

國立政治大學資訊科學系  
Department of Computer Science  
National Chengchi University

碩士論文  
Master's Thesis

The logo of National Chengchi University is a circular emblem with a stylized flower in the center. The Chinese characters '國立政治大學' are arranged around the top inner edge, and 'National Chengchi University' is written around the bottom inner edge. The logo is rendered in a light gray, semi-transparent watermark style.

樂高立體書的自動生成系統  
Automatic Generation of Lego Pop-ups

研究生：湯昊軒

指導教授：紀明德

中華民國一百一十年一月

Jan 2021

# 樂高立體書的自動生成系統

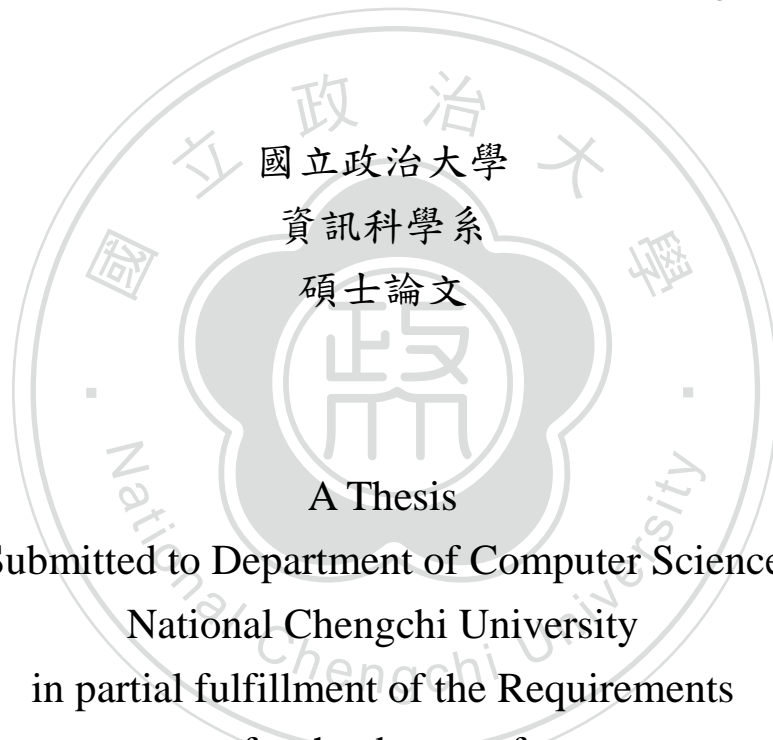
## Automatic Generation of Lego Pop-ups

研究生：湯昊軒

Student：Hao-Hsuan Tang

指導教授：紀明德

Advisor：Ming-Te Chi



Submitted to Department of Computer Science  
National Chengchi University  
in partial fulfillment of the Requirements  
for the degree of  
Master  
in  
Computer Science

中華民國一百一十年一月

Jan 2021

## 摘要

紙摺立體書利用紙張堆疊與交互作用來產生複雜立體結構，其精巧的摺疊設計能讓各年齡層的人皆感到有趣味。而樂高(LEGO)從1940年代發展至今，成為了在建構立體模型的領域中家喻戶曉的一種工具。在2016年，Lego公司的Lego ideas平台出現了將紙摺立體書與樂高積木結合的樂高立體書結構，激發了我們想要設計可以自動化生成樂高立體書之電腦輔助系統的動機。

雖然在先前有針對紙摺立體書與樂高建構的各種研究，但並沒有將兩者結合的相關研究，故本研究在樂高立體書的觀察與分析下，提出了三個樂高立體書的機制，透過公式化來分析樂高磚間如何彼此排列形成合理的樂高立體書結構。接著我們設計一個系統，將輸入的二維影像透過使用者選定的機制與相關參數來生成初始結構，接著透過我們提出的優化目標函式與優化流程，在無窮的三維結構設計複雜度與可行的時間中，來生成同時具有穩定性與複雜度，且合理的樂高立體書結構。在研究結果中，我們輸入多種不同類型的影像與參數產生結果，來驗證我們系統可用於各種輸入影像的可行性。

## Abstract

Paper pop-ups demonstrate complex three-dimensional structures by the arrangement of paper. The delicate design of paper pop-ups fascinates people of all ages. Since the 1940s, Lego becomes a well-known tool in the field of constructing three-dimensional models. In 2016, a V-fold type Lego pop-up book was published on the Lego ideas platform. This amazing work has led to our computer-assisted tools for designing Lego pop-ups.

Although there have been various studies on paper pop-up books and Lego construction, there is no related research that combines the two. Therefore, based on the observation and analysis of Lego pop-ups, we propose three Lego pop-ups mechanisms. We analyze how to arrange the Lego bricks to form a reasonable Lego pop-up structure through general formulation. Based on the proposed mechanisms, we design a system to generate an initial layout design from the input of two-dimensional images and user-specified mechanisms and parameters. Then, through the proposed optimization model, we compute a stable structure that is easy to assemble in a feasible time even the complexity of the three-dimensional is infinite. We verify the feasibility of the proposed method by various kinds of input images of our system.

# 目錄

摘要.....	1
Abstract.....	2
目錄.....	3
圖目錄.....	5
表目錄.....	8
一、 簡介.....	9
1.1 研究動機.....	9
1.2 問題描述.....	10
1.3 論文貢獻.....	10
1.4 論文章節架構.....	11
二、 相關研究.....	12
2.1 樂高積木的建構.....	12
2.2 立體書折疊結構.....	13
2.3 連動結構.....	14
三、 樂高立體書的基本介紹.....	16
3.1 立體書之術語.....	16
3.2 紙立體書的機制.....	16
3.2.1 Step-fold 階梯型機制.....	17
3.2.2 Tent-fold 帳篷型機制.....	17
3.2.3 V-fold V 型機制.....	17
3.3 樂高磚塊厚度與比例限制.....	17
3.4 樂高立體書之術語.....	18
3.5 系統使用的樂高磚集.....	19
四、 樂高立體書之機制分析.....	21
4.1 Tent-fold 帳篷式機制.....	21
4.1.1 定義與可折疊性.....	21
4.1.2 書頁的寬度 $w$ .....	22
4.1.3 書頁的高度 $h$ .....	23
4.1.4 非左右對稱之帳篷式機制.....	23
4.2 Step-fold 階梯式機制.....	27
定義與可折疊性.....	27
4.3 V-fold V 型機制.....	28
4.3.1 定義與可折疊性.....	28
4.3.2 書脊的寬度 $depth$ .....	30
4.3.3 書頁的寬度 $w$ .....	31
4.3.4 書頁的高度 $h$ .....	32

4.3.5 避免合頁磚與書脊產生碰撞.....	33
五、 研究方法.....	35
5.1 系統流程.....	35
5.2 系統建置的前處理 .....	36
5.2.1 樂高模型檔匯入.....	36
5.2.2 樂高磚的資料結構.....	37
5.2.3 樂高磚的物理模擬引擎.....	37
5.3 輸入影像之處理 .....	39
5.3.1 影像感興趣區域擷取.....	39
5.3.2 影像分析.....	40
5.4 初始的設計生成 .....	40
5.5 樂高磚結構調整與優化目標函式 .....	40
5.5.1 優化的目標函式 (Objective Function).....	41
5.5.2 結構複雜度(Model Complexity).....	41
5.5.3 結構穩定性(Model Stability).....	42
5.5.4 結構的支撐磚曝光程度(Exposed Ratio of BRACEs) .....	46
5.6 系統的樂高磚結構優化方法與實作細節 .....	47
5.7 使用者互動介面 .....	49
六、 研究結果.....	51
6.1 階梯式結構 .....	51
單面呈現之結構.....	54
雙面呈現之結構.....	55
6.2 V 型結構.....	56
6.3 書框的生成 .....	59
6.4 研究分析.....	61
七、 結論.....	63
7.1 結論.....	63
7.2 未來展望.....	63
參考文獻.....	65

## 圖目錄

圖 1	由創作者設計的樂高立體書。左為展開的立體結構，右為折疊收合的結果。摘自 Lego ideas 平台。.....	9
圖 2	樂高研究。(a)組合方式與對應的結構表示，摘自[Testuz et al. 2013]。(b)大型樂高結構，摘自[Luo et al. 2015]。(c) 2D pixelart lego，摘自[Kuo et al. 2015]。(d) Silhouette fitting 的成果，摘自[Yun et al 2017]。.....	13
圖 3	立體卡片的相關研究。(a) 以巢狀 step fold 構成，摘自[Li et al. 2010]。(b) 以複合折疊結構構成，摘自[Li et al. 2010]。(c) 以變形折疊結構構成，摘自[Xiao et al. 2018]。.....	14
圖 4	連動機構設計的相關研究。(a)(b)(c)分別為指定運動形式、經演算法設計出的機構、以及製造實體結果。摘自[Zhu et al. 2012]。(d)為另一個以齒輪組設計的機構做法，摘自[Coros et al. 2013]。.....	15
圖 5	紙立體書的術語之圖示，摘自[Ruiz et al. 2014]。.....	16
圖 6	各種類型的紙立體書機制：(a) 階梯型 (b) 帳篷型 (c)(d)不同型態的 V 型，摘自[Ruiz et al. 2014]。.....	16
圖 7	樂高長寬比例示意圖。左：基本 1x1 磚的比例。中：樂高立體書比例分析示意圖。右：不同轉向對齊示意圖。.....	18
圖 8	書頁示意圖。我們將書頁分成左書頁、書脊與右書頁。.....	19
圖 9	系統使用的樂高磚示意圖。(a)基本磚；(b)薄磚與平滑磚；(c)變形基本磚；(d)樞紐磚；(e)科技磚。科技磚第一排從左到右分別為帶插銷孔的1×2基本磚、L型連接磚(Connector#6)與孔樑磚，第二排則為十字軸磚與插銷磚。科技磚因種類繁多，在圖中僅列最常使用的類型。.....	20
圖 10	樂高立體書的機制示意圖。(a)帳篷式，(b)階梯式 (c)V 型，圖中的藍線代表書頁在闔上時移動的方向，紅線代表書中立體結構在闔上時移動的方向。.....	21
圖 11	展開的對稱式帳篷式立體書結構圖(從後往前的視角)。兩側結構與書脊的距離皆為兩個樂高磚單位寬。.....	22
圖 12	闔上的帳篷型立體書結構圖。左圖為前視角，右圖為去除左書頁透視圖(正交視角)。d為兩側的結構與書脊的距離(此圖為四個樂高磚寬)，H為結構的 L 型連接磚到頂端的樞紐磚之樞紐的距離。.....	23
圖 13	展開的非左右對稱帳篷式結構之示意圖(俯瞰視角)。其中左側的結構與書脊的距離為八個樂高磚寬，右側的結構與書脊的距離為四個樂高磚寬，因右側的結構離書脊較近，故其結構高度(L 型連接磚與樞紐磚的距離)需要比左側高 10 個樂高磚單位的高度來達成能正確闔上立體書的條件。.....	24
圖 14	將圖 13 闔上後呈現的非左右對稱帳篷式結構之示意圖(正面視角)。....	25
圖 15	帳篷式機制的開闔多角度示意圖。從上到下分別為0o(完全闔上)、45o、90o、135o與180o(完全展開)。.....	26

