An Application of Mathematics to Computer Programming: Connecting Translation Vectors, the Minkowski Difference, and Collision Detection

Author(s): Younhee Lee, Qi Lu and Woong Lim

Source: *The Mathematics Teacher*, Vol. 112, No. 2 (October 2018), pp. 150-154

Published by: National Council of Teachers of Mathematics

Stable URL: https://www.jstor.org/stable/10.5951/mathteacher.112.2.0150

Younhee Lee, Qi Lu, and Woong Lim

An Application of Mathematics to Computer Programming: Connecting Translation Vectors, the Minkowski Difference, and Collision Detection

Translation by a vector in the coordinate plane is first introduced in precalculus and connects to the basic theory of vector spaces in linear algebra. In this article, we explore the topic of collision detection in which the idea of a translation vector plays a significant role. Because collision detection has various applications in video games, virtual simulations, and robotics (Garcia-Alonso, Serrano, and Flaquer 1994; Rodrigue 2012), using it as a motivator in the study of translation vectors can be helpful. For example, students might be interested in the question, "How does the computer recognize when a player's character gets hit by a fireball?" Computer science provides a rich context for real-life applications of mathematics—programmers use mathematics for coding an algorithm in which the computer recognizes two objects nearing each other or colliding. The Minkowski difference, named after the nineteenth century German mathematician Hermann Minkowski, is used to solve collision detection problems (Ericson 2004). Applying the Minkowski difference to collision detection is based on translation vectors, and programmers use the algorithm as a method for detecting collision in video games.

Teachers can also benefit from meaningful examples in which technology is used to verify mathematical ideas. By presenting three cases of collision detection on two-dimensional objects, we hope to illustrate the use of technology to deepen mathematical understanding and demonstrate an application of computer programming. We hope that readers will gain an appreciation of the essential role of mathematics in creating real-time animations on the computer.

The Minkowski difference (MD) plays an important role in collision detection. The MD of two sets of vectors, A and B, is the set of all vectors obtained by subtracting each vector in B from each vector in A, as shown in the following:

$$A - B = \left\{ \left(\vec{a} - \vec{b} \right) \middle| \, \vec{a} \in A, \, \vec{b} \in B \right\}.$$

Given two convex figures, each of which can be represented by connecting a set of vertices, we can associate each figure with a set of its vertices, say, set A and set B. Hereafter we denote a closed convex figure formed by connecting some or all points in set A such that all points in set A are contained in the figure as *the shape of A*. (When there is only one element in set A, the shape of A is a point; when there are only two different elements in set A, the shape of A is a segment.)

This article aims to illustrate how the binary

Delving Deeper offers a forum for new insights into mathematics, engaging secondary school teachers to extend their own content knowledge. Submit manuscripts via **http://mt.mtsubmit.net**.

Department editors
Daniel Ness, nessd@stjohns.edu, St. John's University, Jamaica, New York; **Nick Wasserman**, wasserman@tc.columbia.edu, Teachers College, Columbia University, New York; and **Benjamin Dickman**, benjamindickman@gmail.com, The Hewitt School, New York

```
Algorithm 1 collision-detection (A, B)
Set the Minkowski Difference MD to be empty
for vector a⃗ in object A do
    for vector b⃗ in object B do
        Put the Minkowski Difference of (a⃗ - b⃗) into MD
    end for
end for
Compute the convex hull (Hull) for the polygon of MD
if the origin (0, 0) in Hull then
    Return "A collides with B"
else
    Return "A does not collide with B"
end if
```

Fig. 1 Pseudo code is a structured description delineating the entire logic of an algorithm.

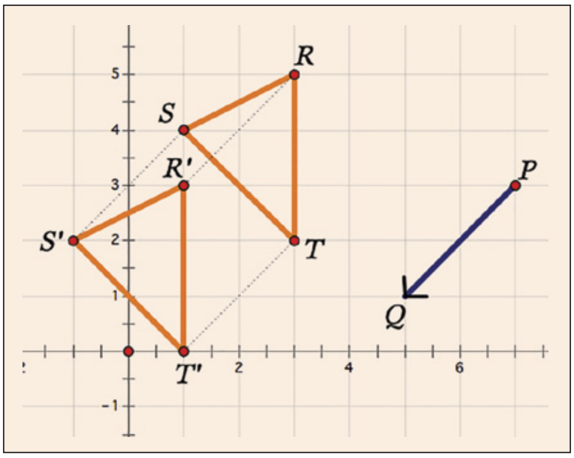


Fig. 2 Vector *PQ* translates triangle *SRT* two units to the left and two units down.

operation of MD on the vertices of the shapes of A and B will result in a new shape and discuss why the two figures A and B collide if and only if the origin is inside that new shape. This concept may not come to light easily. Some students also want to see the relevance of the concept to practical applications. First, we show examples of the way the MD is applied to programming a collision detection algorithm (see **fig. 1**); and second, we use vector translations to illustrate the relationship between the MD and collision detection.

