Neutrosophic Sets and Systems, Vol. 43, 2021
University of New Mexico

On Neutrosophic Crisp Sets and Neutrosophic Crisp Mathematical Morphology

Xiaohong Zhang [1,*], Mengyuan Li [1] and Tao Lei [2]

[1] School of Arts and Sciences, Shaanxi University of Science and Technology, Xi'an 710021; zhangxiaohong@sust.edu.cn // 201706060120@sust.edu.cn

[2] School of Electronic Information and Artificial Intelligence, Shaanxi University of Science and Technology, Xi'an 710021; leitao@sust.edu.cn

* Correspondence: zhangxiaohong@sust.edu.cn; Tel.: +086(optional; include country code)

Abstract: In this paper, we analyze the basic algebraic operations of neutrosophic crisp sets and their properties, show that some operations of neutrosophic crisp sets with type 2 and type 3 are not closed by counter examples, and give some new operations. Then, discuss some morphological operators in neutrosophic crisp mathematical morphology, and study the application to edge extraction of color images, and give some Python programs and related experimental results.

Keywords: Fuzzy Set; Neutrosophic Set; Neutrosophic Crisp Set; Mathematical Morphology; Neutrosophic Crisp Mathematical Morphology

1. Introduction and Preliminaries

The theory of neutrosophic set is established by F. Smarandache, and it is applied to many areas such as non-classical logics, decision science, image processing, algebraic systems and so on [1–8]. In 2014, A.A.Salama and F.Smarandache introduced the new concept of neutrosophic crisp set, and studied the basic operations of neutrosophic crisp sets [9, 10]. On the other hand, mathematical morphology is a branch of image processing, which arose in 1964 [11, 12], and it is generalized to fuzzy mathematical morphology [13]. In 2017, the two research areas are associated, and the new notion of neutrosophic crisp mathematical morphology is firstly proposed [14]. Then, some new articles on this research direction are published [15, 16, 17]. This paper will carry out exploratory research along this direction, mainly discussing the operation and properties of neutrosophic crisp sets, and studying the application of neutrosophic crisp mathematical morphology in color image processing (previously only applied research on binary images and gray images).

At first, let's review some basic concepts in neutrosophic crisp set and neutrosophic crisp mathematical morphology.

Definition 1.1 ([9, 10]) Let X be a non-empty fixed sample space. A neutrosophic crisp set (NCS for short) A is an object having the form $\langle A_1, A_2, A_3\rangle$ where A_1, A_2 and A_3 are subsets of X.

Remark 1.1. In this paper, the set of all neutrosophic crisp sets of X will be denoted NCS(X).

Definition 1.2 ([9, 10]) The object having the form $A=\langle A_1, A_2, A_3\rangle$ is called:
(a) A neutrosophic crisp set of Type 1 (NCS-Type 1) if satisfying

$$A_1\cap A_2=\varnothing,\ A_1\cap A_3=\varnothing \text{ and } A_2\cap A_3=\varnothing.$$

(b) A neutrosophic crisp set of Type 2 (NCS-Type 2) if satisfying

◎ ◎◎ $A_1\cap A_2=\varnothing$, $A_1\cap A_3=\varnothing$, $A_2\cap A_3=\varnothing$, and $A_1\cup A_2\cup A_3=X$.

(c) A neutrosophic crisp set of Type 3 (NCS-Type 3) if satisfying

$A_1\cap A_2\cap A_3=\varnothing$ and $A_1\cup A_2\cup A_3=X$.

Remark 1.2. In this paper, the set of all neutrosophic crisp sets of Type 1, Type 2 and Type 3 of X will be denoted NCS1(X), NCS2(X) and NCS3(X), respectively.

Definition 1.3 ([9, 10]) Let $A=\langle A_1, A_2, A_3\rangle$ be a NCS in X, then the complement of the set A may be defined as three kinds of complements:

(C1) $A^{C1}=\langle A_1{}^C, A_2{}^C, A_3{}^C\rangle$, where $A_1{}^C$, $A_2{}^C$ and $A_3{}^C$ are the complement of the set $A_1{}^C$, $A_2{}^C$ and A_3;

(C2) $A^{C2}=\langle A_3, A_2, A_1\rangle$;

(C3) $A^{C3}=\langle A_3, A_2{}^C, A_1\rangle$, where $A_2{}^C$ is the complement of the set A_2.

Definition 1.4 ([9, 10]) Let X be a non-empty set, and the NCSs A and B be in the form $A=\langle A_1, A_2, A_3\rangle$, $B=\langle B_1, B_2, B_3\rangle$. The inclusion relation may be defined as two types:

Type 1. $A\subseteq_1 B$ if and only if $A_1\subseteq B_1$, $A_2\subseteq B_2$ and $A_3\supseteq B_3$;

Type 2. $A\subseteq_2 B$ if and only if $A_1\subseteq B_1$, $A_2\supseteq B_2$ and $A_3\supseteq B_3$.

Definition 1.5 ([9, 10, 14]) Let X be a non-empty set, and the NCSs A and B be of the form $A=\langle A_1, A_2, A_3\rangle$, $B=\langle B_1, B_2, B_3\rangle$. Then

(1) the intersection of A and B may be defined as two types:

Type 1. $A\cap_1 B = \langle A_1\cap B_1, A_2\cap B_2, A_3\cup B_3\rangle$;

Type 2. $A\cap_2 B = \langle A_1\cap B_1, A_2\cup B_2, A_3\cup B_3\rangle$.

(2) the union of A and B may be defined as two types:

Type 1. $A\cup_1 B = \langle A_1\cup B_1, A_2\cup B_2, A_3\cap B_3\rangle$;

Type 2. $A\cup_2 B = \langle A_1\cup B_1, A_2\cap B_2, A_3\cap B_3\rangle$.

