathcal{Z}[{w_1}^{-n_1} {w_2}^{-n_2} f(n_1, n_2)] = F\,(w_1 z_1)\,(w_2 z_2) \tag{6}$$

If h_1 and h_2 are positive real numbers, then the scaling is interpreted as expansion of the 4D complex (z_1, z_2)-plane. On the other hand, multiplication by ${w_1}^{-n_1}$ and ${w_2}^{-n_2}$ (Equation (6)) shrinks the (z_1, z_2)-domain.

We verify the above theorems in HOL Light as:

[4] https://github.com/adrashid/fa2Dipfholtp

Theorem 6 Scaling in (z_1, z_2)-Domain (Positive/Expansion)

```
⊢thm ∀f z1 z2 n1 h1 h2.
    [A1]: (inv h1 ∗ z1, inv h2 ∗ z2) IN ROC_2d f n1 ∧
    [A2]: (z1, z2) IN ROC_2d f n1
      ⇒ z_transform_2d (λn1 n2. h1 pow n1 ∗ h2 pow n2 ∗ f n1 n2) z1 z2 =
        z_transform_2d f (inv h1 ∗ z1, inv h2 ∗ z2)
```

Theorem 7 Scaling in (z_1, z_2)-Domain (Negative/Shrinking)

```
⊢thm ∀f z1 z2 n1 w1 w2.
    [A1]: (w1 ∗ z1, w2 ∗ z2) IN ROC_2d f n1 ∧
    [A2]: (z1, z2) IN ROC_2d f n1
  ⇒ z_transform_2d (λn1 n2. w1 pow (-n1) ∗ w2 pow (-n2) ∗ f n1 n2) z1 z2 =
      z_transform_2d f (w1 ∗ z1) (w2 ∗ z2)
```

Complex Conjugation Property of the 2D z-Transform: The complex conjugation property facilitates an easy manipulation of the 2D z-transform of conjugated arrays. It is mathematically expressed as follows:

$$\mathcal{Z}[f^*(n_1, n_2)] = F^*({z_1}^*, {z_2}^*) \tag{7}$$

where $f^*(n_1, n_2)$ represents the complex conjugate of an array $f(n_1, n_2)$. The corresponding formalization of the complex conjugation property in HOL Light is given as follows:

Theorem 8 Complex Conjugation

```
⊢thm ∀f z1 z2 n1. [A]: (cnj z1, cnj z2) IN ROC_2d f n1
      ⇒ z_transform_2d (λn1 n2. cnj (f n1 n2)) z1 z2 =
            cnj (z_transform_2d f (cnj z1, cnj z2))
```

Formal Verification of a (L_1, L_2)-Order 2D Infinite Impulse Response (IIR) Image Processing Filter

2D digital filters [1] are integral components of the image processing systems. Their main responsibility includes the decomposition of an image to multiple frequency bands, restricting a 2D array/signal to a certain frequency band and providing the input-output relationship of these systems. For example, a low-pass filter allows a range of frequencies less than a certain threshold [2]. The analysis of an image processing filter mainly involves developing its mathematical model using a 2D difference equation. The next step is to apply 2D z-transform on both sides of the difference equation. Finally, the definition and the classical properties of the 2D z-transform are used to perform transfer function based analysis of the given filter.

The impulse response of a discrete-time system captures its behaviour for the scenario when dirac-delta function is acting as an input array [2]. 2D image processing Infinite Impulse Response (IIR) filters have a non-zero impulse response function over an infinite length of time. For these filters, the present output depends on the present input and all previously computed input and output values.

Mathematically, the 2D image processing filters are described using the following difference equation [16].