Figure 1 shows pseudocode for collision detection using the MD. The objects A and B serve as input; $\vec{a}-\vec{b}$ is computed for each $\vec{a} \in A$, $\vec{b} \in B$, and the result is stored in MD. Then, we compute the *convex hull* for the shape of the MD. The convex hull of MD is the set of all points that constitute the shape of MD. An algorithm called *GJK* can be used to determine whether the origin is in the convex hull of MD. The algorithm is too complicated to include in this short article, but interested readers should go to Gilbert, Johnson, and Keerthi (1988) or Ericson (2004, pp. 400–403). If the origin is in the convex hull of MD, we can claim that object A collides with object B. Otherwise, object A does

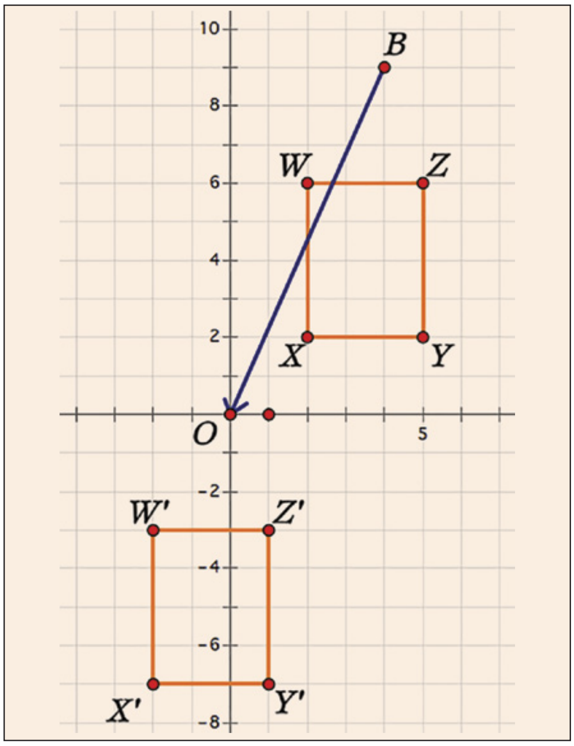


Fig. 3 A rectangle slides by the translation vector *BO*.

not collide with object B. When we execute the program, three cases are produced. In the first case, a point collides with an object; in the second, a segment and a triangle collide; and in the third, two triangles collide.

We now provide a geometric interpretation of collision detection for all cases using a translation vector dynamic mathematics platform. With the increasing complexity of the colliding figures, the programming of collision detection is a compelling example of the contribution of the Minkowski difference to digital technology.

CASE 1: POLYGON AND POINT

Let's consider an example in two-dimensional Euclidean space. $A = \{(2, 2), (5, 2), (5, 6), (2, 6)\}$, the shape of which forms a rectangle; $B = \{(4, 9)\}$, the shape of which forms a point. The MD of A and B is a new set of points $C = \{(-2, -7), (1, -7), (1, -3), (-2, -3)\}$, and the shape of C does not contain the origin $(0, 0)$. This indicates that the shape of B (that is, point B) does not collide with the shape of A. How can one tell from the MD whether the two objects collide?