Remark 1.3. In the papers [9, 10], the union with type 1 of A and B is written by $A\cup_1 B = \langle A_1\cup B_1, A_2\cap B_2, A_3\cup B_3\rangle$. We think that this is a typographical error. In [14], the operation has been changed to the above definition.

Definition 1.6 ([9, 10]) Let X be a non-empty set. Then

(1) φ_N may be defined as the following four types:

(a) Type 1: $\varphi_{N1}=\langle\varnothing, \varnothing, X\rangle$,

(b) Type 2: $\varphi_{N2}=\langle\varnothing, X, X\rangle$,

(c) Type 3: $\varphi_{N3}=\langle\varnothing, X, \varnothing\rangle$,

(d) Type 4: $\varphi_{N4}=\langle\varnothing, \varnothing, \varnothing\rangle$.

(2) X_N may be defined as the following four types:

(a) Type 1: $X_{N1}=\langle X, \varnothing, \varnothing\rangle$,

(b) Type 2: $X_{N2}=\langle X, X, \varnothing\rangle$,

(c) Type 3: $X_{N3}=\langle X, \varnothing, X\rangle$,

(d) Type 4: $X_{N4}=\langle X, X, X\rangle$.

Now, we introduce some basic concepts in mathematical morphology. Consider the space $E = \mathbf{R}^n$ or $\boldsymbol{Z}^n$, with origin $\boldsymbol{O} = (0, ..., 0)$. Given $A\subseteq E$, the complement of $A\subseteq E$ is $A^C = E\backslash A$, and the transpose or symmetrical of A is $A^\vee = \{-x \mid x\in A\}$. For every $p\in E$, the translation by p is the map $E\to E$: $x\to x+p$; it transforms any subset A of E into its translate by p, $A_p = \{x+p \mid x\in A\}$. Most morphological operations on sets can be obtained by combining set-theoretical operations with two basic operators, dilation and erosion. The latter arise from two set-theoretical operations, the Minkowski addition $\oplus$ (Minkowski, 1903) and subtraction Θ (Hadwiger,1950), defined as follows for any A, $B\in P(E)$:

$$A\oplus B = \bigcup_{b\in B} A_b = \bigcup_{a\in A} B_a = \{a+b \mid a\in A, b\in B\}.$$

$$A\Theta B = \bigcap_{b\in B} A_{-b} = \{p\in E \mid B_p \subseteq A\}.$$

Formally speaking, A and B play similar roles as binary operands. However, in real situations, A will stand for the image (which is big, and given by the problem), and B for the structuring element (a small shape chosen by the user), so that $A\oplus B$ and $A\Theta B$ will be transformed images.

Definition 1.7 ([11]) We define the dilation by B, δ_B: $P(E)\rightarrow P(E)$; $A\rightarrow A\oplus B$, and the erosion by B, ε_B: $P(E)\rightarrow P(E)$; $A\rightarrow A\Theta B$.

It should be noted that dilation and erosion are dual by complementation, in other words dilating a set is equivalent to eroding its complement with the symmetrical structuring element:

$$(A\oplus B)^C = A^C\Theta \check{B};\ (A\Theta B)^C = A^C\oplus \check{B}.$$

Definition 1.8 ([14]) Let X = $\mathbf{R}^n$ or $\boldsymbol{Z}^n$, A, $B\in$NCS(X). Then we define two types of the neutrosophic crisp dilation as follows:

Type 1: $A\oplus_1 B=\langle A_1\oplus B_1, A_2\oplus B_2, A_3\Theta B_3\rangle$;

Type 2: $A\oplus_2 B=\langle A_1\oplus B_1, A_2\Theta B_2, A_3\Theta B_3\rangle$.

Definition 1.9 ([14]) Let X = $\mathbf{R}^n$ or $\boldsymbol{Z}^n$, A, $B\in$NCS(X). Then we define two types of the neutrosophic crisp erosion as follows:

Type 1: $A\Theta_1 B=\langle A_1\Theta B_1, A_2\Theta B_2, A_3\oplus B_3\rangle$;

Type 2: $A\Theta_2 B=\langle A_1\Theta B_1, A_2\oplus B_2, A_3\oplus B_3\rangle$.

2. On Some Operations of Neutrosophic Crisp Sets

For NCS-Type 3, the complement of $A=\langle A_1, A_2, A_3\rangle\in$NCS3($X$) of type 3 in [9] (Definition 3.5) or [10] (Definition 1.1.10) is defined as following:

$A^{C3}=\langle A_3, A_2{}^C, A_1\rangle$, where $A_2{}^C$ is the complement of the set A_2.

The following example shows that the definition above is not well, since A^{C3} may be not an NCS-Type 3.

Example 2.1 Let $X = \{a, b, c, d, e, f, g\}$, $A=\langle\{a, b, c\}, \{b, c, d, e\}, \{a, e, f, g\}\rangle$. Then A is an NCS- Type 3 in X, that is, $A\in$NCS3(X). But

$A^{C3}=\langle\{a, e, f, g\}, \{a, f, g\}, \{a, b, c\}\rangle\notin$NCS3(X), since $\{a, e, f, g\}\cap\{a, f, g\}\cap\{a, b, c\} = \{a\} \neq \varnothing$.

Moreover, the intersection and union operations of neutrosophic crisp sets are not applied to NCS-Type 2 and NCS-Type 3, since they are not closed, and some counterexamples are shown as follows.