- 圖 16 展開的 step fold 基本結構示意圖。左圖為透視圖，右圖為正交視角。 .28
- 圖 17 展開的 V 型樂高立體書結構示意圖 (從上往下的視角)。其中綠色區域的樂高磚為書頁部分，棕色區域為連接書頁與立體結構的科技磚，而灰色區域為立體結構呈現的部分。橘色的圈為使結構正常旋轉的樞紐磚。.....29
- 圖 18 V 型結構左側的科技磚結構。透過棕色的十字軸磚與灰色的帶插銷孔的薄磚連結，來達成類似於滑軌的結構來讓立體書正常地開闔。其中箭頭為結構區域性的座標軸。.....29
- 圖 19 闔上的樂高立體書結構示意圖。左圖為前視角；中間圖為上視角；右圖為底部視角，我們透過引入窗框的樂高磚來避免樞紐磚之轉軸與書脊產生物理碰撞。.....30
- 圖 20 去除左書頁之闔上的立體書結構圖 (從左往右的視角)。.....32
- 圖 21 V 型機制的開闔多角度示意圖。從上到下分別為  $0^\circ$ (完全闔上)、 $45^\circ$ 、 $90^\circ$ 、 $135^\circ$ 與 $180^\circ$ (完全展開)。.....34
- 圖 22 系統流程圖。首先我們將輸入的影像透過感興趣區域的擷取以及影像的分析來產生出輸入影像的像素模型，接著將像素模型根據使用者要求的機制產生出初始的結構設計，最後在我們設計的優化演算法之引導下，隨機地調整樂高的設計來使結構擁有良好的穩定性與複雜度。.....35
- 圖 23 開源樂高軟體 Ldraw 的  $1 \times 1$  基本磚模型檔。.....36
- 圖 24 模型複雜度之示意圖。圖左為一個  $1 \times 4$  基本磚組成的結構；圖中為一個  $1 \times 2$  與兩個  $1 \times 1$  組成的結構；圖右為十二個  $1 \times 1$  薄磚組成的結構。.....42
- 圖 25 區域性垂直連結度之示意圖。以每個結構下方的  $1 \times 4$  基本磚計算其螺柱的連結度部分： $a = 0$  ； $b = 17$  ； $c = 27$  ； $d = 37$  ； $e = 27$  ； $f = 37$  ； $g = 47$  ； $h = 57$  ； $i = 67$  ； $j = 1$ 。十個結構的  $1 \times 4$  基本磚計算其螺柱孔的連結度皆為 1。.....44
- 圖 26 區域性水平連結度之示意圖。以圖中灰色的  $2 \times 8$  薄磚為例，該樂高磚的前後左右側有 20 個相鄰的單位空間(如紅框)。.....45
- 圖 27 區域性穩定度相同但結構穩定度具有差異的示意圖。其中紅色磚代表  $1 \times 1$  基本磚；綠色代表  $1 \times 2$  基本磚；藍色代表  $1 \times 3$  基本磚。.....46
- 圖 28 將圖 27 之樂高磚結構轉化的無向圖。其中紅色節點代表  $1 \times 1$  基本磚；綠色節點代表  $1 \times 2$  基本磚；藍色節點代表  $1 \times 3$  基本磚。.....46
- 圖 29 相同輸入的不同設計產出結果。左邊二圖為同一結構，分別為正面視角與側面上方的視角，右邊同理。其中  $EERoBsMleft \cong 0.47$ ， $EERoBsMright \cong 0.01$ 。.....47
- 圖 30 左圖為表格 1 第二列參數生成的結構圖，右圖為第五列的結構圖。.....52
- 圖 31 左圖為表格 2 第二列參數生成的結構圖，右圖為第五列的結構圖。.....53
- 圖 32 呈現立體文字的樂高書結構。左圖為輸入影像，使用者設置結構的寬度為 15，中間圖為系統產出之設計，右圖為將立體書闔上後的結構圖(切除背



- 面的書頁)。.....54
- 圖 33 利用三個相同機制產生的樂高書結構。輸入影像與圖 22 相同，使用者分別將三個結構的結構寬度分別設置為 10,9,10(從左到右)，左圖為結構的正面視角，中間圖為結構的斜上方視角，右圖為將立體書闔上後的結構圖(切除背面的書頁)。.....55
- 圖 34 使用雙面呈現的階梯式機制生成之骷髏結構。左圖為輸入影像，中間圖為 $dL = dR$ 時系統生成的對稱設計，右圖則為 $dL > dR$ 時生成的左右非對稱之設計。我們將輸入影像的白色部分轉為黑色以方便觀察。.....55
- 圖 35 使用四個雙面呈現的階梯式機制生成的立體文字結構。左圖為輸入影像，右圖為系統生成的設計。.....56
- 圖 36 上圖為表格 3 第四列參數生成的結構圖，右圖為第五列的結構圖。可看出較大的結構角度所使用的十字軸磚長度較短。.....57
- 圖 37 呈現山、雲與天空的風景之 V 型結構。左邊三張圖為輸入的影像，從下到上分別為近景、中景與遠景。右圖為輸出的結果，我們將輸入影像中白色的雲轉為黑色以方便觀察。.....58
- 圖 38 呈現簡單房屋的 V 型結構。左圖為輸入影像(參考圖 1)，中間圖為生成之設計，右圖為將立體書闔上後的結構圖(切除左書頁)。.....58
- 圖 39 書框生成的示意圖。(a)系統產出之樂高立體書結構；(b)系統於書頁外加入一層由變形基本磚、基本磚與薄磚組成的書框底座；(c)生成合適高度的書框磚；(d)將書框抬起後，從下方視角可看到書頁上每個變形基本磚的螺柱皆有對準書框樂高磚的螺柱孔；(e)將包含書框的樂高立體書結構闔上後之示意圖。.....60
- 圖 40 使用垂直方向的合併可能產生之結構比較圖。左圖為初始設計，為 $5 \times 5$ 的薄磚空間，中間圖為目前系統優化中可能產生的結構圖，右圖則為使用垂直方向合併可能產生的結構圖。可看出即便中間圖與右圖的結構皆有足夠的穩定性，但右圖的結構相對更難組裝，使用者看起來也更加零碎。..64

## 表目錄

表格 1 單面呈現的階梯式結構的參數值列表(單位：LDU)。結構圖以左為使用者定義之參數，以右為系統自動計算適合的對應參數。.....	52
表格 2 雙面呈現的階梯式結構的參數值列表(單位：LDU)。結構圖以左為使用者定義之參數，以右為系統自動計算適合的對應參數。.....	53
表格 3 V 型結構的參數值列表(單位：LDU)。左三欄為使用者定義之參數；後三欄為系統自動計算適合的對應參數(單位:D、W、d 為 LDU； $\theta$ 、 $\theta'$ 為度)。	57
表格 4 優化權重的設計與優化前後的目標函式值之變化表。.....	62



# 一、簡介

## 1.1 研究動機

紙摺立體書透過精巧的折疊設計，原本壓縮的平面形態可在展開後可呈現令人驚奇的立體結構，帶給讀者鮮明且互動的體驗，成為童書、繪本以及賀卡上常用的設計方式，經過無數的創作者的設計，已發展出許多驚豔的作品。但利用樂高積木做為立體書的設計元件是近幾年才開始，由 Grant Davis 和 Jason Allemann 發表在 Lego 公司所提供的 Lego ideas 平台的樂高立體書是激發本次研究企劃的主要動機之一。他們於 2016 以小紅帽等童話故事為主題，以樂高積木與關節連動零件，設計出立體書(圖 1)，在 Lego ideas 平台上獲得超過 10,000 人支持後，經樂高公司評選後在 2018 年進入商品化，可見樂高積木在創作的可能性與發展性。



圖 1 由創作者設計的樂高立體書。左為展開的立體結構，右為折疊收合的結果。摘自 Lego ideas 平台。

立體物體的呈現方法為電腦圖學的主要研究之一。傳統以光學現象的重現與幾何物體的數位資料結構為主。近年來由於 3D 列印技術的發展，探索實體空間中物體的製造方式，也成為重要的研究議題。雖然 3D 列印技術提供了以積層製造方式將實體物體逐層產生，但速度過慢一直是其主要限制。相對的，如同影像是由像素所構成，實體物體也需由基礎的元件構成，樂高積木具備大量生產且具備高精確度的特性，目前記錄可組成 35 公尺高的物體，甚至可打造出真實比例

的可駕駛汽車，是相當可靠的基本元件。此外，樂高除了常見的基本磚，還有許多類型的樂高磚，可用於打造具備開闔、轉動、伸縮等關節活動，透過多元的零件打造可動樂高作品，使得樂高設計有更多元的變化。

圖 1 將紙摺立體書與利用樂高來建構立體模型兩者做結合，而此類型的作品由於同時要考量立體模型的呈現、連動性與可動性的樂高結構，故大多數的創作皆來自於專業的樂高結構設計師經過長時間不斷的反覆試驗來組裝而成的。而透過觀察與比較類似的樂高立體書作品，我們發現樂高立體書有透過計算機自動化計算與輔助設計的空間，可以有效的利用自動化的計算生成立體書結構，減少創作家設計樂高立體書的時間。

## 1.2 問題描述

前面提到樂高可以透過多種類型的樂高磚結合，來建構出多種可動性的結構。這些元素，使得探討運用樂高打造立體書有極高的可能性與零件運用上的變化，怎樣的設計與組合，才能以樂高組合出來，其中的複雜度與限制如何評估，都是研究需要面對的核心問題。

我們的研究首先將從紙摺立體書的基本結構開始，分析其設計的原理，並推廣至在樂高積木上實現的條件。接著，以提出的立體書基本結構為底，我們開發一套系統將輸入影像透過分析，並利用合理的樂高磚排列來生成合理的結構。除了考量到生成出的結構可以正常排列，不產生非預期的碰撞，我們也必須考量到整體的結構需要擁有一定的穩定性與組裝的方便性。

## 1.3 論文貢獻

我們的貢獻有：一，我們提出了三個樂高立體書的機制，透過公式分析合理的樂高立體書之結構；二，我們提出一套完整的演算法，將二維的輸入影像

轉化為樂高立體書結構；三，我們提出一套以模擬退火模型為基礎的優化目標函式與優化流程，使結構趨於更容易組裝且更穩定的狀態。

#### 1.4 論文章節架構

在後續章節中，我們將在第二章介紹與本論文相關的研究，在第三章簡單的介紹立體書來將立體書從紙的領域推廣到樂高的建構，第四章將介紹並分析我們提出的樂高立體書機制，第五章為主要的系統流程與研究方法，第六章為實驗結果呈現與分析的比較討論，第七章為結論與未來展望。



## 二、相關研究

我們的研究主要聚焦在樂高結構與紙摺立體書，同時為了將兩者結合，我們也需要研究如何利用各種不同種類的樂高磚堆疊成可開闔、伸縮等可動型的結構。故我們在相關研究分別參考了樂高結構、立體書摺疊結構與連動結構的相關領域之研究。

### 2.1 樂高積木的建構

樂高的組合方式需考量到其積木的設計與組合的穩定性，並針對目標的幾何複雜度與顏色等限制做出考量。Testuz 等人[Testuz et al. 2013]提出考量整體結構的穩定度分析的組裝方式，以隨機起始的方式結合相鄰方塊，直到產生出可以組合的樂高磚塊模型，透過分析由樂高磚塊為節點，以邊表示節關係的圖，找出生成的結構中較為脆弱的部分，修正該脆弱節點附近的磚塊組合。在立體模型方面，Ono 等人[Ono et al. 2013]提出簡單明瞭的穩定堆疊作法，藉由合併策略，判斷方塊是否需與相鄰方塊合併。Zhang 等人[Zhang et al. 2015]更進一步考量顏色，經由對輸入顏色模型，進行重新取樣、顏色量化，並考慮整體組成及對稱性，產生立體彩色樂高成品。在穩定性中，最值得注意的是 Legolization: Optimizing LEGO Designs [Luo et al. 2015]，除了探討樂高組合之外，更進一步探討樂高組裝後的穩定度，藉此能組合成不需要以膠水黏接大型樂高造型。

在樂高組合的特色上。清華大學 Kuo 等人探討以樂高拼出 2D pixel art 的方法 [Kuo et al. 2015]，著重在顏色造型的組合，並善用樂高顏色和外型的限制重現復古像素風格。Hong 等人[Hong et al. 2016]以平衡為研究重點，考量內部的配重，透過挖空方塊模型的內部調整重心，經過最佳化後，即使是只有單腳站立的模型也能穩定放置。重現輸入模型的輪廓造型也是樂高組裝的重點，Yun 等人[Yun et al 2017] 使用 Silhouette fitting 的方式移除多餘的方塊，並考量穩定度、對稱性、磚塊使用效率。以上研究，主要使用樂高基本磚，尚未使用其他的樂高關節零件，

這也是我們希望在這一系列計劃處理的問題。

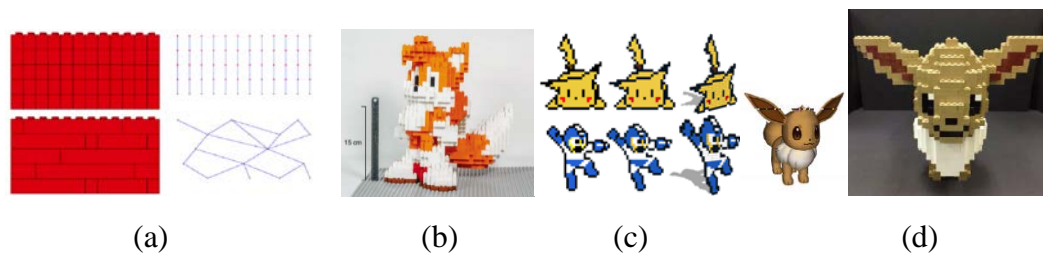


圖 2 樂高研究。(a)組合方式與對應的結構表示，摘自[Testuz et al. 2013]。(b)大型樂高結構，摘自[Luo et al. 2015]。(c) 2D pixelart lego，摘自[Kuo et al. 2015]。(d) Silhouette fitting 的成果，摘自[Yun et al 2017]。

## 2.2 立體書折疊結構

立體書(或卡片)很早就圖學領域就開始研究，Lee 等人[Lee et al. 1996]將紙張利用簡單的幾何拼貼來建構多種風格的立體卡片，同時該研究也設計一輔助設計的系統幫助使用者看到立體卡片開與闔的狀況。Glassner 整理分析主要的折疊架構，並提出互動立體書的設計[Glassner 1998]。Iizuka 等人[Iizuka et al. 2011]也設計了一個提供 V 型摺疊機制與平行折疊機制的立體卡片輔助設計系統。Li 等人設計在單一紙張以折疊和切割的方式[Li et al. 2010]，製作出複雜的 3D 模型。相對 Li 的研究著重於圖 3(a)中類似 2.5D 的高度起伏，Ruiz Jr.等人則透過利用複雜的折疊結構還原輸入模型的外觀[Ruiz Jr. et al. 2014]，圖 3 (b)的卡車由多個結構彼此交錯來組出外觀。更進一步，Xiao 等人的研究[Xiao et al. 2018]提出了改良版的 V 型機制以及具備滑動結構的滑動(Slide)機制，並考慮到平行折疊結構的變化與變形機制，其展開的成果可呈現更多表面細節與貼圖。以上的研究整合了立體折疊結構的基本技巧與進階變化，作為本研究應用樂高積木實現的參考依據。