$$y(n_1,n_2) = \sum_{l_1=0}^{L_1-1}\sum_{l_2=0}^{L_2-1} \alpha(l_1,l_2)x(n_1-l_1,n_2-l_2) \\ - \sum_{k_1=1}^{K_1-1}\sum_{k_2=1}^{K_2-1} \beta(k_1,k_2)y(n_1-k_1,n_2-k_2) \tag{8}$$

where $\alpha(l_1,l_2)$ and $\beta(k_1,k_2)$ are input and output coefficients, respectively. The output array $y(n_1,n_2)$ is a linear combination of the previous K_1-1 and K_2-1 output samples, the present input $x(n_1,n_2)$), and L_1-1 and L_2-1 previous input samples. Moreover, for the shift-invariant filter, $\alpha(l_1,l_2)$ and $\beta(k_1,k_2)$ are the complex constants ($\mathbb{C}$). Therefore, Equation (8) is known as a Linear Constant Coefficient Difference Equation (LCCDE). The 2D z-transform of a $(L_1,L_2)^{th}$ difference represented in the form of $f(n_1,n_2)$ is given as:

$$\mathcal{Z}\left[\sum_{l_1=0}^{L_1-1}\sum_{l_2=0}^{L_2-1} \alpha(l_1,l_2)f(n_1-l_1,n_2-l_2)\right] = F(z_1,z_2)\sum_{l_1=0}^{L_1-1}\sum_{l_2=0}^{L_2-1} \alpha(l_1,l_2){z_1}^{-l_1}{z_2}^{-l_2} \tag{9}$$

The corresponding transfer function of the 2D IIR filter is mathematically expressed as [16]:

$$H(z_1,z_2) = \frac{Y(z_1,z_2)}{X(z_1,z_2)} = \frac{\sum_{l_1=0}^{L_1-1}\sum_{l_2=0}^{L_2-1} \alpha(l_1,l_2){z_1}^{-l_1}{z_2}^{-l_2}}{\sum_{k_1=0}^{K_1-1}\sum_{k_2=0}^{K_2-1} \beta(k_1,k_2){z_1}^{-k_1}{z_2}^{-k_2}} \tag{10}$$

To formally verify the transfer function of the 2D filter (Equation (10)), we formalize the $(L_1,L_2)^{th}$ difference as follows:

Definition 11 Formalization of the $(L_1,L_2)^{th}$ Difference
⊢*def* ∀f c L1 L2 n1 n2.
 l1l2th_difference f c L1 L2 n1 n2 =
  vecsum (0..L1) (λl1. vecsum (0..L2) (λl2. c l1 l2 ∗ f (n1 - l1) (n2 - l2)))

The function l1l2th_difference accepts a function f: ℕ → ℕ → ℂ, coefficients of the difference equation c l1 l2, the order (L1, L2) of the 2D difference equation and the variables n1 and n2, and returns the $(L_1,L_2)^{th}$ difference. It uses the function vsum s f twice to capture the double summation.

Next, we formalize a general LCCDE (Equation (8)) as follows:

Definition 12 Formalization of the LCCDE
⊢*def* ∀y x M1 M2 N1 N2 n1 n2 a b. LCCDE x y a b M1 M2 N1 N2 n1 n2 ⇔
 y (n1, n2) = l1l2th_difference y a M1 M2 n1 n2 - l1l2th_difference x b N1 N2 n1 n2

Next, we verify the 2D z-transform of the $(L_1,L_2)^{th}$ difference (Equation (9)) as:

Theorem 9 The 2D z-Transform of the $(L_1, L_2)^{th}$ Difference
$\vdash_{thm}$ ∀f c L1 L2 z1 z2 n1.
[A1]: (z1, z2) IN ROC_2d f n1 ∧
[A2]: in_fst_quad_2d f
⇒ z_transform_2d (λn1 n2. l1l2th_difference f c L1 L2 n1 n2) z1 z2 =
z_transform_2d f z1 z2 ∗ vecsum (0..L1)
(λl2. vecsum (0..L2) (λl1. z1 cpow - Cx (&l1) ∗ z2 cpow - Cx (&l2) ∗ c l1 l2))

where Assumption A1 ensures that (z_1, z_2) are in the region of convergence of the function f. Assumption A2 implies that the function f is in the first quadrant. Finally, the conclusion provides the 2D z-transform of the $(L_1, L_2)^{th}$ difference. The verification of the above theorem is mainly based on induction on N1 and N2, and Theorems 2 and 4 alongwith the following lemma about the summability of $(L_1, L_2)^{th}$ difference equation.