Example 2.2 Let $X = \{a, b, c, d, e\}$, $A=\langle\{a, b\}, \{c, d\}, \{e\}\rangle$, $B=\langle\{a\}, \{b, c\}, \{d, e\}\rangle$. Then A, $B\in$NCS2(X). But

Type 1. $A\cap_1 B = \langle A_1\cap B_1, A_2\cap B_2, A_3\cup B_3\rangle = \langle\{a\}, \{c\}, \{d, e\}\rangle\notin$NCS2($X$);

Type 2. $A\cap_2 B = \langle A_1\cap B_1, A_2\cup B_2, A_3\cup B_3\rangle = \langle\{a\}, \{b, c, d\}, \{d, e\}\rangle\notin$NCS2($X$).

Example 2.3 Let $X = \{a, b, c, d, e, f\}$, $A=\langle\{a, b, f\}, \{c, d\}, \{d, e, f\}\rangle$, $B=\langle\{a, f\}, \{b, f\}, \{c, d, e\}\rangle$. Then A, $B\in$NCS3(X). But

Type 1. $A\cap_1 B = \langle A_1\cap B_1, A_2\cap B_2, A_3\cup B_3\rangle = \langle\{a, f\}, \varnothing, \{c, d, e, f\}\rangle\notin$NCS3($X$);

Type 2. $A\cap_2 B = \langle A_1\cap B_1, A_2\cup B_2, A_3\cup B_3\rangle = \langle\{a, f\}, \{b, c, d, f\}, \{c, d, e, f\}\rangle\notin$NCS3($X$).

Example 2.4 Let $X = \{a, b, c, d, e\}$, $A=\langle\{a, b\}, \{c, d\}, \{e\}\rangle$, $B=\langle\{a\}, \{b, c\}, \{d, e\}\rangle$. Then A, $B\in$NCS2(X). But

Type 1. $A\cup_1 B = \langle A_1\cup B_1, A_2\cup B_2, A_3\cap B_3\rangle = \langle\{a, b\}, \{b, c, d\}, \{e\}\rangle\notin$NCS2($X$);

Type 2. $A\cup_2 B = \langle A_1\cup B_1, A_2\cap B_2, A_3\cap B_3\rangle = \langle\{a, b\}, \{c\}, \{e\}\rangle\notin$NCS2($X$).

Example 2.5 Let $X = \{a, b, c, d, e, f\}$, $A=\langle\{a, b, f\}, \{c, d\}, \{b, e, f\}\rangle$, $B=\langle\{a, f\}, \{b, f\}, \{b, c, d, e\}\rangle$. Then A, $B\in$NCS3(X). But

Type 1. $A\cup_1 B = \langle A_1\cup B_1, A_2\cup B_2, A_3\cap B_3\rangle = \langle\{a, b, f\}, \{b, c, d, f\}, \{b, e\}\rangle\notin$NCS3($X$);

Type 2. $A\cup_2 B = \langle A_1\cup B_1, A_2\cap B_2, A_3\cap B_3\rangle = \langle\{a, b, f\}, \varnothing, \{b, e\}\rangle\notin$NCS3($X$).

In [10] (Proposition 1.1.1), the authors show that $\varphi_N \subseteq A$ and $A \subseteq X_N$ where $A \in$NCS(X). But the following examples show that this result is not true.

Example 2.6 Let $X = \{1, 2, 3, 4, 5\}$, $A=\langle\{1, 2, 3\}, \{4\}, \{5\}\rangle$. Then
$\varphi_{N1}=\langle\varnothing, \varnothing, X\rangle \not\subset_2 A$, $A \not\subset_1 X_{N1}=\langle X, \varnothing, \varnothing\rangle$;
$\varphi_{N2}=\langle\varnothing, X, X\rangle \not\subset_1 A$, $A \not\subset_1 X_{N2}=\langle X, X, \varnothing\rangle$;
$\varphi_{N3}=\langle\varnothing, X, \varnothing\rangle \not\subset_1 A$, $\varphi_{N3}=\langle\varnothing, X, \varnothing\rangle \not\subset_2 A$, $A \not\subset_1 X_{N3}=\langle X, \varnothing, X\rangle$, $A \not\subset_2 X_{N3}=\langle X, \varnothing, X\rangle$;
$\varphi_{N4}=\langle\varnothing, \varnothing, \varnothing\rangle \not\subset_1 A$, $\varphi_{N4}=\langle\varnothing, \varnothing, \varnothing\rangle \not\subset_2 A$, $A \not\subset_1 X_{N4}=\langle X, X, X\rangle$, $A \not\subset_2 X_{N4}=\langle X, X, X\rangle$.

Proposition 1.1.1 in [10] should be revised to the following result (the proof is omitted).

Proposition 2.1 Let $A=\langle A_1, A_2, A_3\rangle$ be an NCS in X, then
(1) $\varphi_{N1}=\langle\varnothing, \varnothing, X\rangle \subseteq_1 A$, and $A \subseteq_1 X_{N2}=\langle X, X, \varnothing\rangle$;
(2) $\varphi_{N2}=\langle\varnothing, X, X\rangle \subseteq_2 A$, and $A \subseteq_2 X_{N1}=\langle X, \varnothing, \varnothing\rangle$.

In [10] (Proposition 1.1.2), the authors show that De Morgan law hold for neutrosophic crisp sets. But the following examples show that this result is not true.

Example 2.7 Let $X = \{1, 2, 3, 4, 5\}$, $A=\langle\{1, 2, 3\}, \{4\}, \{5\}\rangle$, $B=\langle\{1, 2\}, \{3, 4\}, \{5\}\rangle$. Then
$(A\cap_1 B)^{C2} = \langle\{1, 2\}, \{4\}, \{5\}\rangle^{C2} = \langle\{5\}, \{4\}, \{1, 2\}\rangle$,
$A^{C2} \cup_1 B^{C2} = \langle\{5\}, \{4\}, \{1, 2, 3\}\rangle \cup_1 \langle\{5\}, \{3, 4\}, \{1, 2\}\rangle = \langle\{5\}, \{3, 4\}, \{1, 2\}\rangle \neq (A\cap_1 B)^{C2}$.