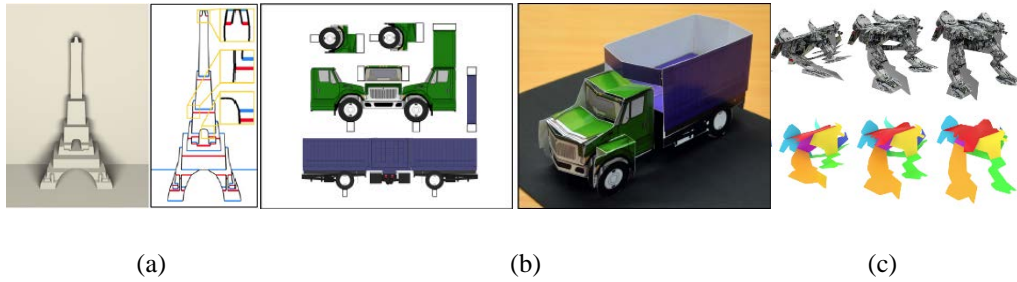


圖 3 立體卡片的相關研究。(a) 以巢狀 step fold 構成，摘自[Li et al. 2010]。(b) 以複合折疊結構構成，摘自[Li et al. 2010]。(c) 以變形折疊結構構成，摘自[Xiao et al. 2018]。

### 2.3 連動結構

在立體書的基本結構之外，如何以樂高完成連動的關係，並設計進階的變化用以擴充創作上的可能性，瞭解機械連動機制與連動關係為重要的相關研究。Zhu 等人在 2012 發表 Motion-Guided Mechanical Toy Modeling[Zhu et al. 2012]，使用者輸入一個要製作動作的模型，並且指定各部位的動作，每個部位的動作形式及軌跡決定了每個單獨可動部件的基礎動作結構，為了符合輸入的目標運動曲線，尋找合適的滑輪或曲軸，對齊驅動軸所在的位置作為初步設計，另需避免部件的互相衝突以及考慮到所使用的部件種類與部件數量等限制對整體結構進行最佳化。其優點為透過單一動力來源的複雜動作，可指定每個部件的動作軌跡。延伸到樂高需考量由於樂高沒有不對稱齒輪，難以透過齒輪形狀修正動作。另外，Coros 等人[Coros et al. 2013]則利用系統會產生出齒輪組以連接每個單獨的部件，達到以符合每一個可動部件的輸入軌跡曲線，其優點為大致上使用對稱的齒輪，比較適合使用樂高齒輪，沒有中央驅動軸，齒輪配置較為自由。此外，避免零件相互卡住也是研究議題之一，我們將參考 Koo 等人的研究[Koo et al. 2014]，在給定一個簡略的模型，並指定模型中部件間的關係，像是部件間的關節類型與目標位置等。這個系統會考慮到兩個部件目標狀態之間的對應，以及支撐力的對應，透過修正部件的細節形狀，確保部件能夠裝進目標位置，並且計算關節的位置及長度，讓部件能夠順利滑動，避免卡在另一個部件上。考慮到有體積的部件在轉



動或移動時所需的空間及關節類型，可應用於樂高方塊的旋轉及移動。

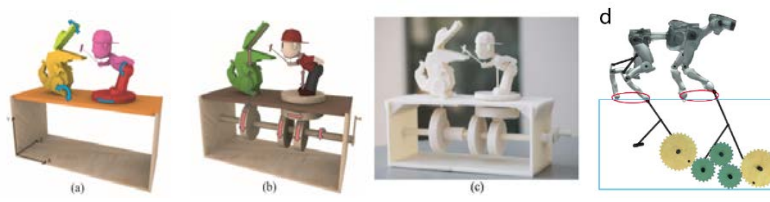


圖 4 連動機構設計的相關研究。(a)(b)(c)分別為指定運動形式、經演算法設計出的機構、以及製造實體結果。摘自[Zhu et al. 2012]。(d)為另一個以齒輪組設計的機構做法，摘自[Coros et al. 2013]。



### 三、樂高立體書的基本介紹

#### 3.1 立體書之術語

紙立體書是透過將不同的紙張彼此相黏後而產生的。使用者可以將紙立體書做開闔的動作，來讓立體書在展開時呈現立體的結構。如圖 5 所示，紙立體書在結構上由一個多邊形紙張(稱做 patches)的集合組成，多邊形紙張彼此透過相連的線段(稱做樞紐 hinges)相黏著。所有的多邊形紙張需要坐落在兩片大的多邊形紙張間(稱做 ground 與 backdrop)，且這兩片大的多邊形紙張連接的線段稱為 central fold。Ground 與 backdrop 所夾的夾角稱為 fold angle。

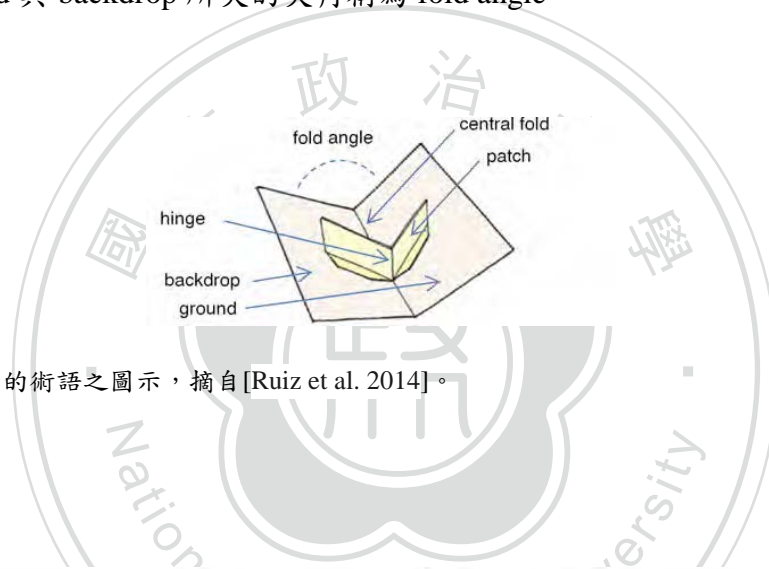


圖 5 紙立體書的術語之圖示，摘自[Ruiz et al. 2014]。

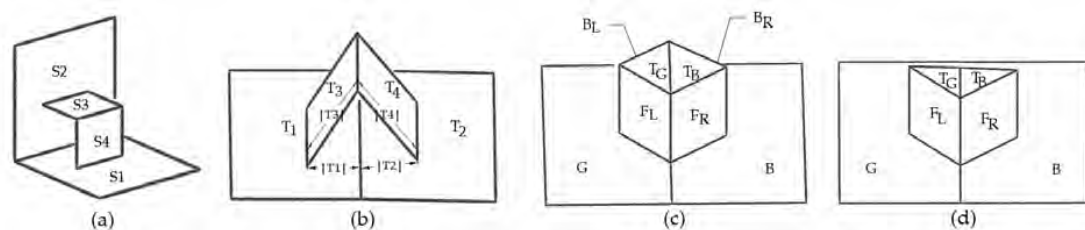


圖 6 各種類型的紙立體書機制：(a) 階梯型 (b) 帳篷型 (c)(d)不同型態的 V 型，摘自[Ruiz et al. 2014]。

#### 3.2 紙立體書的機制

紙立體書在發展上已有一段時間，相關研究[Ruiz et al. 2014]整理出紙立體書有數種不同的排列機制，接著在底下介紹。

### 3.2.1 Step-fold 階梯型機制

圖 6(a)為階梯型機制，透過將 ground 與 backdrop (S1 與 S2) 兩張紙闔成垂直的狀態，使得中間的 S3 與 S4 兩張紙形成一個如同階梯的結構，這樣的機制可看出 S1 與 S3、S2 與 S4 互為平行且兩組紙互相垂直。

### 3.2.2 Tent-fold 帳篷型機制

圖 6(b)為帳篷型機制，透過將 ground 與 backdrop (S1 與 S2) 兩張紙展開，使得中間的 T3 與 T4 形成一似帳篷的結構，此結構必須確保 T3 與 T4 的長度相等且兩紙底部與 central fold 的距離也需相同。

### 3.2.3 V-fold V 型機制

圖 6(c)與(d)為 V 型機制，圖(c)將 {B,FR}、{G,FL}、{FL,TG}、{FR,TB}、{FL,RT,BR,BL}相黏，使得紙張在展開時，FL 與 FR 從正面視角看過去會形成一個 V 型結構，同時其連接到的其他面也會隨之移動至適合的位置來產生一個菱形柱體的外觀。圖(d)就如同圖(c)，不過移除了後面的兩張紙 BL 與 BR，最後產生一個三角柱的外觀。

除了以上提到較基本的機制，紙立體書在近年的研究[Xiao et al. 2018]也有提出更為複雜的結構，例如 box-fold,改良版的 V-fold 與 Slide 等機制，種種的紙立體書機制之驗證已經在提出的研究中被證實，但這些機制因為複雜的限制，要套用到樂高上的可行性尚未被證實，故目前的樂高立體書研究會先以基本的機制為主。

## 3.3 樂高磚塊厚度與比例限制

雖然紙立體書的研究因為紙張本身非常薄的關係，可以假設紙張厚度為 0 來

降低研究的複雜度，但我們無法忽略樂高磚本身的厚度。以開源的樂高軟體 LDraw 格式作為依據，樂高的長度單位稱為 LDU(LDraw Unit)，在許多 SNOT 的作品當中，巧妙拿捏零件與零件之間的距離，是創造驚奇樂高結構的不二法門。以立體書的例子而言，書頁的外框，是一種交錯的結構，組裝時一方採用對齊 Z 軸(圖 6 中綠色實線箭頭)的卡榫，另一方採用對齊 Y 軸(圖 7 中藍色實線箭頭)的卡榫，兩者組裝的方向互相垂直且對齊，如此的組合需要利用到磚塊各個邊長之間的最小公倍數互相搭配。從圖 7 右圖可知，每兩個橫躺的 1x1 灰色變形基本磚之間，為了要與上方正坐的黑白相間平磚合理的排列，中間所夾的 1x1 薄磚數只能是  $2 + 5 \times n$ ,  $n \in \mathbb{N}$  個。

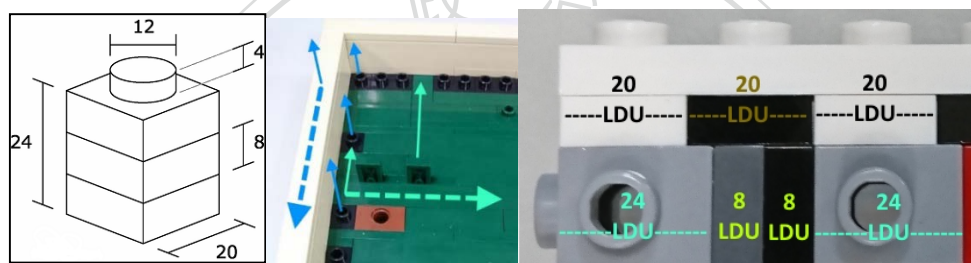
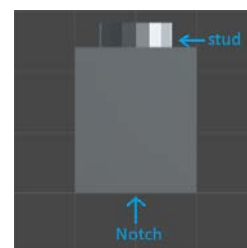


圖 7 樂高長寬比例示意圖。左：基本 1x1 磚的比例。中：樂高立體書比例分析示意圖。右：不同轉向對齊示意圖。

### 3.4 樂高立體書之術語

相較於紙立體書，樂高立體書是透過樂高磚的堆疊而產生的。在結構上樂高立體書由基本磚、薄磚、斜磚、科技磚...等各種磚的集合組成。如右圖，樂高磚彼此間透過樂高特有的螺柱與螺柱孔來相互連接，螺柱在 LDraw 格式的專用術語為 stud，而螺柱孔在 LDraw 中並沒有明確定義，我們以 notch 來稱呼。