Lemma 1 Summability of the $(L_1, L_2)^{th}$ Difference
$\vdash_{thm}$ ∀f c L1 L2 z1 z2 n1.
[A1]: (z1, z2) IN ROC_2d f n1 ∧
[A2]: in_fst_quad_2d f
⇒ (z1, z2) IN ROC_2d (λn1 n2. l1l2th_difference f c L1 L2 n1 n2) n1

To verify the transfer function of the 2D filter (Equation (10)), we have to ensure that the 2D input and output arrays exist in the first quadrant only. Moreover, the denominator of Equation (10) should be non-zero. We formalize both these requirements in HOL Light as follows:

Definition 13 First Quadrant Input and Output 2D Arrays for LCCDE
$\vdash_{def}$ in_fst_quad_2d_lccde x y ⇔ in_fst_quad_2d x ∧ in_fst_quad_2d y

Definition 14 ROC LCCDE
$\vdash_{def}$ ∀ x y K1 K2 lst n1 ROC_2d_LCCDE x y K1 K2 lst n1 =
(ROC_2d x n1) INTER (ROC_2d y n1) DIFF
{(z1, z2) | vecsum (0..K1) (λk2. vecsum (0..K2)
(λk1. z1 cpow - Cx (&k1) ∗ z2 cpow - Cx (&k2) ∗ EL k1 lst)) = Cx (&0)} DIFF
{(z1, z2) | z_transform_2d x z1 z2 = Cx (&0)}

where, the function in_fst_quad_2d_lccde (Definition 13) accepts the input and output 2D arrays x and y and asserts the first quadrant condition for both arrays. Similarly, ROC_2d_LCCDE (Definition 14) provides the ROC of the input and output 2D arrays. It uses the HOL Light function DIFF to exclude all values of the denominator, where the transfer function of the 2D IIR filter becomes undefined.

Now, we provide the formal verification of the transfer function of a 2D IIR filter in HOL Light as follows:

Theorem 10 Transfer Function of a 2D IIR Filter
$\vdash_{thm}$ ∀x y a b L1 L2 K1 K2 z1 z2 n1.

```
[A1]: (z1, z2) IN ROC_2d_LCCDE x y K1 K2 blst n1 ∧
[A2]: in_fst_quad_2d_lccde x y ∧
[A3]: (∀n1 n2. LCCDE x y a b L1 L2 K1 K2 n1 n2)
⇒ z_transform_2d y z1 z2 / z_transform_2d x z1 z2 =
        vecsum (0..K1) (λk2. vecsum (0..K2)
             (λk1. z1 cpow - Cx (&k1) ∗ z2 cpow - Cx (&k2) ∗ a k1 k2)) /
        vecsum (0..L1) (λl2. vecsum (0..L2)
             (λl1. z1 cpow - Cx (&l1) ∗ z2 cpow - Cx (&l2) ∗ b l1 l2))
```

Assumption A1 provides the ROC for LCCDE. Assumption A2 ensures that the input and output 2D arrays are in the first quadrant. Assumption A3 captures the time-domain model of the 2D IIR filter, i.e., the LCCDE (Equation (8)). Finally, the conclusion presents the transfer function of the 2D IIR filter. The proof process of the above theorem is based on the linearity and shifting properties of the 2D z-transform (Theorems 4 and 5), summability of the $(L_1, L_2)^{th}$ difference (Lemma 1) alongwith some complex arithmetic reasoning. Theorem 10 provides the transfer function of a generic 2D IIR image processing filter and is quite useful in the verification of the second-order 2D medical image processing filter described in Section *Formal Verification of a Second-Order 2D Image Processing Filter*.

Formal Verification of a Second-Order 2D Image Processing Filter

To illustrate the practical utilization and effectiveness of the proposed formalization of the 2D z-transform, we apply it to formally analyze a second-order image processing filter that is widely used for performing various tasks, such as noise removal [1], image smoothing [2] and quality enhancement [5].