Example 2.8 Let $X = \{1, 2, 3, 4, 5\}$, $A=\langle\{1, 2, 3\}, \{4, 5\}, \{5\}\rangle$, $B=\langle\{1\}, \varnothing, \{2\}\rangle$. Then
$(A\cap_2 B)^{C1}= \langle\{1\}, \{4, 5\}, \{2, 5\}\rangle^{C1} = \langle\{2, 3, 4, 5\}, \{1, 2, 3\}, \{1, 3, 4\}\rangle$,
$A^{C1} \cup_2 B^{C1} = \langle\{4, 5\}, \{1, 2, 3\}, \{1, 2, 3, 4\}\rangle \cup_1 \langle\{2, 3, 4, 5\}, \{1, 2, 3, 4, 5\}, \{1, 3, 4, 5\}\rangle = \langle\{2, 3, 4, 5\}, \{1, 2, 3, 4, 5\}, \{1, 3, 4\}\rangle \neq (A\cap_2 B)^{C1}$.

Proposition 1.1.2 in [10] should be revised to the following assertion.

Proposition 2.2 Let X be a non-empty set, and the NCSs A and B be of the form $A=\langle A_1, A_2, A_3\rangle$, $B=\langle B_1, B_2, B_3\rangle$. Then
(1) $(A\cap_1 B)^{C1} = A^{C1} \cup_1 B^{C1}$, and $(A\cup_1 B)^{C1} = A^{C1}\cap_1 B^{C1}$;
(2) $(A\cap_1 B)^{C3} = A^{C3} \cup_1 B^{C3}$, and $(A\cup_1 B)^{C3} = A^{C3}\cap_1 B^{C3}$;
(3) $(A\cap_2 B)^{C2} = A^{C2} \cup_1 B^{C2}$, and $(A\cup_2 B)^{C2} = A^{C2}\cap_1 B^{C2}$;
(4) $(A\cap_2 B)^{C3} = A^{C3} \cup_2 B^{C3}$, and $(A\cup_2 B)^{C3} = A^{C3}\cap_2 B^{C3}$.

Proof. (1) By the definitions of the complement ^{C1}, intersection and union (see Definition 1.3 and Definition 1.5) we have
$(A\cap_1 B)^{C1}$
$= \langle A_1\cap B_1, A_2\cap B_2, A_3\cup B_3\rangle^{C1}$
$= \langle (A_1\cap B_1)^C, (A_2\cap B_2)^C, (A_3\cup B_3)^C\rangle$
$= \langle A_1^C\cup B_1^C, A_2^C\cup B_2^C, A_3^C \cap B_3^C\rangle$
$= \langle A_1^C, A_2^C, A_3^C\rangle \cup_1 \langle B_1^C, B_2^C, B_3^C\rangle$
$= \langle A_1, A_2, A_3\rangle^{C1} \cup_1 \langle B_1, B_2, B_3\rangle^{C1}$
$= A^{C1} \cup_1 B^{C1}$.
Similarly, we can get that $(A\cup_1 B)^{C1} = A^{C1}\cap_1 B^{C1}$.
(2) By the definitions of the complement ^{C3}, intersection and union (see Definition 1.3 and Definition 1.5) we have
$(A\cap_1 B)^{C3}$
$= \langle A_1\cap B_1, A_2\cap B_2, A_3\cup B_3\rangle^{C3}$
$= \langle A_3\cup B_3, (A_2\cap B_2)^C, A_1\cap B_1\rangle$
$= \langle A_3\cup B_3, A_2^C\cup B_2^C, A_1\cap B_1\rangle$
$= \langle A_3, A_2^C, A_1\rangle \cup_1 \langle B_3, B_2^C, B_1\rangle$
$= \langle A_1, A_2, A_3\rangle^{C3} \cup_1 \langle B_1, B_2, B_3\rangle^{C3}$
$= A^{C3} \cup_1 B^{C3}$.

Similarly, we can get that $(A\cup_1 B)^{C3} = A^{C3}\cap_1 B^{C3}$.

(3) By the definitions of the complement ^{C2}, intersection and union (see Definition 1.3 and Definition 1.5) we have

$(A\cap_2 B)^{C2}$
$= \langle A_1\cap B_1, A_2\cup B_2, A_3\cup B_3\rangle^{C2}$
$= \langle A_3\cup B_3, A_2\cup B_2, A_1\cap B_1\rangle$
$= \langle A_3, A_2, A_1\rangle \cup_1 \langle B_3, B_2, B_1\rangle$
$= \langle A_1, A_2, A_3\rangle^{C2} \cup_1 \langle B_1, B_2, B_3\rangle^{C2}$
$= A^{C2} \cup_1 B^{C2}$;

$(A\cup_2 B)^{C2}$
$= \langle A_1\cup B_1, A_2\cap B_2, A_3\cap B_3\rangle^{C2}$
$= \langle A_3\cap B_3, A_2\cap B_2, A_1\cup B_1\rangle$
$= \langle A_3, A_2, A_1\rangle \cap_1 \langle B_3, B_2, B_1\rangle$
$= \langle A_1, A_2, A_3\rangle^{C2} \cap_1 \langle B_1, B_2, B_3\rangle^{C2}$
$= A^{C2} \cap_1 B^{C2}$.