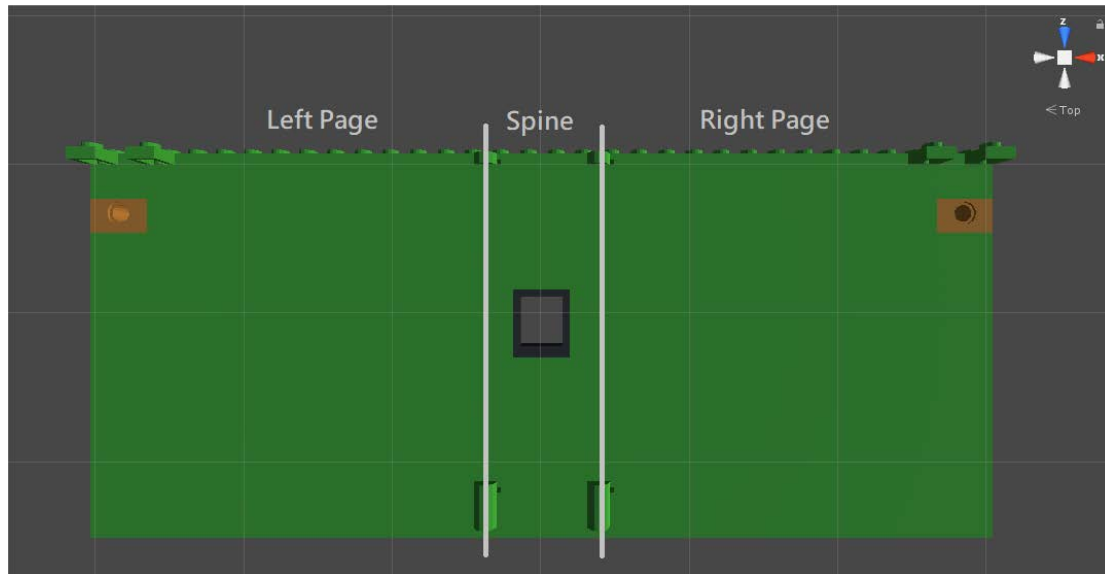
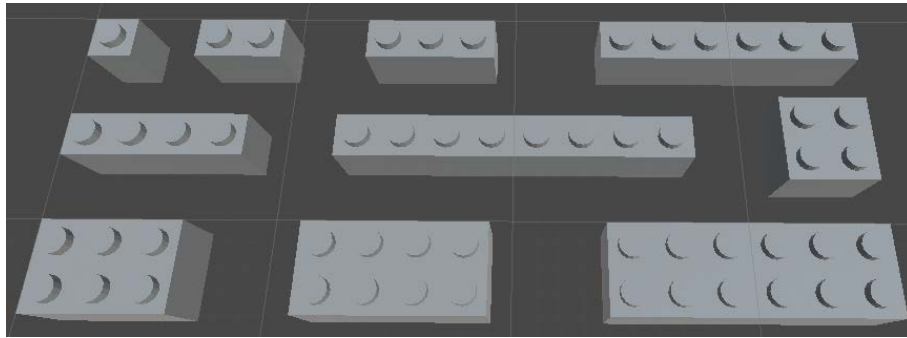


圖 8 書頁示意圖。我們將書頁分成左書頁、書脊與右書頁。

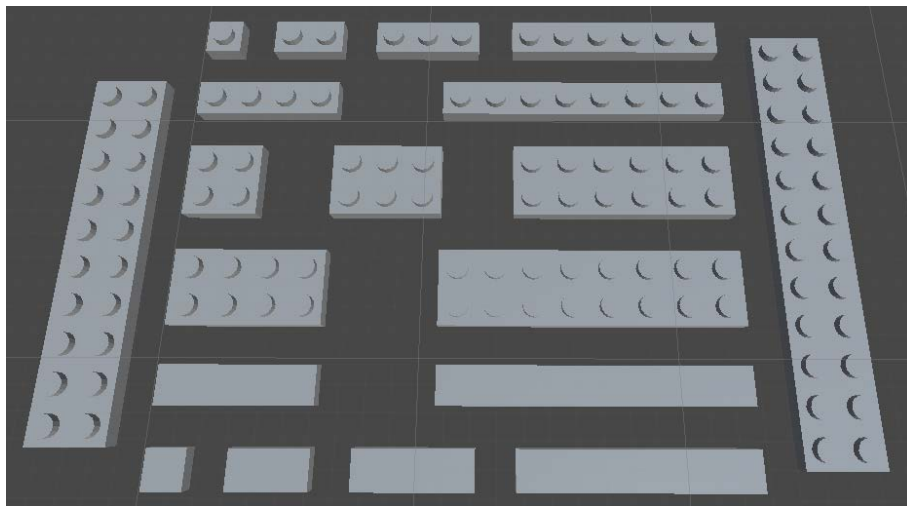
圖 8 為書頁的示意圖，圖上的左右水平軸為  $x$  軸，上下垂直軸為  $z$  軸，從結構平面到視點，且同時表示立體結構高度的軸為  $y$  軸。如前面所提到的內容可知，因為樂高有厚度的限制，我們無法使用單一線段的 central fold 來連接兩面書頁，故我們將書頁分成三部分，分別稱為左書頁、書脊與右書頁。為了達成可開闔的目標，左書頁與書脊間、書脊與右書頁間需要透過樞紐磚(如圖 17) 做連結，使得開闔過程中，左書頁與右書頁可以有不同的旋轉角度，且不會與書脊分離。

### 3.5 系統使用的樂高磚集

圖 9 為系統使用的樂高磚集，圖(a)為基本磚集，圖(b)薄磚與平滑磚，這兩個集合多用於生成書頁與書中的立體結構，而平滑磚則用於書框的生成(6.3 節)。圖(c)為變形基本磚集，圖(d)為樞紐磚集，這兩個集合用於生成可動結構與書框的底座；圖(e)為科技磚的類別，由於我們系統使用到的科技磚類型較多，故每一種類別只列出一種，科技磚多用於生成書頁與書中立體結構的可動連結磚結構。



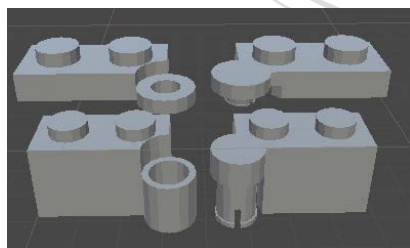
(a)



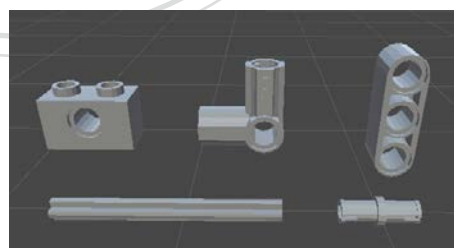
(b)



(c)



(d)



(e)

圖 9 系統使用的樂高磚示意圖。(a)基本磚；(b)薄磚與平滑磚；(c)變形基本磚；(d)樞紐磚；(e)科技磚。科技磚第一排從左到右分別為帶插銷孔的1×2基本磚、L型連接磚(Connector#6)與孔樑磚，第二排則為十字軸磚與插銷磚。科技磚因種類繁多，在圖中僅列最常使用的類型。

## 四、樂高立體書之機制分析

在這個章節，我們將第三章提到的數種紙摺立體書機制推廣至建構樂高的結構，並提出了(1)帳篷式機制、(2)階梯式機制和(3)V型機制(圖 10)。如第三章提到的，樂高立體書的書頁分為三個部分，在立體書展開時三個部分將會呈現平行(如圖 17)，或是將左右書頁皆往內折 $45^\circ$ ，形成垂直的狀態(如圖 16)。而將立體書闔上時，書頁會完全收合，而中間的立體結構將會呈現如圖 14、圖 19 的狀態。

在以下三個小節，我們將分別討論提出的三種樂高立體書之機制，並解釋樂高的立體結構如何排列並坐落在開闔的書頁當中，最後探討如何計算書頁的大小以包覆住立體的結構。

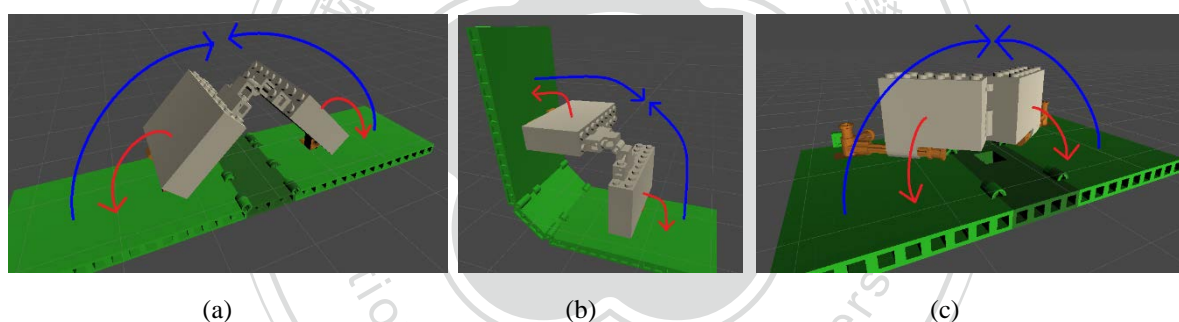


圖 10 樂高立體書的機制示意圖。(a)帳篷式，(b)階梯式 (c)V型，圖中的藍線代表書頁在闔上時移動的方向，紅線代表書中立體結構在闔上時移動的方向。

### 4.1 Tent-fold 帳篷式機制

#### 4.1.1 定義與可折疊性

圖 11 為展開的帳篷型的樂高立體書結構圖，同上一節提到，各個顏色的樂高磚功能與定位與圖 的 V 型結構相同。褐色的科技磚部分從 y 方向，下往上依序是孔樑(beam)磚、十字軸(axle)磚、帶插銷孔的薄磚(plate with peghole)磚，帳篷的主結構，最頂端由樞紐磚接在變形基本磚上，作為兩側結構的連接點。帳篷式立體書在闔上時利用十字軸插銷磚使得 L 型連接磚以及其上方的主要結構沿

著 z 軸旋轉以貼合書頁(如圖 12)。

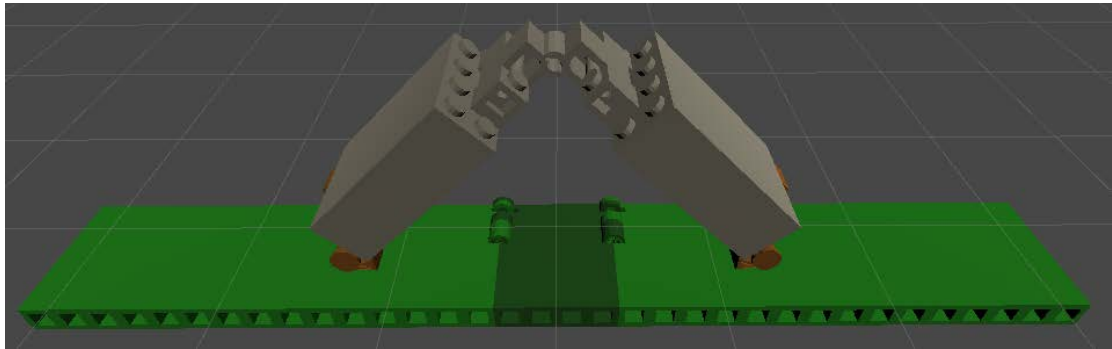


圖 11 展開的對稱式帳篷式立體書結構圖(從後往前的視角)。兩側結構與書脊的距離皆為兩個樂高磚單位寬。

為了確保帳篷型結構的合理性，結構兩側的厚度 (D)與結構高度 (H)必須相同，厚度就如同 V 型結構的介紹所提到，兩者若不相等必然無法闔上；且兩側結構的高度必然要相等，若不相等代表頂端必然不是由兩個樞紐磚組成，代表在開闔時結構根本不會相連。接著可從圖 11 與圖 12 看出，帳篷型結構在展開時結構是往  $x = 0$  之平面與 y 軸做延伸，故與 V 型相同，只需要考量結構是否能正確地闔上。在闔上時需要考量到帳篷型結構之厚度不能超出 D 的範圍；而寬度則可適當地在書頁的高度範圍中做延伸。接著來看書頁的部分，書脊的厚度也與 V 型結構相同，所以只需探討左右書頁寬度與高度的限制。

#### 4.1.2 書頁的寬度 w

可從圖 12 推知，左右書頁的寬度取決於兩個點：(1)beam 磚與書脊的水平距離(圖 12 的 d) (2)帳篷式結構的高度 H，兩者之和不得超過書頁的寬度。將兩者公式化後為：

$$d + H \leq w$$



### 4.1.3 書頁的高度 h

在帳篷式結構下，書頁的高度則較為彈性，因為此結構基本上是往 +y 的方向進行延伸，故增加帳篷型結構的寬度不會如 V 型結構般可能會與連結的科技磚產生碰撞。故在進行設計時，在結構的穩固性之考量下，可以適當地延伸結構的寬度，而此寬度最終不要超出書頁的高度即可。