We introduce a geometric interpretation of this relationship using a translation vector. *Translation by a vector* $\vec{v}$, often denoted as $T_{\vec{v}}$, is a type of isometry that slides a geometric object along the vector $\vec{v}$. For example (see **fig. 2**), triangle SRT is mapped to triangle $S'R'T'$ under a translation by vector PQ. When the coordinates of S, R, T, P, and Q are given, we can identify the coordinates of

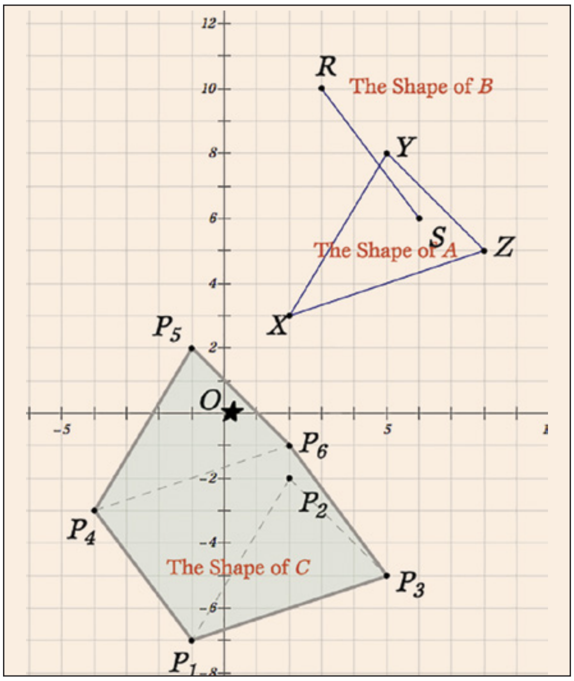


Fig. 4 The shape of the MD contains the origin, where (MD of A and B) = {P_1, P_2, P_3, P_4, P_5, P_6}, given A = {X, Y, Z} and B = {R, S}.

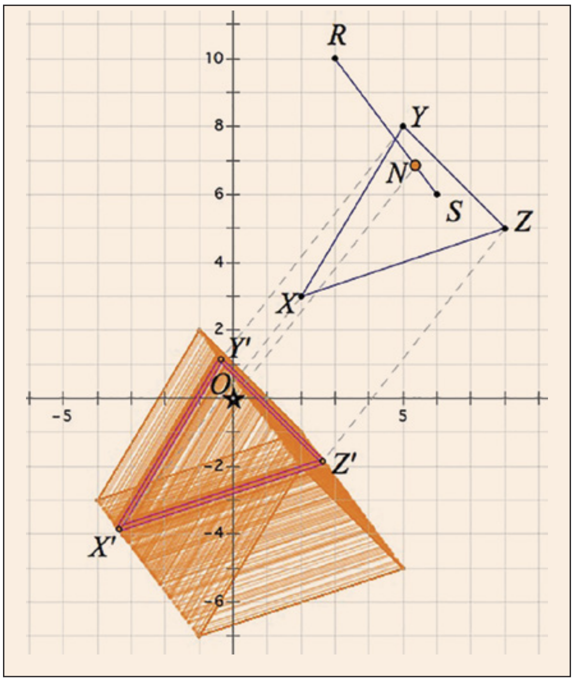


Fig. 5 The trace of the MD of A and N is shown as N moves along segment RS.

S', R', and T' by adding vector PQ (i.e., (–2, –2)), to each of the vectors OS, OR, and OT (O refers to the origin). Note that adding vector PQ is equivalent to subtracting vector QP.

The operation performed to obtain the MD of A and B is equivalent to translating the shape of A by vector BO. The four points of the shape of A (i.e., X, Y, Z, and W; see **fig. 3**) are mapped to the four points of the image of shape of A (i.e., X', Y', Z', and W') under the translation by vector BO. This means that the coordinates of X', Y', Z', and W' can be obtained by adding vector BO to the vectors OX, OY, OZ, and OW; in other words, subtracting vector OB from the coordinates of OX, OY, OZ, and OW. Hence, the MD of A and B is the translation of the shape of A by vector BO. Now, let's consider the importance of the origin (0, 0) in collision detection. The origin is our point of interest because it is the image of point B under the translation by vector BO. Whether the origin is contained in the shape of the image of A tells us if the shape of A and point B will collide since *collisionality* is preserved under translation. In this way, we only need to examine the relationship between the MD and the origin to detect collision between a polygon and a point.

CASE 2: POLYGON AND SEGMENT

Still working in two-dimensional Euclidean space, consider $A = \{(2, 3), (5, 8), (8, 5)\}$, the shape of which forms a triangle, and $B = \{(3, 10), (6, 6)\}$, the shape of which forms a segment. The MD of A and B is a new set of 6 points:

$$C = \{(-1, -7), (2, -2), (5, -5), (-4, -3), (-1, 2), (2, -1)\}$$

The shape of C is a large convex polygon that includes all other elements of the set. The points in set A (see **fig. 4**) are marked as X, Y, and Z; the points in set B are marked as R and S; and the points in set C are marked as P_1, P_2, P_3, . . . , P_6. Applying our insight from the first case, we know that the shape of C contains the origin, so the two figures collide.