(4) By the definitions of the complement ^{C3}, intersection and union (see Definition 1.3 and Definition 1.5) we have

$(A\cap_2 B)^{C3}$
$= \langle A_1\cap B_1, A_2\cup B_2, A_3\cup B_3\rangle^{C3}$
$= \langle A_3\cup B_3, (A_2\cup B_2)^C, A_1\cap B_1\rangle$
$= \langle A_3\cup B_3, A_2^C\cap B_2^C, A_1\cap B_1\rangle$
$= \langle A_3, A_2^C, A_1\rangle \cup_2 \langle B_3, B_2^C, B_1\rangle$
$= \langle A_1, A_2, A_3\rangle^{C3} \cup_2 \langle B_1, B_2, B_3\rangle^{C3}$
$= A^{C3} \cup_2 B^{C3}$;

$(A\cup_2 B)^{C3}$
$= \langle A_1\cup B_1, A_2\cap B_2, A_3\cap B_3\rangle^{C3}$
$= \langle A_3\cap B_3, (A_2\cap B_2)^C, A_1\cup B_1\rangle$
$= \langle A_3\cap B_3, A_2^C\cup B_2^C, A_1\cup B_1\rangle$
$= \langle A_3, A_2^C, A_1\rangle \cap_2 \langle B_3, B_2^C, B_1\rangle$
$= \langle A_1, A_2, A_3\rangle^{C3} \cap_2 \langle B_1, B_2, B_3\rangle^{C3}$
$= A^{C3} \cap_2 B^{C3}$.

Next, we give some new operations on neutrosophic crisp sets.

Definition 2.1 [18] Let X be a non-empty set, and the NCSs A and B be of the form $A=\langle A_1, A_2, A_3\rangle$, $B=\langle B_1, B_2, B_3\rangle$. Then we can define the intersection and union with type 3 as follows:

Type 3. $A\cap_3 B = \langle A_1\cap B_1, A_2\cap B_2, A_3\cap B_3\rangle$;
Type 3. $A\cup_3 B = \langle A_1\cup B_1, A_2\cup B_2, A_3\cup B_3\rangle$.

Definition 2.2 Let X be a non-empty set, and $A=\langle A_1, A_2, A_3\rangle$, $B=\langle B_1, B_2, B_3\rangle\in$NCS2($X$) or NCS3($X$). Then we can define the intersection and union with star * as follows:

$A\cap^* B = \langle A_1\cap B_1, X-(A_1\cap B_1)\cup(A_3\cup B_3), A_3\cup B_3\rangle$;
$A\cup^* B = \langle A_1\cup B_1, X-(A_1\cup B_1)\cup(A_3\cap B_3), A_3\cap B_3\rangle$.

We can easily verify that the following asserts are true (the proofs are omitted).

Proposition 2.3 Let $A=\langle A_1, A_2, A_3\rangle$, $B=\langle B_1, B_2, B_3\rangle$ be two NCSs in X, then

(1) $(A\cap_3 B)^{C1} = A^{C1} \cup_3 B^{C1}$, and $(A\cup_3 B)^{C1} = A^{C1} \cap_1 B^{C1}$;
(2) $(A\cap_3 B)^{C2} = A^{C2} \cap_3 B^{C2}$, and $(A\cup_3 B)^{C2} = A^{C2} \cup_3 B^{C2}$.

Proposition 2.4 Let X be a non-empty set, and $A=\langle A_1, A_2, A_3\rangle$, $B=\langle B_1, B_2, B_3\rangle\in$NCS2($X$) or NCS3($X$). Then $A\cap^* B$, $A\cup^* B\in$NCS2(X) or NCS3(X), and

(1) $(A\cap^* B)^{C1} = A^{C1} \cup^* B^{C1}$, and $(A\cup^* B)^{C1} = A^{C1} \cap^* B^{C1}$;
(2) $(A\cap^* B)^{C2} = A^{C2} \cup^* B^{C2}$, and $(A\cup^* B)^{C2} = A^{C2} \cap^* B^{C2}$.

3. On Neutrosophic Crisp Mathematical Morphology and Applications

In this section, we firstly give the new definitions of neutrosophic crisp dilation and erosion, and then applies them to the edge segmentation of color images. It should be noted that the neutrosophic crisp morphology operations can only be applied to binary image processing before, and our innovative method is as follows: (1) the color image is divided into three grayscale images according to three color channels (R, G, B); (2) three grayscale images are converted to binary value images, respectively; (3) the neutrosophic crisp dilation and erosion operations are applied to them respectively (we use three kinds of operations for comparison); combine the results of the three color channels to obtain the binary value edges of the original color image.

Definition 3.1 Let $X = \mathbf{R}^n$ or $\boldsymbol{Z}^n$, A, $B\in$NCS(X). Then we define new neutrosophic crisp dilation and erosion as follows:

$A\oplus_3 B = \langle A_1\oplus B_1, A_2\oplus B_2, A_3\oplus B_3\rangle$;

$A\Theta_3 B = \langle A_1\Theta B_1, A_2\Theta B_2, A_3\Theta B_3\rangle$.

Remark 3.1. In this paper, for binary value image (as a multidimensional vector), the operations "x+y" and "x−y" will be replaced by "max{x, y}" and "min{x, 1−y}", respectively.

Now, we apply three different neutrosophic crisp morphological operators (see Definition 1.8, Definition 1.9 and Definition 3.1) to extract the edges of the color image (as shown in the figure 1).

Figure 1. The original color image.

First, the RGB three channels of the original image are separated into three grayscale images. We use Python program as follows, and the separation results are shown in Figure 2 (a), (b) and (c).