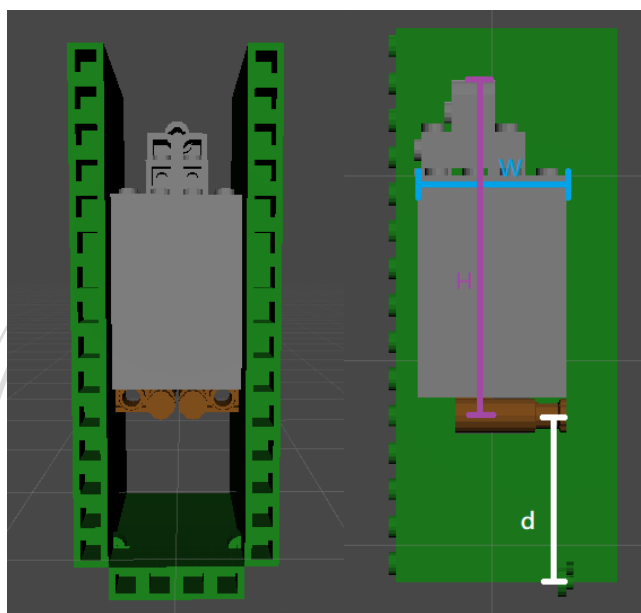


圖 12 闔上的帳篷型立體書結構圖。左圖為前視角，右圖為去除左書頁透視圖(正交視角)。d 為兩側的結構與書脊的距離(此圖為四個樂高磚寬)，H 為結構的 L 型連接磚到頂端的樞紐磚之樞紐的距離。

### 4.1.4 非左右對稱之帳篷式機制

基本的帳篷型結構假設兩邊的結構高度相同、且兩邊結構坐落於書頁上的位置也以書脊的中心線為對稱軸，而我們也經過計算提出了非左右對稱之帳篷式機制。以圖 12 所示，為了確保非左右對稱的兩邊結構能正常地闔上，兩邊(左、右)的結構高度也必須相同：

$$d_L + H_L = d_R + H_R$$

首先先計算結構中固定的部分(以圖 12 為實例參考)，L 型連接磚磚的高度為 40LDU、technic axle 磚為 60LDU，其中兩者相連結的部分為 20LDU，頂端由

樞紐磚接在變形基本磚上，其高度為 $24LDU + 40LDU - 20LDU$ (連結的部分) =  $44LDU$ 。令左右側的結構高度分別為 $Y_L$ 、 $Y_R$ 個一般樂高磚高度(8 LDU)，可得：

$$H_L = 40 + 60 - 20 + 8 \times Y_L + 44 = 8 \times Y_L + 124 ,$$

$$H_R = 8 \times Y_R + 124 ,$$

$$|H_L - H_R| = H' = 8Y' ,$$

接著令左右結構的 L 型連接磚與書脊的水平距離分別為 $X_L$ 、 $X_R$ 個一般樂高磚寬度(20 LDU)，可得：

$$d_L = 20 \times X_L , d_R = 20 \times X_R , |d_L - d_R| = d' = 20X' ,$$

為了滿足 $X'$ 、 $Y'$ 皆需為整數的條件，可得：

$$|X_L - X_R| = 2x' , x' \in N , |Y_L - Y_R| = 5y' , y' \in N$$

也就是兩邊結構與書脊的水平距離相差偶數個基本磚寬度單位，且同持兩邊結構的高度相差 5 的倍數個基本磚高度單位，才能達成非左右對稱的帳篷式機制之合理性。

根據以上的公式，我們可以產出合理的非左右對稱之帳篷結構，如圖 13 所示，可看出兩邊結構頂端之樞紐磚並不在結構的中心線上，而是偏向左邊的。該結構闔上後的狀況則如圖 14 所示，且因為前面提到的公式計算，使得書頁闔上時可以使兩邊結構正確的連接。

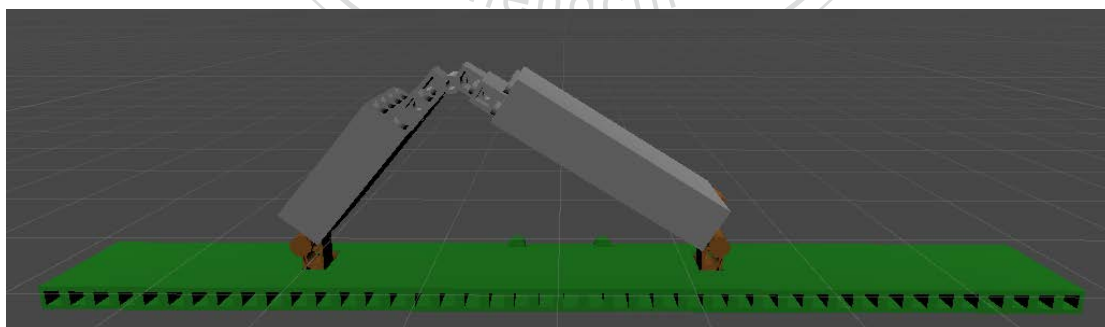


圖 13 展開的非左右對稱帳篷式結構之示意圖(俯瞰視角)。其中左側的結構與書脊的距離為八個樂高磚寬，右側的結構與書脊的距離為四個樂高磚寬，因右側的結構離書脊較近，故其結構高度(L 型連接磚與樞紐磚的距離)需要比左側高 10 個樂高磚單位的高度來達成能正確闔上立體書的條件。

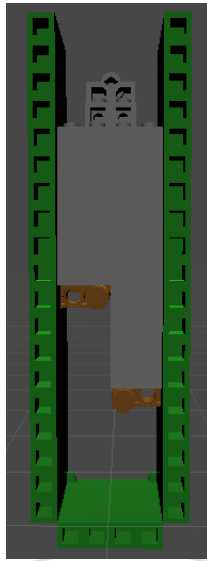


圖 14 將圖 13 闔上後呈現的非左右對稱帳篷式結構之示意圖(正面視角)。



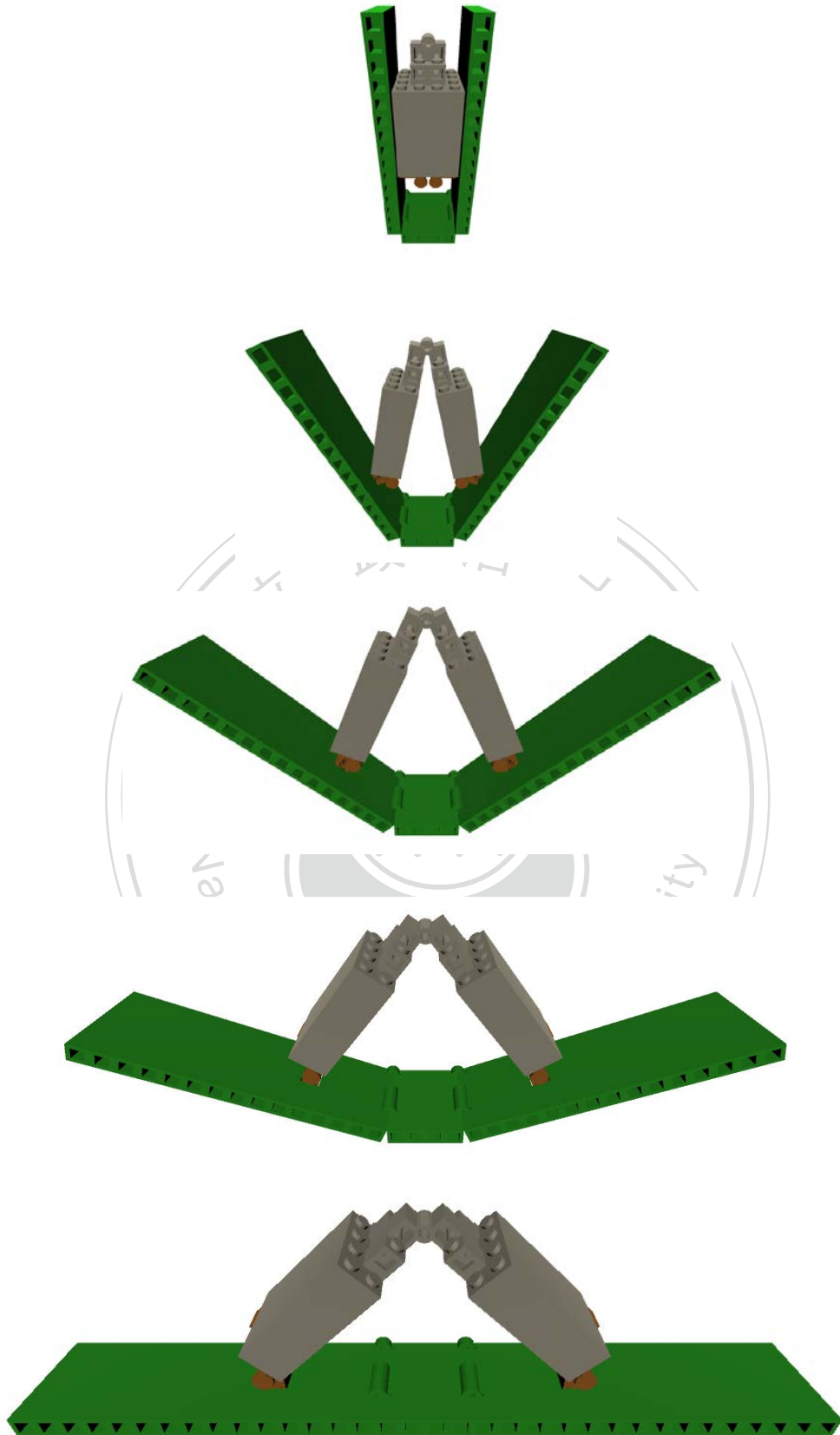


圖 15 帳篷式機制的開闔多角度示意圖。從上到下分別為 $0^{\circ}$ (完全闔上)、 $45^{\circ}$ 、 $90^{\circ}$ 、 $135^{\circ}$ 與 $180^{\circ}$ (完全展開)。

## 4.2 Step-fold 階梯式機制

### 定義與可折疊性

在紙立體書的階梯式結構中，其強調的是將書頁折至夾角 $90^\circ$ 時，中間的兩張紙也形成垂直的立體結構，這個結構需要兩張紙在書頁上的直線與 central fold 的距離以及兩張紙的高度皆相等。將階梯式套用在樂高立體書上時，我們可以將一帳篷型結構，接著透過將左書頁與右書頁皆往中間翻轉 $45^\circ$ ，使得書脊與左書頁、書脊與右書頁的夾角皆為 $135^\circ$ ，來形成如紙立體書般左右書頁垂直的效果(如圖 16)。一般帳篷式的結構在左右書頁垂直的狀況下，結構頂端的樞紐磚夾角必然小於 $90^\circ$ ，為了要使兩側逼近垂直，該結構的高度  $H$  與十字軸插銷磚與書頁的垂直距離  $H'$  之和需要滿足：

$$H + H' \approx d + \frac{2D}{\sqrt{2}},$$

因為樂高有長度與比例的限制，高度的計算與 beam 磚和書脊的水平距離  $d$  互相影響。套用於 4.2.4 節提到的非對稱機制後，可將上式轉變為：

$$H_R + H' \approx d_L + \frac{2D}{\sqrt{2}}, \quad H_L + H' \approx d_R + \frac{2D}{\sqrt{2}},$$

另外，因為結構的高度  $H$  太小的關係，使得該立體書是無法完全展開至 $180^\circ$ 的。以圖 6(a)為例，若將該書頁展開至 $180^\circ$ ，會發現結構完全貼合書頁，不會產生立體的效果，套用至樂高立體書時，展開 $90^\circ$ 度時極接近垂直的結構因為樂高本身有厚度的關係，在展開至 $180^\circ$ 的過程便會因為與書脊產生碰撞而失敗。

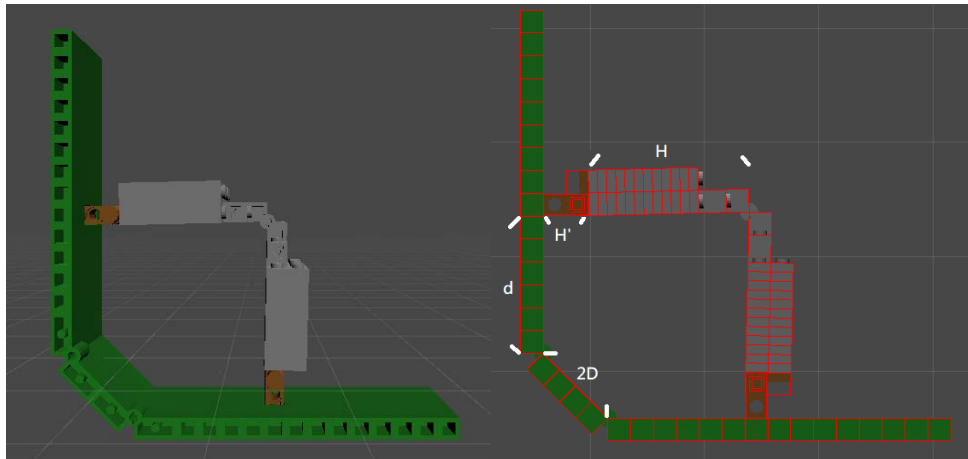


圖 16 展開的 step fold 基本結構示意圖。左圖為透視圖，右圖為正交視角。

階梯式的機制因為本身就是從帳篷式的機制變化(開闔角度調整、立體結構高度限制)而成，故其書頁的厚度、寬度、高度等計算與帳篷式的機制相似。常見的階梯式機制之立體書都是以看向其中一面書頁的視角去做呈現(如圖 3(a)的巴黎鐵塔)，將所有呈現的立體結構內容擺在正面，而背面的立體結構單純用來支撐與連接正面的結構，這樣的設計可以使正面立體結構的寬度與高度有更大的彈性，所能呈現的內容也更多樣化。故我們在研究中也採用類似的型式，我們將階梯式結構的立體書之兩側分為背面與正面，僅以正面的部分來生成符合輸入的樂高磚結構，而背面則主要以支撐與連接的功能為主。