How can we apply the idea of vector translation to make sense of this case when there is no single vector that defines a vector translation? The difference between the shape of B in case 1 (i.e., point B) and the shape of B in case 2 (i.e., segment RS) is that the former consists of a single point and the latter consists of a set of infinitely many points on the segment. Detecting a collision between a segment and a polygon is equivalent to detecting collisions between *any points on the segment* and the polygon. Hence, we can continue to apply the method used in case 1 to all points lying on segment RS and check if there exists a point on segment RS, say N, such that the shape of the MD of A and N contains the origin.

We consider the shape of the MD of A and any point N on segment RS (see **fig. 5**), which results in triangle $X'Y'Z'$ containing the origin. Now imagine how the MD would behave as point N moves along segment RS. Using the *trace* feature in the Geometer's Sketchpad® (GSP is available at http://www.dynamicgeometry.com), we captured the dynamic traces of the MDs (see **fig. 5**).

Notice how the union of the orange triangles (i.e., the shapes of the MDs of A and N *for all points* N on segment RS) forms the polygon $P_1P_3P_6P_5P_4$ (compare **fig. 5** and **fig. 4**). By the definition of MD, we obtained the points P_1, P_2, P_3, P_4, P_5, and P_6 by subtracting the endpoints of the shape of B from the three vertices of the shape of A one by one. **Figures 4** and **5** illustrate that the traces of triangle $X'Y'Z'$ remain in the polygon $P_1P_3P_6P_5P_4$. Hence, we conclude that the relationship between the origin and the MD of A and B suffices to identify a collision between the shape of A and the shape of B. Next, we generalize our collision detection using the Minkowski difference and translation vectors to two polygons.

CASE 3: POLYGON AND POLYGON

Consider the two triangles $A = \{(2, 3), (5, 8), (8, 5)\}$ and $B = \{(3, 10), (6, 6), (10, 9)\}$. The MD of A and B is a new set of 9 points

$$C = \{(-1, -7), (2, -2), (5, -5), (-4, -3), (-1, 2), (2, -1), (-8, -6), (-5, -1), (-2, -4)\},$$

and this set forms a large convex polygon that includes all other elements in the same set. The points in set A are marked as X, Y, and Z; points in set B are marked as S, R, and T; and the points in set C are marked as $P_1, P_2, P_3, \ldots, P_9$ (see **fig. 6**). The shape of C contains the origin, and this indicates that the two figures collide.

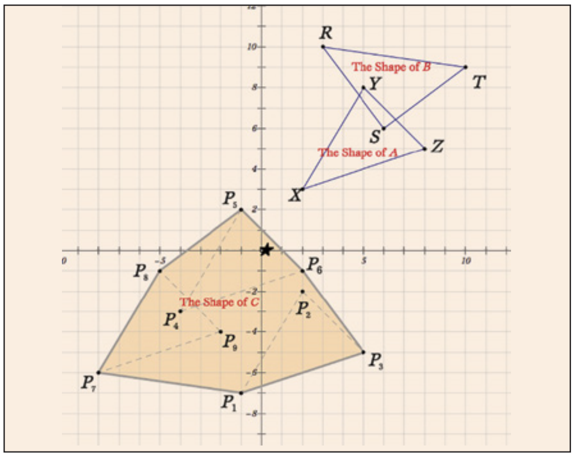


Fig. 6 The shape of the MD contains the origin, where (MD of A and B) = $\{P_1, P_2, P_3, P_4, P_5, P_6, P_7, P_8, P_9\}$, given $A = \{X, Y, Z\}$ and $B = \{R, S, T\}$.

Case 3 is an extension of case 2 in that set A is identical in both cases, and set B in case 2 is a subset of set B in case 3. One of the edges forming triangle RST is already taken care of in case 2. In case 3, for each of the points M_1, M_2, and M_3 (see **fig. 7**), three translation vectors include vector M_1O, vector M_2O, and vector M_3O.