A second-order image processing filter is graphically represented by the flowgraph given in Figure 2. A flowgraph is a collection of branches (directed connections) and nodes (input and output 2D arrays), where nodes are connected using branches. The constants c_{01}, c_{10}, c_{11}, c_{02}, c_{12}, c_{20}, c_{21} and c_{22} in Figure 2 represent the gains of each branches. Whereas, ${z_1}^{-1}$ and ${z_2}^{-1}$ present the shift right (horizontal delay) and shift up (vertical delay) operations, respectively. We can mathematically describe this filter using the following linear difference equation.

$$y(n_1, n_2) = x(n_1, n_2) + \sum_{\substack{k_1=0 \\ (k_1,k_2)\neq 0}}^{2} \sum_{k_2=0}^{2} c_{k_1 k_2} y(n_1 - k_1, n_2 - k_2), \tag{11}$$

Alternatively, Equation (11) can be represented as:

$$\begin{aligned} y(n_1, n_2) =& x(n_1, n_2) + c_{01} y(n_1, n_2 - 1) + c_{10} y(n_1 - 1, n_2) + \\ & c_{11} y(n_1 - 1, n_2 - 1) + c_{02} y(n_1, n_2 - 2) + \\ & c_{12} y(n_1 - 1, n_2 - 2) + c_{20} y(n_1 - 2, n_2) + \\ & c_{21} y(n_1 - 2, n_2 - 1) + c_{22} y(n_1 - 2, n_2 - 2) \end{aligned} \tag{12}$$

The transfer function corresponding to the difference equation based model (Equation (11)) is given as:

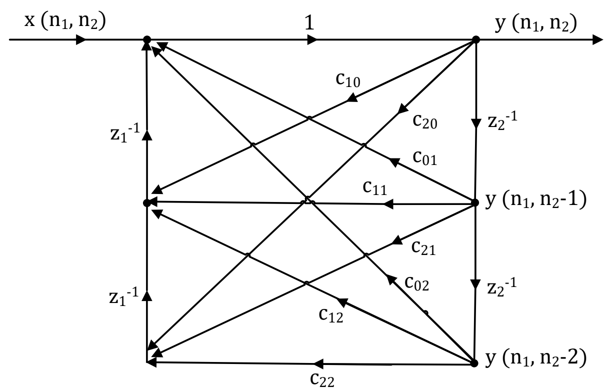

Figure 2: Flowgraph of a Second-order Image Processing Filter

$$H(z_1, z_2) = \frac{Y(z_1, z_2)}{X(z_1, z_2)} = \frac{1}{1 - \sum_{n_1=1}^{2} \sum_{n_2=1}^{2} c_{n_1 n_2} {z_1}^{-n_1} {z_2}^{-n_2}}, (k_1, k_2) \neq 0 \tag{13}$$

Alternatively, the above equation can be represented as:

$$H(z_1, z_2) = \frac{1}{\begin{array}{r} 1 - c_{01}{z_2}^{-1} - c_{10}{z_1}^{-1} - c_{11}{z_1}^{-1}{z_2}^{-1} \\ -c_{02}{z_2}^{-2} - c_{12}{z_1}^{-1}{z_2}^{-2} - c_{20}{z_1}^{-2} \\ - c_{21}{z_1}^{-2}{z_2}^{-1} - c_{22}{z_1}^{-2}{z_2}^{-2} \end{array}} \tag{14}$$

To verify the transfer function expressed in Equation (13), we need to formalize the difference equation based model of the filter (Equation (11)), which is given in HOL Light as:

Definition 15 Difference Equation Based Model of the Second-Order Filter
$\vdash_{def}$ ∀y x n1 n2 a b. second_order_filter x y a b n1 n2 ⇔
y (n1, n2) = l1l2th_difference y a 2 2 n1 n2 - l1l2th_difference x b 0 0 n1 n2

where a and b are the coefficients of input and output 2D arrays. The function second_order_filter accepts input and output 2D arrays, their coefficients a and b and returns the linear difference equation describing the second-order image processing filter.