Python Program 3.1

```
from PIL import Image                      # import Image class , from PIL package
from matplotlib import pyplot as plt       # import pyplot class , rename it ply
import cv2                                 # import the cv2 package
import numpy as np                         # import the numpy package and rename it np
img1 = plt.imread('yellow_duck.jpg')       # read the picture to be used
red = img1[:,:,0]                          # get the red channel of the picture
green = img1[:,:,1]                        # get the green channel of the picture
blue = img1[:,:,2]                         # get the blue channel of the picture
```

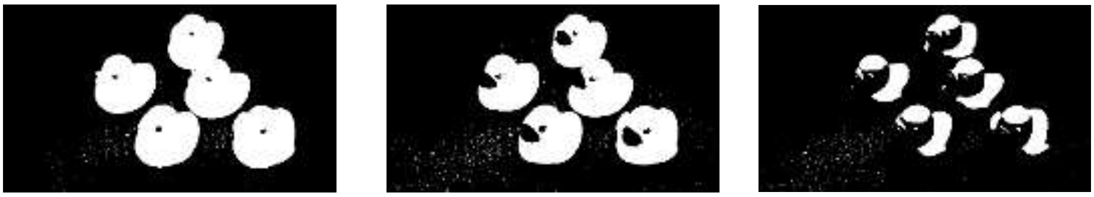
(**a**) (**b**) (**c**)

Figure 2. Separated three grayscale images: (**a**) R-channel; (**b**) G-channel; (**b**) B-channel.

Second, binarize the above three gray images to obtain three black and white images, see following Python program and Figure 3 (a), (b) and (c).

Python Program 3.2

```
# Define a function to convert grayscale image to binary image
def threshold(img, Maxvalue,choice):
    array=(cv2.THRESH_BINARY,cv2.THRESH_BINARY_INV,cv2.THRESH_TRUNC,cv2.TH
RESH_TOZERO,cv2.THRESH_TOZERO_INV,cv2.THRESH_BINARY)
    ret, binary = cv2.threshold(img, Maxvalue, 255, array[choice])
    return binary
# call threshold function to convert the grayscale images of the three channel into binary
images respectively.
A1 = threshold(red,90,0)
A2 = threshold(green,130,0)
A3 = threshold(blue,90,0)
```



(**a**) A_1 (**b**) A_2 (**c**) A_3

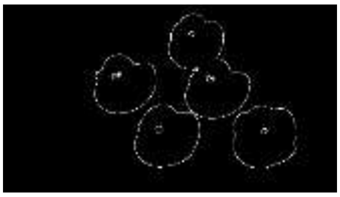
Figure 3. The binarization results of the three gray images: (**a**) A_1; (**b**) A_2; (**b**) A_3.

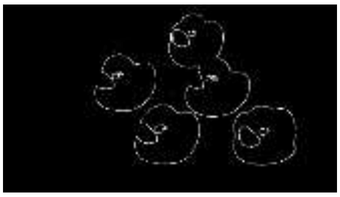
Third, by Definition 3.1, we obtain the binary value edges of the three channels, that is, putting $A=\langle A_1, A_2, A_3\rangle$, $B=\langle B_1, B_2, B_3\rangle$, where $B_1 = B_2 = B_3 = (1, 1, 1, 1, 1; 1, 1, 1, 1, 1; 1, 1, 1, 1, 1; 1, 1, 1, 1, 1; 1, 1, 1, 1, 1)$, $A\Theta_3 B=\langle A_1\Theta B_1, A_2\Theta B_2, A_3\Theta B_3\rangle$. $C_1= A_1–A_1\Theta B_1$, $C_2= A_2–A_2\Theta B_2$, $C_3= A_3–A_3\Theta B_3$, see following Figure 4 (a), (b) and (c). And, putting $D= C_1 +C_2 +C_3$, obtain the binary value edges of the original color image, see following Python program and Figure 5.,

Python Program 3.3

```
# Define a function to achieve the operation "x+y" replaced by "max{x, y}"
def Plus(C1,C2,C3):
    edge= C1+C2+C3
    for i in range(edge.shape[0]):
        for j in range(edge.shape[1]):
            if (edge[i, j]) >= 255:
                edge[i, j]=255
    return edge
# Define a function to achieve the dilation operation.
def dilate(binary, kernel):
    Kernel_Dilate = np.ones((kernel.shape[0], kernel.shape[1]), np.uint8)
    for i in range(kernel.shape[0]):
        for j in range(kernel.shape[1]):
            Kernel_Dilate[i,j] = kernel[i,j]
    return cv2.dilate(binary, Kernel_Dilate)
# Define a function to achieve the erosion operation
def erode(binary, kernel):
    Kernel_Erode = np.ones((kernel.shape[0], kernel.shape[1]), np.uint8)
    for i in range(kernel.shape[0]):
        for j in range(kernel.shape[1]):
```

```
            Kernel_Erode[i, j] = kernel[i, j]
        return cv2.erode(binary, Kernel_Erode)
B1 = np.ones((5,5), np.uint8)            # define an one matrix with five rows and five columns
C1 = erode(A1,np.array(B1))              # C1=A1ΘB1
C1 = A1- C1                              # C1= A1–A1ΘB1
C2 = erode(A2,np.array(B1))              # C2=A2ΘB2
C2 = A2- C2                              # C2= A2–A2ΘB2
C3 = erode(A3,np.array(B1))              # C3=A3ΘB3
C3 = A3- C3                              # C3= A3–A3ΘB3
D=Plus(C1,C2,C3)                         # D1= C1 +C2 +C3
```


(**a**) C_1


(**b**) C_2


(**c**) C_3

Figure 4. (**a**) the edge of image A_1; (**b**) the edge of image A_2; (**c**) the edge of image A_3.