### 4.3 V-fold V 型機制

#### 4.3.1 定義與可折疊性

圖 17 為展開的 V 型的樂高立體書結構圖，綠色樂高磚為書頁的部分；灰色樂高磚為立體書的內容部分；而褐色科技磚則為將書頁與立體書連接起來的部分，該結構由一個 L 型連接(connector #6)磚與數個十字軸磚組成。如圖 18，灰色樂高磚的底部為帶插銷孔的薄磚，其釘孔與連接 L 型連接磚的十字軸磚形成一個類似滑軌的結構，讓立體書在闔上的過程中，整個 V 型的結構可以沿著區域性的 x 軸旋轉。同時，隨著立體書闔上，立體結構受到書脊的壓迫，整體結構會沿

著 x 軸往 L 型連接磚移動並貼齊，結構十字軸磚與帶插銷孔的薄磚在此就產生了能讓結構旋轉與平移的滑軌功能。最終立體書闔上將會呈現如圖 19 的狀態。

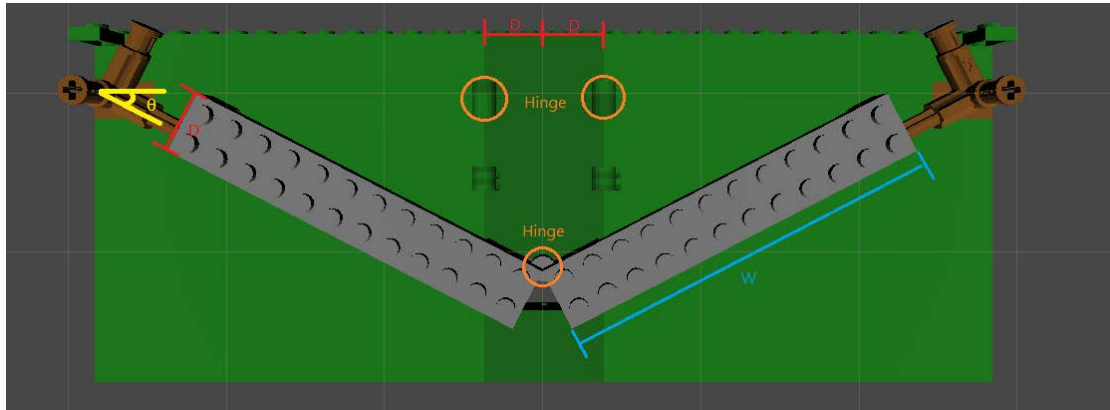


圖 17 展開的 V 型樂高立體書結構示意圖（從上往下的視角）。其中綠色區域的樂高磚為書頁部分，棕色區域為連接書頁與立體結構的科技磚，而灰色區域為立體結構呈現的部分。橘色的圈為使結構正常旋轉的樞紐磚。

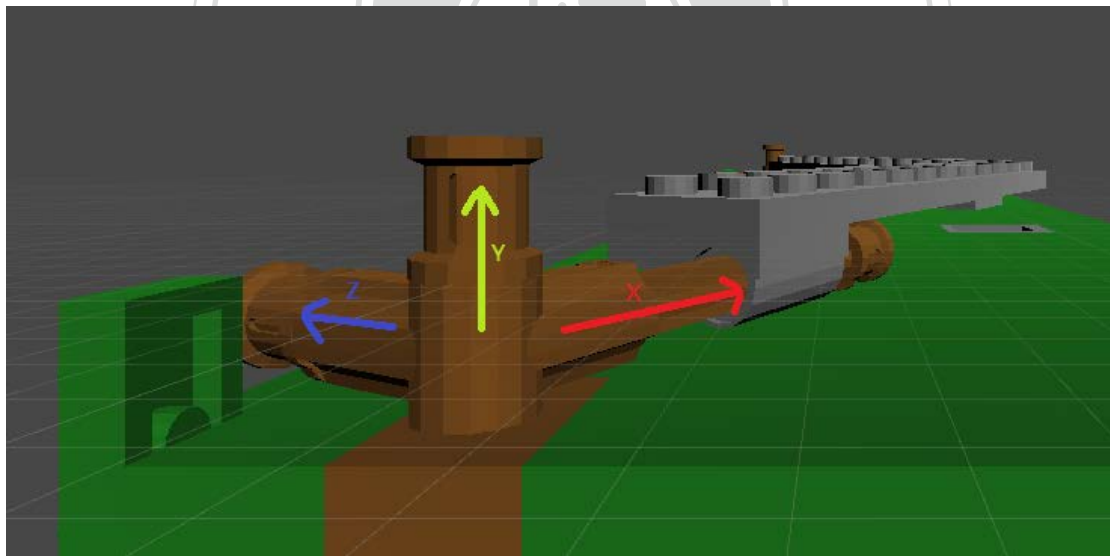


圖 18 V 型結構左側的科技磚結構。透過棕色的十字軸磚與灰色的帶插銷孔的薄磚連結，來達成類似於滑軌的結構來讓立體書正常地開闔。其中箭頭為結構區域性的座標軸。

為了確保 V 型結構的合理性，V 型兩側的結構需成對稱，也就是兩側的結構厚度 (D) 與 寬度 (W) 必須要相同，若兩者厚度不同，兩邊 V 型結構勢必無法在闔上時收進書脊中；若兩者寬度不同，闔上時結構中心不會在書脊中心上，

一樣無法正常闔上。接著為了驗證 V 型結構的可折疊性，從圖 17 可看出，樂高立體書在展開時，V 型結構的內容是朝著  $+y$  方向增加，結構上方的樂高磚排列不會對展開的書頁造成影響，所以在考量 V 型結構的可折疊性，我們可以忽略展開的部分，僅關注樂高立體書是否能正常的闔上。如圖 17 與圖 19 所示，在闔上時第一要素為 V 型結構上方的樂高磚排列不得超出  $D \times W$  的底座範圍，厚度的範圍超出就會同本段前面所提到，寬度超出則會與該側的滑軌結構或是 V 型結構之另一側產生碰撞，也是不合理的。接著，V 型結構的厚度  $D$ 、寬度  $W$ 、以及結構上方的內容高度也會影響到書頁的厚度  $depth$ 、寬度  $w$  與高度  $h$ ，接著在本節後段進行探討。

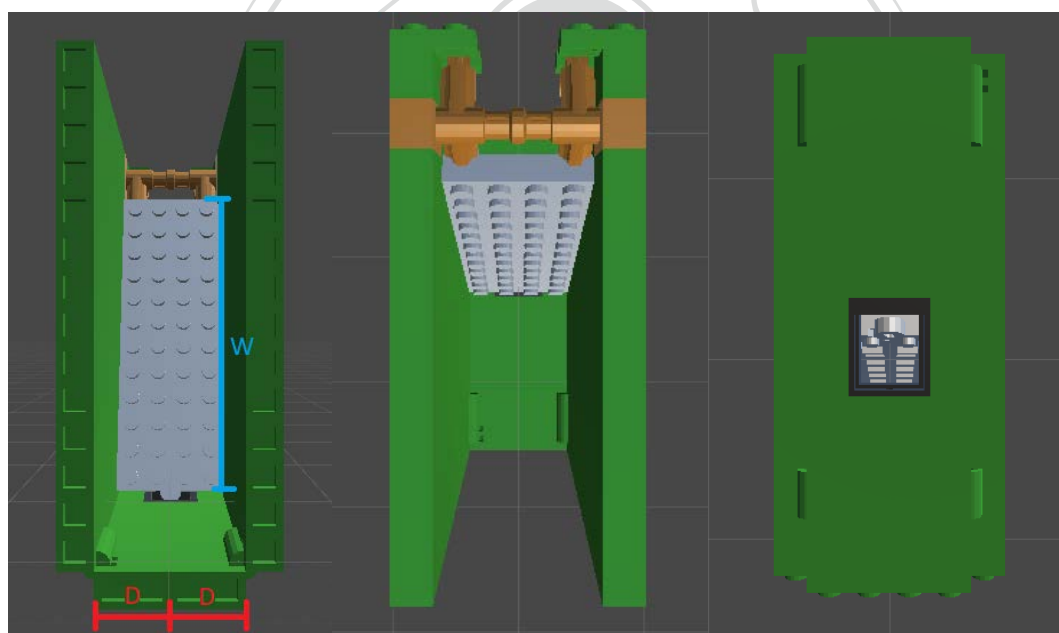


圖 19 闔上的樂高立體書結構示意圖。左圖為前視角；中間圖為上視角；右圖為底部視角，我們透過引入窗框的樂高磚來避免樞紐磚之轉軸與書脊產生物理碰撞。

#### 4.3.2 書脊的寬度 $depth$

可從圖 19 推知，書脊的寬度取決於 V 型結構中兩面內容的厚度，以圖 17 的結構為例子，立體書兩邊內容結構的底部為  $2 \times 12$  的薄磚，其厚度  $D =$



40 LDU、寬度  $W = 240$  LDU，為確保整個立體書能闔上，書脊的寬度

$$\text{depth} = 2D,$$

### 4.3.3 書頁的寬度 $w$

書頁的寬度取決於兩項因素：連結磚結構的位置以及整體立體結構的全域性水平分量，兩者皆受使用者給定的結構寬度  $W$  與 V 型機制的角度(圖 17 之 $\theta$ )影響。首先我們需要根據給定的 $W$ 、 $\theta$ 來計算連結磚結構的位置才能計算立體結構的全域性水平分量。在前面 4.1.1 節我們提到當立體書闔上時，立體的結構需要貼齊 L 型的連接磚(如圖 20)，故我們令連接磚與書脊的距離為 $d$ ，可得：

$$d = \text{Round}\left(\frac{(W + 1.5)}{\cos(\theta)}\right)$$

由於樂高的比例限制， $d$  必須要是整數，故 $\theta$ 在計算完 $d$ 以後會有些許的變更以符合 $d$ 。以圖 20 為例，令 $W$ 為9， $\theta$ 為 $30^\circ$ ，計算完 $d$ 後回推 $\theta$ 會約等於 $31^\circ$ 。接下來我們要計算立體結構的全域性水平分量，我們先看圖 20 的 $\Delta ABC$ ，假設 A 點的樂高磚結構高度  $h(A) = h_A$ ，可得全域的水平分量為  $h_A \sin \theta + (W + 1.5 - \overline{AD}) \cos \theta$ 。接著我們計算結構中 $D \times W$ 個點(在以下公式中令 $\Phi$ 為每個點的節點與對於連結磚的相對座標)的全域水平分量取最大值，最後在與 $d$ 取其大值作為書頁的寬度 $w$ 。

$$w \geq \text{Max}(d, \text{Max}_\Phi( h(\Phi) \sin \theta + (W + 1.5 - |\Phi|) \cos \theta))$$