Now, we formally verify the transfer function (Equation (13)) in HOL Light as follows:

Theorem 11 Transfer Function of a Second-Order Filter
$\vdash_{thm}$ ∀x y a b z1 z2 n1 c11 c12 c21 c22.
[A1]: (z1, z2) IN ROC_2d_LCCDE x y 2 2 b n1 ∧
[A2]: in_fst_quad_2d_lccde x y ∧
[A3]: cond_2d_diff_eq_coeff a b b01 b10 b11 b02 b12 b20 b21 b22 ∧
[A4]: ¬(z1 = Cx (&0)) ∧
[A5]: ¬(z2 = Cx (&0)) ∧

[A6]: (∀n1 n2. second_order_filter x y a b n1 n2)
⇒ z_transform_2d y z1 z2 / z_transform_2d x z1 z2 =
Cx (&1) / (Cx (&1) - b01 ∗ z2 cpow - Cx (&1) - b10 ∗ z1 cpow - Cx (&1) -
b11 ∗ z1 cpow - Cx (&1) ∗ z2 cpow - Cx (&1) - b02 ∗ z2 cpow - Cx (&2) -
b12 ∗ z1 cpow - Cx (&1) ∗ z2 cpow - Cx (&2) - b20 ∗ z1 cpow - Cx (&2) -
b21 ∗ z1 cpow - Cx (&2) ∗ z2 cpow - Cx (&1) -
b22 ∗ z1 cpow - Cx (&2) ∗ z2 cpow - Cx (&2))

Assumption A1 provides the ROC for the differential equation based model of the second-order filter. Assumption A2 ensures that the input and output 2D arrays x and y are in the first quadrant. Assumption A3 asserts that the input and output coefficients are constant. Assumptions A4 and A5 ensure that the complex variables z1 and z2 are non-zero. Assumption A6 captures the time-domain model of the second-order filter, i.e., Equation (11). Finally, the conclusion presents the transfer function of the second-order filter. The verification of the above theorem is mainly based on Theorem 10 alongwith some complex arithmetic reasoning. Theorem 11 is the formal verification result of the second-order image processing filter based on our formalization of the 2D z-transform described in Sections *Formalization of* 2*D* z*-Transform* and *Formal Verification of the Properties of the* 2*D* z*-Transform*.

Now, a specialized case of a second-order image processing filter is graphically represented by the flowgraph given in Figure 3. This filter can be mathematically represented, by setting the values of the gains of each branches as $c_{01} = c_{11} = c_{20} = c_{21} = c_{22} = 0$, $c_{10} = \frac{1}{4}$, $c_{02} = -\frac{1}{4}$ and $c_{12} = \frac{1}{4}$ in Equation (11), as follows.

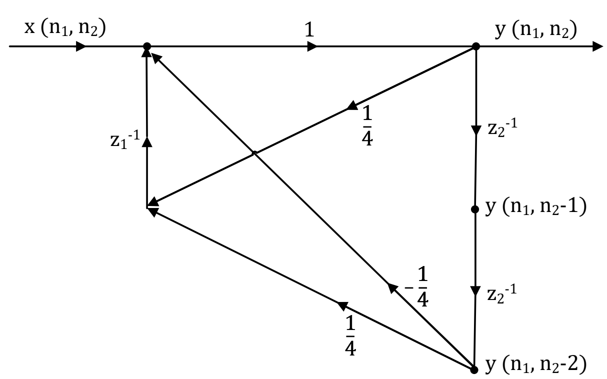

Figure 3: Flowgraph of a Specialized Second-order Image Processing Filter

$$y(n_1, n_2) = x(n_1, n_2) + \frac{1}{4}y(n_1 - 1, n_2) - \frac{1}{4}y(n_1, n_2 - 2) + \frac{1}{4}y(n_1 - 1, n_2 - 2) \quad (15)$$

The transfer function corresponding to the difference equation based model (Equation (15)) is described as:

$$H(z_1, z_2) = \frac{Y(z_1, z_2)}{X(z_1, z_2)} = \frac{1}{1 - \frac{1}{4}{z_1}^{-1} + \frac{1}{4}{z_2}^{-2} - \frac{1}{4}{z_1}^{-1}{z_2}^{-2}} \quad (16)$$

We formally verify the transfer function (Equation (16)) as:

Theorem 12 Transfer Function of a Specialized Second-Order Filter

$\vdash_{thm}$ ∀x y c d z1 z2 n1.
  [A1]: (z1, z2) IN ROC_2d_LCCDE_spec x y 2 2 b n1 ∧
  [A2]: in_fst_quad_2d_lccde_spec x y ∧
  [A3]: ¬(z1 = Cx (&0)) ∧
  [A4]: ¬(z2 = Cx (&0)) ∧
  [A5]: (∀n1 n2. second_order_filter_spec x y c d n1 n2)
 ⇒ z_transform_2d y z1 z2 / z_transform_2d x z1 z2 =
    Cx (&1) / (Cx (&1) - (1 / 4) ∗ z1 cpow - Cx (&1) +
     (1 / 4) ∗ z2 cpow - Cx (&2) -
      (1 / 4) ∗ z1 cpow - Cx (&1) ∗ z2 cpow - Cx (&2))

Assumption A1 captures the ROC for the differential equation based model of the specialized second-order filter. Assumption A2 asserts the first quadrant conditions on the input and output 2D arrays x and y. Assumptions A3 and A4 ensure that the complex variables z1 and z2 are non-zero. Assumption A5 presents the time-domain model of the specialized second-order filter, i.e., Equation (15). Finally, the conclusion captures the transfer function of the specialized second-order filter. The verification of the above theorem is done almost automatically using Theorem 11, which illustrates the effectiveness of our proposed approach.

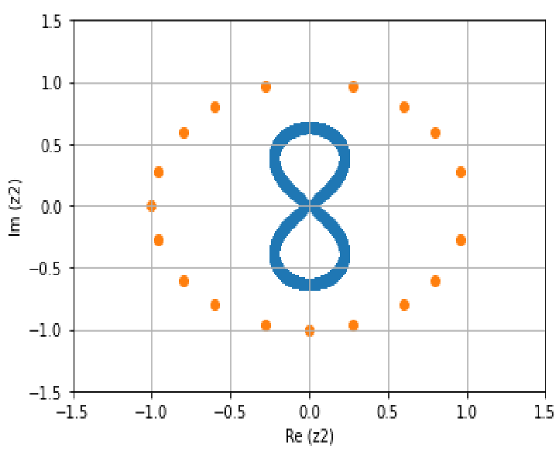

Figure 4: Root Map for the Specialized Second-order Filter

Next, we implement the transfer function of the specialized second-order filter, verified as Theorem 12, in Python. In particular, we implemented the poles (denominator of Equation (16)) of the transfer function, i.e., the characteristic equation $1-\frac{1}{4}{z_1}^{-1}+\frac{1}{4}{z_2}^{-2}-\frac{1}{4}{z_1}^{-1}{z_2}^{-2}=0$ on the complex plane z_2 for $z_1 = e^{i\omega_1}, \omega_1 \in [0, \pi]$. Figure 4 provides the root map capturing the poles of the transfer function and their placement with respect to unit circle in the complex plane can be used for analyzing the 2D stability of the corresponding system. In the case of the specialized second-order filter (Figure 4), the presence of poles inside the unit circle provide a sufficient condition for the stability of the corresponding system. However, in case of poles outside the unit circle, the corresponding system will be unstable. Similarly, the one-dimensional (1D) stability can be analyzed by implementing the characteristic equation for all z_1 with $z_2 = 1$ and observing the placement of the poles in the complex z_1 plane.

Discussions

The distinguishing feature of our proposed framework, as compared to the traditional analysis techniques is that all verified theorems are of generic nature, i.e., all of the functions and variables involved in these theorems are universally quantified and thus can be specialized based on the requirement of the analysis of the image processing filter of any order. For example, Theorem 10 provides the verification of the transfer function of a generic (L_1, L_2)-order 2D IIR image processing filter and it can be directly used for analyzing an image processing filter of any order, such as, second order filter (Theorems 11 and 12). We only need to specialize the gains ($\alpha(l_1, l_2)$, $\beta(k_1, k_2)$ in Equations (8), (9) and (10)) of an image processing filter based on a particular scenario. Whereas, in the case of computer based simulations, we need to model each filter based on its corresponding order, individually that can add a lot of complexity for the case of higher-order filters. Thus, the generic nature of the formalized theorems in our proposed approach makes it better than the transitional analysis methods. Another advantage of our proposed approach is the inherent soundness of the theorem proving technique. It ensures that all the required assumptions are explicitly present alongwith the theorem, which are often ignored in the conventional simulations based analysis and their absence may affect the accuracy of the corresponding analysis. For example, for a given system (second-order image processing filter), if we do not incorporate the constraints captured as Assumptions `A3`, `A4` and `A5` of Theorem 11 and Assumptions `A3` and `A4` of Theorem 12, it may lead to some undesired results, such as, it may result into a transfer function that can make a stable system as an unstable system. One of the main limitations of the proposed approach is the significant user involvement in the proposed formalization of z-transform, due to the undecidable nature of the higher-order logic. However, we have developed simplifiers, such as ROC_SIMP_TAC, DIFF_EQ_SIMP_TAC and TRANS_FUN_TAC that significantly reduce the user guidance in the reasoning process. More details of the proof process can be viewed in our proof script[5].