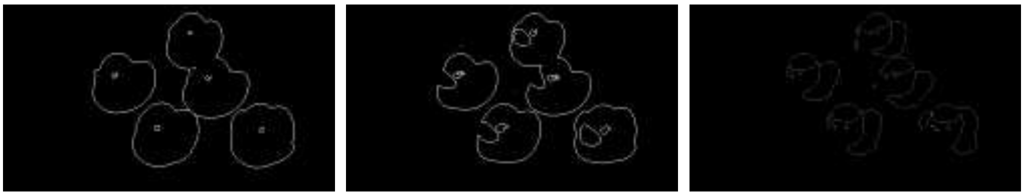
Figure 5. Merged edges of the original image (D).

Next, we can apply Definition 1.8 (Type 2) to give another edge extraction method similar to the above method. Putting $A=\langle A_1, A_2, A_3\rangle$, $B=\langle B_1, B_2, B_3\rangle$, where B_1 = (0, 0, 0, 1, 0, 0, 0; 0, 0, 1, 1, 1, 0, 0; 0, 1, 1, 1, 1, 1, 0; 1, 1, 1, 1, 1, 1, 1; 0, 1, 1, 1, 1, 1, 0; 0, 0, 1, 1, 1, 0, 0; 0, 0, 0, 1, 0, 0, 0), B_2 = (0, 0, 0, 1, 0, 0, 0; 0, 0, 1, 1, 1, 0, 0; 0, 1, 1, 0, 1, 1, 0; 1, 1, 0, 0, 0, 1, 1; 0, 1, 1, 0, 1, 1, 0; 0, 0, 1, 1, 1, 0, 0; 0, 0, 0, 1, 0, 0, 0),B_3 =(0, 0, 0, 0, 0, 0, 0; 0, 0, 0, 0, 0, 0, 0; 0, 0, 0, 1, 0, 0, 0; 0, 0, 1, 1, 1, 0, 0; 0, 0, 0, 1, 0, 0, 0; 0, 0, 0, 0, 0, 0, 0; 0, 0, 0, 0, 0, 0, 0), $A\oplus_2 B=\langle A_1\oplus B_1, A_2\Theta B_2, A_3\Theta B_3\rangle$. $C_1=(A_1\oplus B_1)-A_1$, $C_2= A_2-(A_2\Theta B_2)$, $C_3=A_3-(A_3\Theta B_3)$, see following Figure 6 (a), (b) and (c). And, putting $D_1= C_1+C_2+C_3$, obtain the binary value edges of the original color image, see following Python program and Figure 7.

Python Program 3.4

```
B1 = [[0,0,0,1,0,0,0],                   # define the matrix named B1
      [0,0,1,1,1,0,0],
      [0,1,1,1,1,1,0],
      [1,1,1,1,1,1,1],
      [0,1,1,1,1,1,0],
      [0,0,1,1,1,0,0],
      [0,0,0,1,0,0,0]]
B2 = [[0,0,0,1,0,0,0],                   # define the matrix named B2
      [0,0,1,1,1,0,0],
      [0,1,1,0,1,1,0],
      [1,1,0,0,0,1,1],
      [0,1,1,0,1,1,0],
      [0,0,1,1,1,0,0],
      [0,0,0,1,0,0,0]]
```

```
B3 = [[0,0,0,0,0,0,0],                    # define the matrix named B3
      [0,0,0,0,0,0,0],
      [0,0,0,1,0,0,0],
      [0,0,1,1,1,0,0],
      [0,0,0,1,0,0,0],
      [0,0,0,0,0,0,0],
      [0,0,0,0,0,0,0]]
C1 = dilate(A1, np.array(B1))             # C1=A1⊕B1
C1 = C1- A1                               # C1=(A1⊕B1)−A1
C2 = erode(A2, np.array(B2))              # C2= A2ΘB2
C2 = A2- C2                               # C2= A2−(A2ΘB2)
C3 = erode(A3, np.array(B3))              # C3= A3ΘB3
C3 = A3- C3                               # C3= A3−(A3ΘB3)
D=Plus(C1, C2, C3)                        # D1= C1+ C2+ C3
```



(**a**) C_1 (**b**) C_2 (**c**) C_3

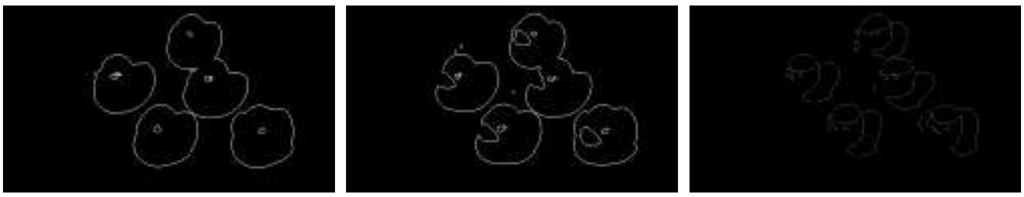
Figure 6. (**a**) the edge of image A_1; (**b**) the edge of image A_2; (**c**) the edge of image A_3.

Figure 7. Merged edges of the original image (D_1).

Finally, we can apply Definition 1.9 (Type 2) to give another edge extraction method similar to the above method. Putting $A=\langle A_1, A_2, A_3\rangle$, $B=\langle B_1, B_2, B_3\rangle$, where B_1, B_2, B_3 not change (see above), $A\Theta_2 B=\langle A_1\Theta B_1, A_2\oplus B_2, A_3\oplus B_3\rangle$. $C_1= A_1-(A_1\Theta B_1)$, $C_2= (A_2\oplus B_2)-A_2$, $C_3= (A_3\oplus B_3)-A_3$, see following Figure 8 (a), (b) and (c). And, putting $D_2= C_1+ C_2+ C_3$, see following Python program and Figure 9.