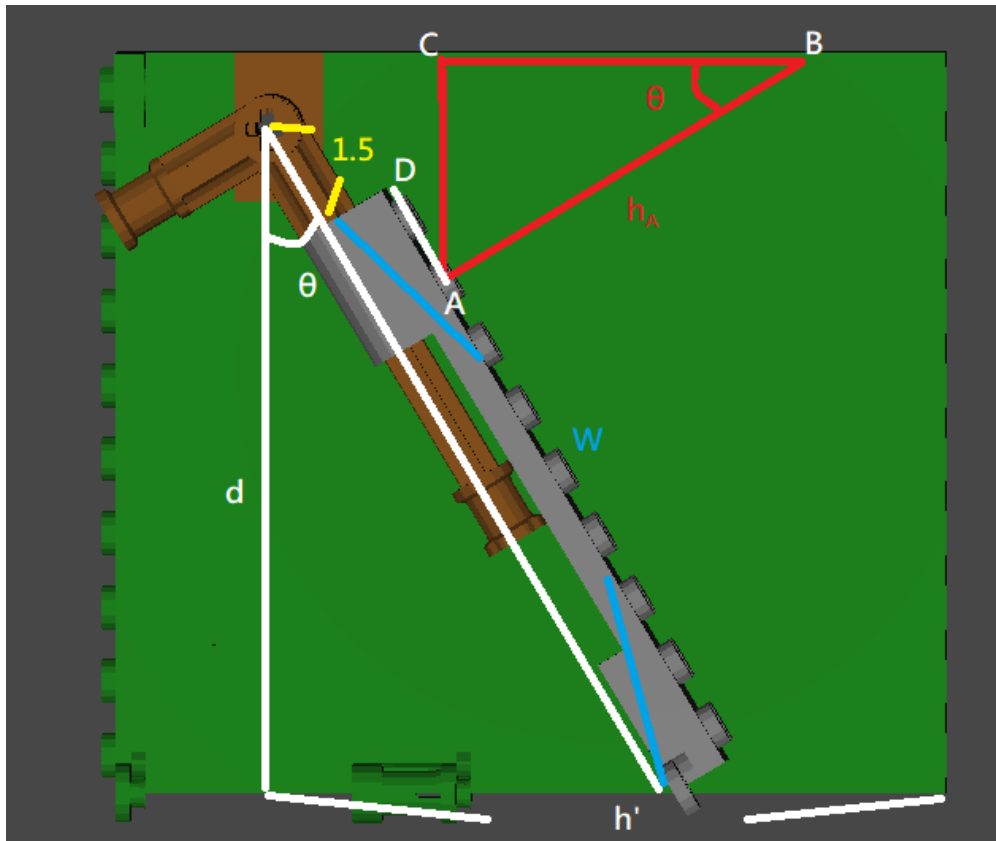


圖 20 去除左書頁之闔上的立體書結構圖 (從左往右的視角)。

#### 4.3.4 書頁的高度 $h$

因為 V 型結構整體皆是在連結書頁與立體結構的十字軸磚下，故十字軸磚上方的高度通常是固定的。我們假設十字軸磚與立體書的最下方之垂直距離為  $h'$ 。接著我們計算結構全域的垂直分量，可得：

$$h' \geq h(\Phi) \cos \theta + |\Phi| \sin \theta$$

而為了包覆整個 V 型結構(包含十字軸磚上方為了固定連結磚所用的 plate 磚)，書頁的整體高度

$$h \geq h' + 4,$$

將會比  $h'$  多四個一般樂高磚的高度單位。

#### 4.3.5 避免合頁磚與書脊產生碰撞

為了防止立體書結構闔上時，立體書內容中心的樞紐磚之轉軸處可能會與書脊產生物理碰撞，在生成時我們需要一個空心的磚來避免此狀況。如圖 19 右所示，我們引入一個窗框外觀的樂高磚專門處理此問題。該磚需要擺放在立體結構的中心處，故其必然位在書脊中，其  $y$  座標位置可以透過計算樞紐磚的全域垂直分量得出

$$y_{\text{hinge}} = \text{Round}((W + 1.5) \sin \theta),$$

最後將窗框磚擺放在書脊中從連結磚的  $y$  座標往下  $y_{\text{hinge}}$  單位處。



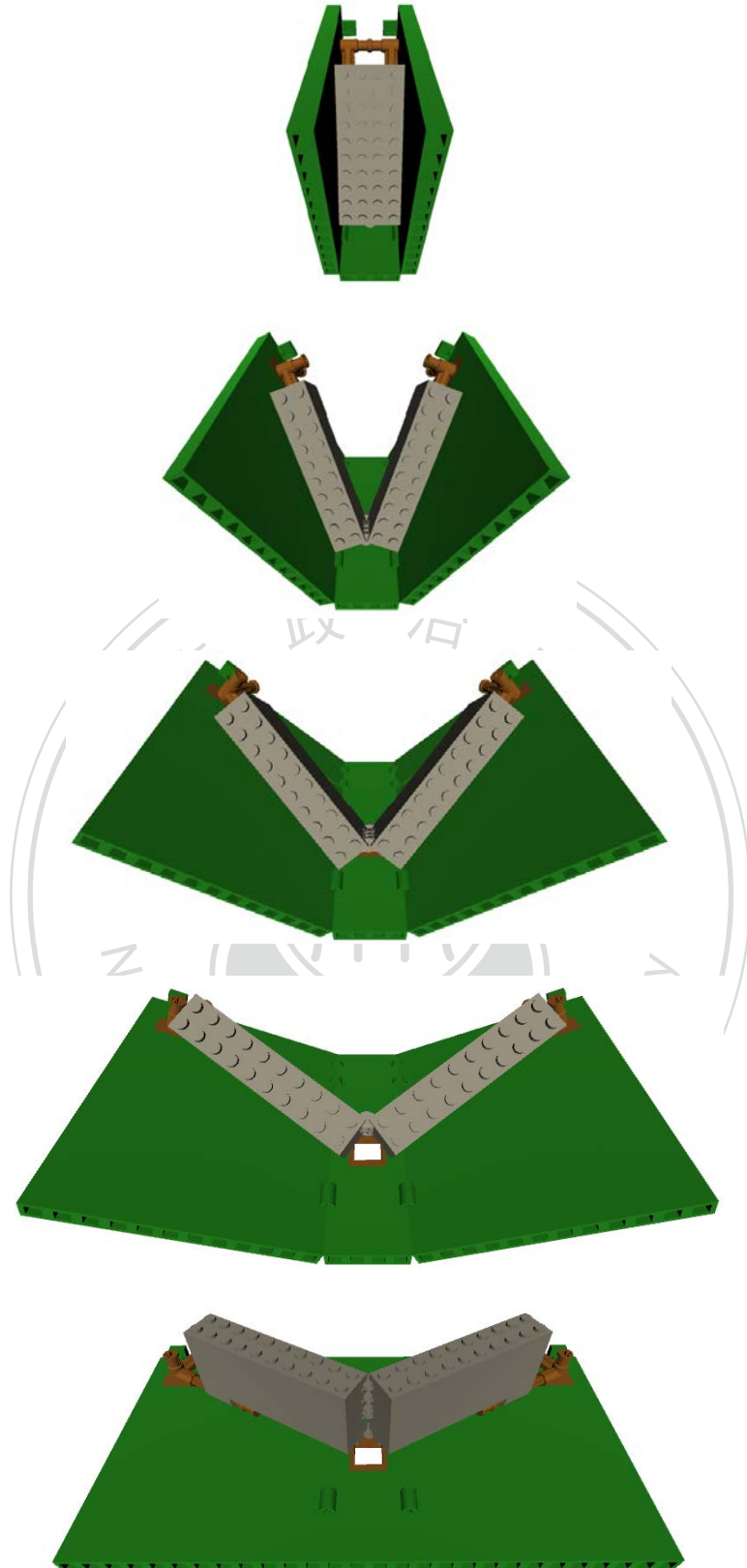


圖 21 V 型機制的開闔多角度示意圖。從上到下分別為 $0^{\circ}$ (完全闔上)、 $45^{\circ}$ 、 $90^{\circ}$ 、 $135^{\circ}$ 與 $180^{\circ}$ (完全展開)。

## 五、研究方法

此章節將會介紹樂高立體書的自動生成系統之建成，包含前置作業、輸入方式、立體書生成之演算法等內容。

### 5.1 系統流程

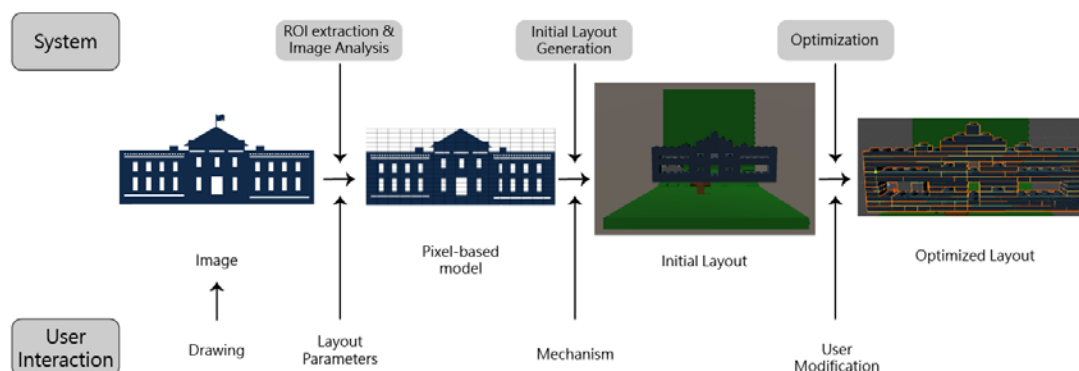


圖 22 系統流程圖。首先我們將輸入的影像透過感興趣區域的擷取以及影像的分析來產生出輸入影像的像素模型，接著將像素模型根據使用者要求的機制產生出初始的結構設計，最後在我們設計的優化演算法之引導下，隨機地調整樂高的設計來使結構擁有良好的穩定性與複雜度。

圖 22 為系統的流程圖。在相關的紙立體書研究中，通常假設紙的厚度極小甚至為零，故可以將大量的紙張結構夾入紙立體書的書頁當中。而樂高的立體書因受限於帶有厚度的立體樂高結構也須全部包覆在闔上的書頁當中，故書頁內的結構厚度通常不會很大。以圖 1 為例，該立體書的結構厚度其實只有兩個樂高磚大。故我們令系統的主要輸入為二維的影像，其中包含影像中需要用樂高磚組出的部分、每個區域的顏色等。同時為了達成電腦輔助設計樂高立體書的系統，使用者也可在系統中進行簡單繪畫，並以其做為系統的輸入。

系統的輸出為一個樂高結構的設計 $\beta$ ，由樂高磚的集合 $\beta = \{B_1, \dots, B_n\}$ 組成，其中每個樂高磚間彼此透過適當的堆疊與連結來形成完整的結構。每一個樂高磚 $B_i = \{p, q, t\}$ 組成，其中  $p$  代表樂高磚的位置， $q$  代表樂高磚的三維空間旋轉(四元數)， $t$  則為樂高磚的類別。

首先，我們將輸入的影像進行影像分析的步驟，將影像解析成系統生成所需要的屬性，例如目標的形狀輪廓、對稱性和每個區域的顏色...等。使用者可以決定生成結果之立體書的各種屬性(結構的長度、厚度、和高度以及書頁的長度、厚度和高度)以及選擇要使用何種類型的立體書機制去生成結構和連結磚。接著系統會根據使用者給定的參數以及分析後的影像屬性，利用大量的 1×1 薄磚來生成初始的樂高設計，接著系統透過優化的過程以及使用者的喜好來調整設計，終至生成最後的樂高結果之設計。

## 5.2 系統建置的前處理

### 5.2.1 樂高模型檔匯入

研究以 unity3D 引擎做為開發平台，在研究開始進行前首先要將開源的樂高軟體 LDraw 所提供的模型檔引入至 unity 中，圖 23 為 LDraw 提供的 1×1 基本磚模型檔，其中開頭為 0 的行是描述模型的資料，而開頭為 1~4 的行則是實際記錄模型網格的資料，而程式完成的部分就是能讀取這些資料並正確地生成這些模型網格，生成的結果就如圖 8 開始的各種立體書結構中之樂高磚。

```
0 Brick 1 x 1
0 Name: 3005.dat
0 Author: James Jessiman
0 !LDraw_ORG Part UPDATE 2002-03
0 !LICENSE Redistributable under CCAL version 2.0 : see CReadme.txt

0 BFC CERTIFY CCW

0 !HISTORY 2001-10-26 [PTadmin] Official Update 2001-01
0 !HISTORY 2002-05-07 [unknown] BFC Certification
0 !HISTORY 2002-06-11 [PTadmin] Official Update 2002-03
0 !HISTORY 2007-05-07 [PTadmin] Header formatted for Contributor Agreement
0 !HISTORY 2008-07-01 [PTadmin] Official Update 2008-01

0 BFC INVERTNEXT
1 16 0 24 0 6 0 0 0 -20 0 0 0 6 box5.dat

4 16 10 24 10 6 24 6 -6 24 6 -10 24 10
4 16 -10 24 10 -6 24 6 -6 24 -6 -10 24 -10
4 16 -10 24 -10 -6 24 -6 6 24 -6 10 24 -10
4 16 10 24 -10 6 24 -6 6 24 6 10 24 10

1 16 0 24 0 10 0 0 0 -24 0 0 0 10 box5.dat

1 16 0 0 0 1 0 0 0 1 0 0 0 1 stud.dat
0
```

圖 23 開源樂高軟體 Ldraw 的 1×1 基本磚模型檔。

### 5.2.2 樂高磚的資料結構

我們令引入的所有樂高磚之集合為  $T = \{T_1, \dots, T_n\}$ ，其中元素皆為單一類別，每一個類別  $T_i = \{nb_i, tb_i, hb_i\}$  代表一種類別的磚之屬性。每個樂高磚都有基本的幾何屬性：(1)樂高磚之按座標軸排列的邊界框(axis-aligned bounding box)；(2)螺柱的位置；(3)螺柱孔的位置。我們在實作中令基本的幾何屬性  $nb_i = \{b, s, n\}$ ，第一項  $b$  為該類別磚的 aabb(axis-aligned bounding box)，包含邊界框中心位置與邊界框的大小等屬性；後二項  $s = \{vp_{s1}^i, \dots, vp_{sM}^i\}$ ， $n = \{vp_{n1}^i, \dots, vp_{nM}^i\}$  分別代表樂高磚的一般螺柱屬性與螺柱孔屬性。其中每一個向量對  $vp = \{\text{pos}, \text{dir}\}$  包含位置與方向，例如一般  $1 \times 1$  磚中，朝上的螺柱為  $\{\text{pos} = (0, 0, 0), \text{dir} = (0, 1, 0)\}$ ；而有些類別的磚並不只有基本的幾何屬性(如科技磚或樞紐磚...etc)，故我們需要其他的結構來儲存特殊類別的磚。實作上我們令類別中的科技磚屬性  $tb_i = \{ts, tn\}$ ，分別紀錄科技磚的螺柱(插銷 pin 與十字軸 axle)以及螺柱孔(插銷孔與十字軸孔)之位置與方向；並令類別中的  $hb_i$  紀錄樞紐磚中樞紐的位置與方向。