Conclusions

2D image processing systems include processing of the images, such as image filtering, enhancement, compression and restoration. These systems are typically analyzed using the 2D z-transform. This paper proposed a HOL theorem proving based framework for formally analyzing 2D image processing filters. In particular, we formalized the 2D z-transform and formally verified its various classical properties, such as linearity, shifting in time, scaling in (z_1, z_2)-domain and complex conjugation. Moreover, we formally analyzed a generic 2D IIR image processing filter. Finally, to demonstrate the practical utilization and effectiveness of the proposed framework, we presented the formal analysis of a second-order image processing filter.

[5] https://github.com/adrashid/fa2Dipfholtp

In future, we aim to formalize the 2D inverse z-transform [16] that will enable us to find the time-domain solutions of the time-domain models of the image processing systems. Another future direction is to formalize the 2D convolution [2] that can greatly simplify the reasoning about systems-of-systems [16].

Author details
[1]School of Electrical Engineering and Computer Science (SEECS), National University of Sciences and Technology (NUST), Islamabad, Pakistan. [2]Computer Engineering Department, College of Engineering and Petroleum, Kuwait University, Kuwait City, Kuwait.

References

1. Lim, J.S.: Two-dimensional Signal and Image Processing. Englewood Cliffs (1990)
2. Woods, J.W.: Multidimensional Signal, Image, and Video Processing and Coding. Elsevier, ??? (2006)
3. Hussain, R., Zeadally, S.: Autonomous Cars: Research Results, Issues, and Future Challenges. IEEE Communications Surveys & Tutorials **21**(2), 1275–1313 (2018)
4. Blasinski, H., Farrell, J., Lian, T., Liu, Z., Wandell, B.: Optimizing Image Acquisition Systems for Autonomous Driving. Electronic Imaging **2018**(5), 161–1 (2018)
5. Behrenbruch, C., Petroudi, S., Bond, S., Declerck, J., Leong, F., Brady, J.: Image Filtering Techniques for Medical Image Post-processing: An Overview. The British Journal of Radiology **77**(suppl_2), 126–132 (2004)
6. Hemalatha, G., Sumathi, C.: Preprocessing Techniques of Facial Image with Median and Gabor Filters. In: Information Communication and Embedded Systems, pp. 1–6 (2016). IEEE
7. Durán, A.J., Pérez, M., Varona, J.L.: The Misfortunes of a Mathematicians' Trio using Computer Algebra Systems: Can We Trust? CoRR **abs/1312.3270** (2013)
8. Hasan, O., Tahar, S.: Formal Verification Methods. Encyclopedia of Information Science and Technology, IGI Global Pub, 7162–7170 (2015)
9. Harrison, J.: Handbook of Practical Logic and Automated Reasoning. Cambridge University Press, ??? (2009)
10. Gordon, M.J.: HOL: A Proof Generating System for Higher-order Logic. In: VLSI Specification, Verification and Synthesis. SECS, vol. 35, pp. 73–128. Springer, ??? (1988)
11. Harrison, J.: HOL Light: A Tutorial Introduction. In: Formal Methods in Computer-Aided Design. LNCS, vol. 1166, pp. 265–269 (1996). Springer
12. Harrison, J.: HOL Light: A Tutorial Introduction. In: Srivas, M., Camilleri, A. (eds.) Proceedings of the First International Conference on Formal Methods in Computer-Aided Design (FMCAD'96). Lecture Notes in Computer Science, vol. 1166, pp. 265–269. Springer, ??? (1996)
13. Paulson, L.: ML for the Working Programmer. Cambridge University Press, ??? (1996)
14. Siddique, U., Mahmoud, M.Y., Tahar, S.: On the Formalization of Z-Transform in HOL. In: Interactive Theorem Proving, pp. 483–498 (2014). Springer
15. Taqdees, S.H., Hasan, O.: Formalization of Laplace Transform using the Multivariable Calculus Theory of HOL Light. In: Logic for Programming Artificial Intelligence and Reasoning, pp. 744–758 (2013). Springer
16. Dudgeon, D.E.: Multidimensional Digital Signal Processing. Engewood Cliffs (1983)