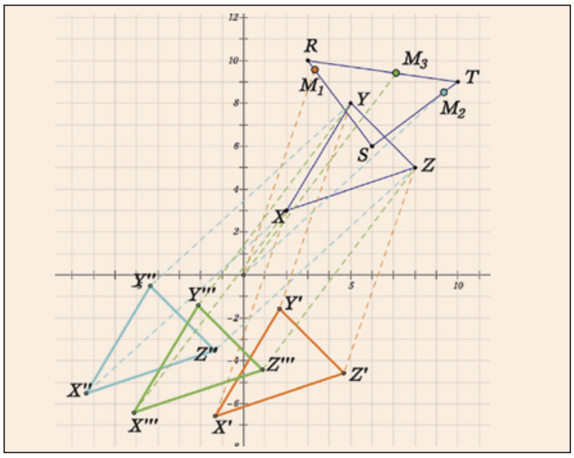


Fig. 7 (MD of A and M_1) = $\{X', Y', Z'\}$, given $A = \{X, Y, Z\}$ and a point M_1; (MD of A and $M2$) = $\{X'', Y'', Z''\}$, given $A = \{X, Y, Z\}$ and a point M_2; (MD of A and $M3$) = $\{X''', Y''', Z'''\}$, given $A = \{X, Y, Z\}$ and a point M_3.

Each translation vector produces three different images of triangle XYZ. Triangle $X'Y'Z'$ is an image of triangle XYZ under the translation by vector M_1O; triangle $X''Y''Z''$ is an image of triangle XYZ under the translation by vector M_2O; and triangle $X'''Y'''Z'''$ is an image of triangle XYZ under the translation by vector M_3O. From case 2, we know how a trace of triangle $X'Y'Z'$ would look as M_1 moves along the segment RS. The same goes for two other images as M_2 or M_3 moves along each segment. The dynamic tracing represents possible images of XYZ (see **fig. 8**).

Now we need to consider the union of these three traces because the collision between the shape of A and the shape of B means the collision between triangle XYZ and one (or more) of the edges of triangle RST. The three traces share some of their boundaries, and their union forms a hexagon (see **fig. 9**). Sharing boundaries occurs when M_1 and M_2 meet at S, when M_2 and M_3 meet at T, and when M_3 and M_1 meet at R. Hence, the vertices of the hexagon formed by the union of the three traces are determined by the vertices of the shape of A and the shape of B. The MD of A and B produces the vertices of the hexagon as illustrated by the equivalent hexagons (see **fig. 9** and **fig. 6**). We do not consider the MD for the points inside triangle RST, since these points define translation vectors under which the image of triangle XYZ always lies within the boundaries of the hexagon.

One pattern from cases 1, 2, and 3 is that using the points on the boundary for translation vectors is sufficient for collision detection. For example, to detect collision between two-dimensional triangles in case 3, it was sufficient to consider one-dimensional edges surrounding the face of each triangle. This was reduced to considering