Python Program 3.5

```
C1 = erode (A1,np.array(B1))              # C1= A1ΘB1
C1 = A1- C1                               # C1= A1−(A1ΘB1)
C2 = dilate (A2,np.array(B2))             # C2= A2⊕B2
C2 = C2- A2                               # C2= (A2⊕B2)−A2
C3 = dilate (A3,np.array(B3))             # C3= A3⊕B3
C3 = C3- A3                               # C3= (A3⊕B3)−A3
D=Plus(C1, C2, C3)                        # D2= C1+ C2+ C3
```



(**a**) C_1 (**b**) C_2 (**c**) C_3

Figure 8. (**a**) the edge of image A_1; (**b**) the edge of image A_2; (**c**) the edge of image A_3.

Figure 9. Merged edges of the original image (D_2).

4. Conclusions

In this paper, some properties of the existing algebraic operations of neutrosophic crisp sets are discussed, and some new operations are given. The results shown that many different algebraic operation systems can be set up for neutrosophic crisp sets, they can be selected according to different applications. Meanwhile, this paper studied the application of neutrosophic crisp mathematical morphology in color image edge extraction, and the experimental results by Python shown that different morphological operators can be selected in this kind of application.

Because the color image binarization processing first in this paper, and then extract the edge by using morphological operator. So, the theory of neutrosophic crisp mathematical morphology need to do further research, so that we can deal directly with gray image or color image by using new morphological operators.

Funding: This research was funded by the National Natural Science Foundation of China, grant number 61976130.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. F.Smarandache, *A Unifying Field in Logics:Neutrosophic Logic. Neutrosophy, Neutrosophic Set, Neutrosophic Probability*; American Research Press: Rehoboth, NM, 1999.
2. F.Smarandache, Neutrosophic Set, A generalization of the intuituionistics fuzzy sets, *International Journal of Pure and Applied Mathematics* **2005**, 24, 287–297.
3. J.Ye, Simplified neutrosophic harmonic averaging projection based method for multiple attribute decision making problems, *International Journal of Machine Learning and Cybernetics* **2017**, 8(3), 981–987.
4. Y.Guo, A.S.Ashour, Neutrosophic multiple deep convolutional neural network for skin dermoscopic image classification. In *Neutrosophic Set in Medical Image Analysis*; Elsevier, 2019, pp. 269–285.
5. X.Zhang, C.Bo, F.Smarandache, J.Dai, New inclusion relation of neutrosophic sets with applications and related lattice structure, *International Journal of Machine Learning and Cybernetics* **2018**, 9, 1753–1763.
6. X.Zhang, C.Bo, F.Smarandache, C.Park, New operations of totally dependent-neutrosophic sets and totally dependent-neutrosophic soft sets, *Symmetry* **2018**, 10(6), 187; https://doi.org/10.3390/ sym10060187.
7. X.Zhang, F.Smarandache, Y.Ma, Symmetry in hyperstructure: neutrosophic extended triplet semi-hypergroups and regular hypergroups, *Symmetry* **2019**, 11(10), 1217; https://doi.org/10.3390/sym11101217.
8. J.Wang, X.Zhang, Multigranulation single valued neutrosophic covering-based rough sets and their applications to multi-criteria group decision making, *Iranian Journal of Fuzzy Systems* **2020**, 17(5), 109–126.
9. A.A.Salama and F.Smarandache, Neutrosophic crisp set theory, *Neutrosophic Sets and Systems* **2014**, 5, 27–35.
10. A.A.Salama, F.Smarandache, *Neutrosophic Crisp Set Theory*; Educational Publisher: Columbus, 2015.
11. I.Bloch, H.Heijmans, C.Ronse, Mathematical Morphology, In *Hand book of Spatial Logics*; M.Aiello, I.Pratt-Hartmann and J.van Benthem (eds.); Springer, 2007; pp. 857–944.
12. A.M.Raid, W.M.Khedr, M.A.El-dosuky, Mona Aoud, Image restoration based on morphological operations, *International Journal of Computer Science, Engineering and Information Technology* **2014**, 4(3), 9–21.

13. I.Bloch, H.Maitre, Fuzzy mathematical morphology, *Annals of Mathematics and Artificial Intelligence* **1994**, 10, 55–84.
14. E.M.El-Nakeeb, H.Elghawalby, A.A.Salama, S.A.El-Hafeez, Neutrosophic crisp mathematical morphology, *Neutrosophic Sets and Systems* **2017**, 16, 57–69.
15. E.M.El-Nakeeb, H.Elghawalby, A.A.Salama, S.A.El-Hafeez, Foundation for neutrosophic mathematical morphology, In *New Trends in Neutrosophic Theories and Applications;* Pons Editions Brussels, Belgium, 2016; pp. 363–379.
16. A.A.Salama, F.Smarandache, H.Elghawalby, Neutrosophic approach to grayscale images domain, *Neutrosophic Sets and Systems* **2018**, 21, 13–19.
17. A.A.Salama, H.Elghawalby, A.M.Nasr, Retract neutrosophic crisp system for gray scale image, *Asian Journal of Mathematics and Computer Research* **2018**, 24(3), 104–117.
18. Ahmed B. AL-Nafee, .Riad K. Al-Hamido , Florentin Smarandache, Separation Axioms in Neutrosophic Crisp Topological Spaces, Neutrosophic Sets and Systems, Vol. 25, 25-32, 2019.

Received: March 6, 2021. Accepted: Jun 1, 2021