Availability of data and materials
All data generated or analyzed during this study are included in this paper.

Abbreviations
2**D**: Two-dimensional
CCTV: Closed-circuit Television
IIR: Infinite Impulse Response
LCCDE: Linear Constant Coefficient Difference Equation

Competing interests
The authors declare that they have no competing interests.

Acknowledgements
This work was supported and funded by Kuwait University, Research Project No. (EO $07/19$).

Funding
This work was funded by Kuwait University, Research Project No. (EO $07/19$).

Contributions
The research and results of this paper are mainly compiled by the first author. All authors read and approved the final manuscript.

Author Information
Adnan Rashid received his PhD degree in Information Technology from School of Electrical Engineering and Computer Science (SEECS), National University of Science and Technology (NUST), Islamabad, Pakistan, in 2019. Prior to this, he received the M.Sc. and M.Phil. degrees in Electronics from the Department of Electronics, Quaid-i-Azam University (QAU), Islamabad, Pakistan, in 2008 and 2012, respectively. He worked as a postdoctoral fellow at the System Analysis and Verification (SAVe) laboratory of NUST for one year till March 2020. He has also worked as a Visiting Researcher at Hardware Verification Group (HVG), Concordia University, Canada in 2018. Currently, he is an Assistant Professor at SEECS. NUST, Islamabad, Pakistan. He has a strong interest in Formal Methods, with their applications in Control Systems, Analog Circuits, Biological Systems, Robotic Systems, Cell

injection Systems, Communication Systems and Transportation Systems. He has served as a chair of the Doctoral program at Conference on Intelligent Computer Mathematics, Edinburgh, UK, in 2017.
Sa'ed Abed received his B.Sc. and M.Sc. in Computer Engineering from Jordan University of Science and Technology, Jordan in 1994 and 1996, respectively. In 2008, he received his Ph.D. in Computer Engineering from Concordia University, Canada. He has previously worked at Hashemite University, Jordan, as an Assistant Professor from 2008-2014. Currently, he is an Associate Professor in Computer Engineering Department at Kuwait University. His research interests include Formal Methods, SAT Solvers and VLSI Design. Dr. Abed also served as a reviewer for various international conferences and journals. He published over 90 papers in reputable journals and conferences.
Osman Hasan received his BEng (Hons) degree from the University of Engineering and Technology, Peshawar Pakistan in 1997, and the MEng and PhD degrees from Concordia University, Montreal, Quebec, Canada in 2001 and 2008, respectively. Before his PhD, he worked as an ASIC Design Engineer from 2001 to 2004 at LSI Logic. He worked as a postdoctoral fellow at the Hardware Verification Group (HVG) of Concordia University for one year until August 2009. Currently, he is a Professor at the School of Electrical Engineering and Computer Science of National University of Science and Technology (NUST), Islamabad, Pakistan. He is the founder and director of System Analysis and Verification (SAVe) Lab at NUST, which mainly focuses on the design and formal verification of energy, embedded and e-health related systems. He has received several awards and distinctions, including the Pakistan's Higher Education Commission's Best University Teacher (2010) and Best Young Researcher Award (2011) and the President's gold medal for the best teacher of the University from NUST in 2015. Dr. Hasan is a senior member of IEEE, member of the ACM, Association for Automated Reasoning (AAR) and the Pakistan Engineering Council.

Ethics declarations

Ethics approval and consent to participate
All procedures performed in this paper were in accordance with the ethical standards of research community.

Consent for publication
Not applicable.

Competing interests
The authors declare that they have no competing interests.