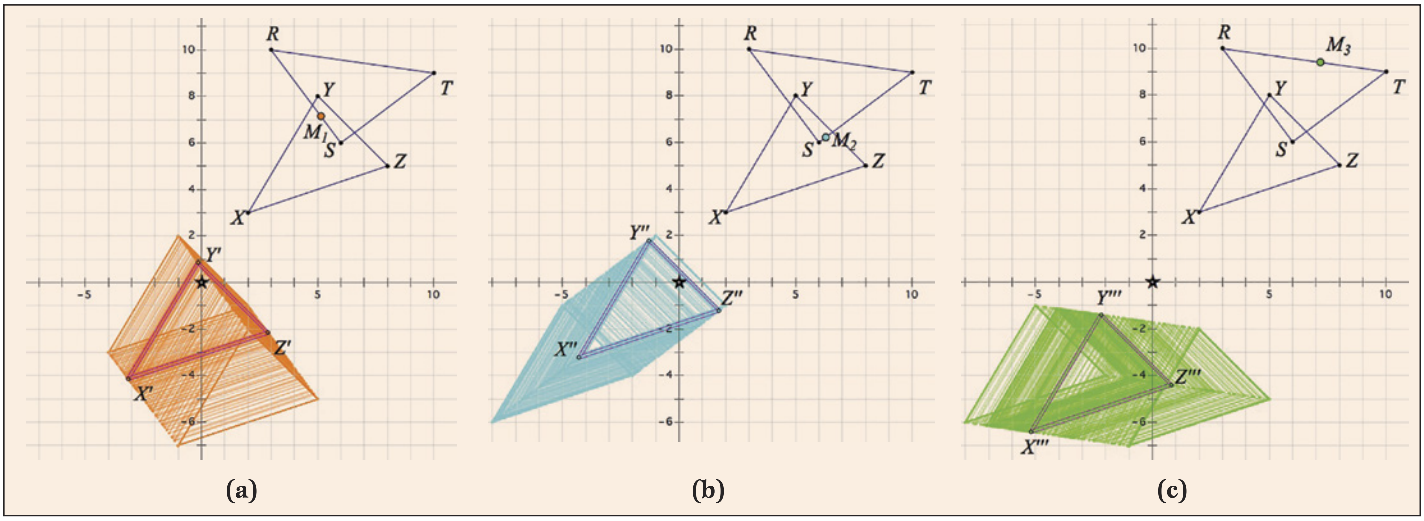


Fig. 8 Trace of (MD of *A* and M_1) as M_1 moves along the segment *RS* is shown in (a); trace of (MD of *A* and M_2) as M_2 moves along the segment *ST* is shown in (b); and trace of (MD of *A* and M_3) as M_3 moves along the segment *TR* is shown in (c).

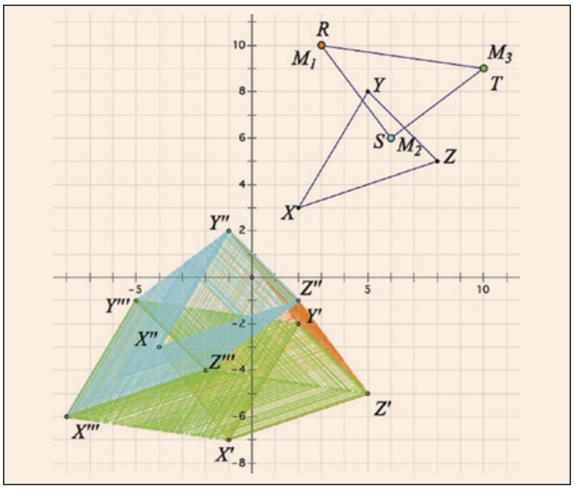


Fig. 9 The union of all three traces forms a hexagon.

zero-dimensional vertices defining each of the edges. Eventually, three-dimensional-object collision can be reduced to considering two-dimensional faces surrounding the space, then to considering one-dimensional edges surrounding the faces, then to considering zero-dimensional vertices defining the edges. In this regard, using the Minkowski difference for detecting collision between two objects signifies the important role of vertices in collision detection and enables us to program an algorithm to execute collision detection in computer graphics.

The context of coding and collision detection can be intriguing for some students who seek to explore interesting connections between mathematics and other disciplines. Using the graphical displays found with GSP, as illustrated in this article, can also contribute to increasing student insight and intuition related to vector translation.

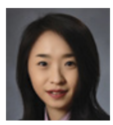
REFERENCES

Ericson, Christer. 2004. *Real-Time Collision Detection*. Boca Raton, FL: CRC Press.

Garcia-Alonso, Alejandro, Nicolás Serrano, and Juan Flaquer. 1994. “Solving the Collision Detection Problem.” *IEEE Computer Graphics and Applications* 14, no. 3: 36–43.

Gilbert, Elmer G., Daniel W. Johnson, and S. Sathiya Keerthi. 1988. “A Fast Procedure for Computing the Distance between Complex Objects in Three-Dimensional Space.” *IEEE Journal on Robotics and Automation* 4, no. 2: 193–203.

Rodrigue, Benjamin. 2012. “Collision Detection in Video Games.” In *Algorithmic and Architectural Gaming Design: Implementation and Development*, pp. 221–252. Hershey, PA: IGI Global.



YOUNHEE LEE, yul182@psu.edu, is a PhD student in mathematics education at Pennsylvania State University in State College. Her research focuses on connections between college mathematics and school mathematics and the use of technology in teaching mathematics. **QI LU,** lukey11@unm.edu, is a PhD student in computer science at the University of New Mexico. He is interested in autonomous robotics, self-organizing systems, and swarm robotics and enjoys connecting mathematics to computer programming. **WOONG LIM,** woonglim@unm.edu, is an assistant professor of mathematics education at the University of New Mexico. His research interests include using computing and digital technology to support teaching and learning and mathematical discourse.