### 5.2.3 樂高磚的物理模擬引擎

根據前節引入了樂高磚的模型網格與建立了資料結構後，因為樂高結構是由樂高磚間彼此堆疊而成，故在樂高磚的堆疊過程中需要建立屬於樂高磚的物理碰撞偵測來確認複雜旋轉下的樂高磚擺放之位置是否合理。

unity3D 引擎本身有提供網格碰撞器(Mesh Colliders)的功能，但若要其產生碰撞的偵測，引擎會將物體轉換成全凸面(convex)的碰撞體並且最多只支援 255 個三角面的全凸面網格碰撞器。而樂高磚因本身複雜的結構，通常都有數百到數千個三角面，故引擎本身提供的網格碰撞器不列入我們研究的使用範圍。

因為前面提到的原因，在研究中設計了樂高之簡單的物理碰撞偵測。如演算法 2 樂高的物理碰撞偵測之演算法。演算法 2 所示，我們在指定的位置與旋轉

情況下生成樂高磚物件時，我們(1)計算這個物件的邊界框是否有與其他樂高磚的邊界框相交，若有相交則代表此樂高磚在該位置與旋轉條件下會與其他物件產生碰撞。若未發生相交狀況，則繼續(2)針對每一個螺柱，檢查螺柱的邊界框是否有與其他物體相交，若有產生相交，則檢查該碰撞體的螺柱孔是否有與該螺柱相同的位置與方向(利用演算法 1)，在實作中我們為了避免浮點數精確度誤差的問題，故令演算法 1 的  $posEpsilon = 10^{-1}$ ， $dirEpsilon = 10^{-3}$ 。若有滿足條件的螺柱孔，代表在該螺柱上有合理的樂高磚堆疊於上方，反之則否。若每一個螺柱都檢查完且全數滿足沒有樂高磚與螺柱產生相交或是有樂高磚與螺柱相交並且是正確的堆疊情況之後，接著(3)針對每一個螺柱孔做如同(2)的檢查，差別是在每一個螺柱孔是尋找其他樂高磚的螺柱來做檢查。倘若以上三項檢查都沒有與其他樂高磚產生不合理的相交狀況，則判斷該樂高磚沒有產生物理碰撞，並可以正常的生成在系統中。

---

**Algorithm 1 Stud Validation**

---

**Input :**  $vp1, vp2$

- 1:  $equalPos \leftarrow |vp1.pos - vp2.pos| < posEpsilon$
  - 2:  $equalDir \leftarrow |vp1.dir - vp2.dir| < dirEpsilon$
  - 3: return  $equalPos$  AND  $equalDir$
- 

演算法 1 螺柱連結合理性驗證之演算法。



---

**Algorithm 2** Collision detection of Lego bricks

---

**Input :** A LEGO object L      **Output :** true - no collision detected;  
false - collision detected

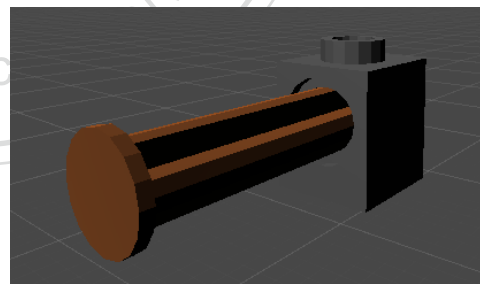
```
1: objs ← OverlapBox(L.b)
2: if(|objs| !=0)                                ▷ overlap with another bricks
3:   return false
4: for each vp ∈ L.s do
5:   if ( !CheckValidationOfConnectivity(vp) )    ▷ check if any of notch
   connect with this stud properly by Algorithm 1
6:     return false
7: end for
8: for each vp ∈ L.n do
9:   if ( !CheckValidationOfConnectivity(vp) )    ▷ check if any of stud
   connect with this notch properly by Algorithm 1
10:    return false
11: end for
12: return true
```

---

演算法 2 樂高的物理碰撞偵測之演算法。

演算法 2 同時也是一個理想上的假設，在實務上，兩個樂高磚的邊界框在相交時不全然是不合理的碰撞產生，如右圖所示，圖中為一個十字軸磚與帶插銷孔的基本磚的連結圖，可以看出棕色十字軸磚的釘有一部份是在灰色基本磚的邊界框中，若單用前面提到的演算法在第三行處就會結束並偵測到不合理的碰撞。

故在實作上，該處還會再做科技磚部分的 *technic validation* 演算法，其結構類似於演算法 2 中第 4 行到第 11 行的內容，只是將演算法中的螺柱(stud)與螺柱孔(notch)置換成科技磚的螺柱(插銷/十字軸)與螺柱孔(插銷孔/十字軸孔)的屬性去做迴圈的檢測。



### 5.3 輸入影像之處理

#### 5.3.1 影像感興趣區域擷取

首先，輸入影像要先找到當中需要由系統去計算並生成的內容，也就是影像

中的感興趣區域(Region of Interest, ROI)。系統的輸入影像大致有兩種主要類型，第一種是包含 Alpha 與色彩通道的影像(RGBA 影像)，使用者在透過系統互動製作輸入影像時可以將非感興趣區域的像素設為透明，使得系統在分析輸入影像時能直接專注在感興趣區域的內容；第二種是單純色彩的影像(RGB 影像或灰階影像)，此種影像大多數為使用者選擇影像後輸入，且通常包含使用者想要系統生成的前景像素之物件以及非感興趣區域的後景像素。而針對第二種的色彩影像，我們使用 OpenCV 套件的 GrabCut 函數來偵測 RGB 影像中感興趣區域的部分，接著將偵測到的部分以外之區域全部設為透明後來做為系統所需的輸入影像。

### 5.3.2 影像分析

獲得影像的感興趣區域後，使用者接著需要操作系統來決定使用的立體書機制，其中不同的機制有不同的參數需要使用者預先設置，例如在 V 型結構中，使用者需要先輸入預期的結構寬度  $W$  與厚度  $D$ ；在帳篷式與階梯式機制中，使用者需要先輸入兩側結構坐落在書頁上的位置  $(d_L, d_R)$ 、結構的寬度  $W$  與厚度  $D$ 。最後系統將使用者給定的結構寬度  $W$  與樂高的比例為基礎，把輸入影像像素化(Pixelization)轉換為像素模型以供系統準備進行初始的設計生成步驟。

### 5.4 初始的設計生成

在以上的步驟完成後，系統會根據輸入、使用者選擇的機制與參數，利用  $1 \times 1$  的薄磚來產生立體結構的初始樂高磚設計，並利用  $1 \times 1$  的薄磚與  $1 \times 1$  的基本磚來產生符合使用者需求之書頁的樂高磚設計。

### 5.5 樂高磚結構調整與優化目標函式

以上步驟所產生的初始設計是由大量的  $1 \times 1$  樂高磚所堆疊而成。首先，這種

設計在組裝上是無法實現的，即使樂高磚上下有螺柱與螺柱孔做為連接的方式，但在同高度下的樂高磚前後左右間沒有任何連接的方式。再來，即使如[Zhou et al. 2019]的研究中，將缺乏連接的同高度之樂高磚透過在結構底下加入一層支撐的樂高磚層(supporting layer)來確保組裝出的模型的穩固性，但透過大量的相同樂高磚進行組裝也是非常缺乏效率的，故在此階段系統將會透過我們設計的優化方式引導下進行結構上的調整，最終讓結構趨於能穩定與合理地組裝。

### 5.5.1 優化的目標函式 (Objective Function)

產生的樂高磚結構為了讓使用者可以合理地組裝以及整體的結構要足夠穩定，我們定義了一個目標函式去評估產生的樂高磚結構  $M$  之優劣性：

$$E(M) = w_{Simp}E_{Simp}(M) + w_{Stab}E_{Stab}(M)$$

其中， $E_{Simp}$  為結構的複雜度(Complexity)因子； $E_{Stab}$  為結構的穩定性(Stability)因子； $E_{EoIGs}$  為結構的支撐磚曝光程度(Exposed Ratio of BRACEs)。而  $w_{Simp}$ 、 $w_{Stab}$  和  $w_{EoIGs}$  分別為各項的權重。

### 5.5.2 結構複雜度(Model Complexity)

為了使結構更加地易於組裝，我們希望相同的輸入下，產生的整體結構能以更少的樂高磚總數組成。我們將產出的樂高磚模型之結構複雜度定義為：

$$E_{simp}(M) = \frac{|M| - 1}{SizeUnit(M) - 1}$$

其中  $SizeUnit(M)$  表示樂高磚整體的模型切割成最小樂高磚 ( $1 \times 1$  薄磚) 大小的單位量。以圖 24 為例，我們令圖中從左到右的結構分別為  $M_1$ 、 $M_2$ 、 $M_3$ ，三個模型由不同的樂高磚組成，三個模型所堆疊出的結構皆為  $1 \times 4$  大小的基本磚，切割成  $1 \times 1$  薄磚大小後，可得其  $SizeUnit(M_1) = SizeUnit(M_2) = SizeUnit(M_3) = \frac{4}{1} \times \frac{3}{1} \times \frac{1}{1} = 12$ ，而三個模型的  $|M|$  分別為 1、3、12，可從人眼判斷圖左結構的複雜性是最低的，量化其結構複雜度  $E_{simp}(M_1) = 0$ ；中間圖的

結構量化其複雜度  $E_{simp}(M_2) = \frac{2}{11}$ ；而圖右結構用最多樂高磚，複雜度最高，量化其複雜度  $E_{simp}(M_3) = 1$ 。利用此式可以引導優化過程傾向使用更少量的樂高磚來組成結構。



圖 24 模型複雜度之示意圖。圖左為一個  $1 \times 4$  基本磚組成的結構；圖中為一個  $1 \times 2$  與兩個  $1 \times 1$  組成的結構；圖右為十二個  $1 \times 1$  薄磚組成的結構。

### 5.5.3 結構穩定性(Model Stability)

為了使結構更加的穩定，我們希望整體的樂高磚結構不會在常見的情況下(例如：重力影響、簡單的立體書開闔)分解或產生不合理的移動與旋轉，故我們將樂高磚模型  $M$  的結構穩定度定義為：

$$E_{stab}(M) = \frac{w_l}{SizeUnit(M)} \sum_{B \in M} SizeUnit(B) \times [w_v E_v(B) + w_h E_h(B)] + w_g E_g(M)$$

我們根據結構中的樂高磚計算其垂直連結度(Vertical Connectivity)  $E_v$  與水平(Horizontal)連結度  $E_h$ ，並用其磚的  $SizeUnit$  做加權平均，作為每個樂高磚的區域性(Local)穩定度。接著計算整體結構的全域性(global)穩定度  $E_g$ ，將兩者利用  $w_l$  與  $w_g$  權重量化模型的結構穩定度。

#### 區域性垂直連結度

我們希望結構中的每個樂高磚其螺柱與螺柱孔都能連結其他的樂高磚，若一樂高磚沒有任何與其他磚的連結性則必然會受重力影響落下。接著我們也希望樂高磚傾向與更多的樂高磚做連結，使得結構能在交錯間堆疊，實務上樂高磚的交錯堆疊大幅的增加結構整體的穩定性，我們令樂高磚的垂直連結度為：

$$E_v(B) = \frac{1}{2} \left( 1 - \frac{|ConnectedBricks| + |Studs| - |EmptyS| - 1}{2|Studs| - 1} \right) + \frac{1}{2} \left( 1 - \frac{|ConnectedBricks| + |Notches| - |EmptyN| - 1}{2|Notches| - 1} \right)$$

垂直連結度分為計算螺柱連結性與計算螺柱孔連結性的部分，其中  $|ConnectedBricks|$  代表連結的樂高磚之數量； $|Studs|$  代表該樂高磚螺柱的數量； $|EmptyS|$  代表該樂高磚沒有與其他樂高磚連結的螺柱數量。而螺柱孔也是做相同的計算。以圖 25(f)為例，當系統在計算結構下方的  $1 \times 4$  基本磚之垂直連結度時，可得其螺柱的  $|ConnectedBricks| = 2$ ， $|Studs| = 4$ ， $|EmptyS| = 1$ ， $\left( 1 - \frac{|ConnectedBricks| + |Studs| - |EmptyS| - 1}{2|Studs| - 1} \right) = \frac{3}{7}$ ，且該基本磚的螺柱孔沒有與任何樂高磚連結，故其計算螺柱孔的垂直連結度為 1，得其區域性的垂直連結度為  $\frac{5}{7}$ 。利用此式可以讓系統傾向用樂高磚去做垂直上交錯的連結，以增加整體結構的穩定性。

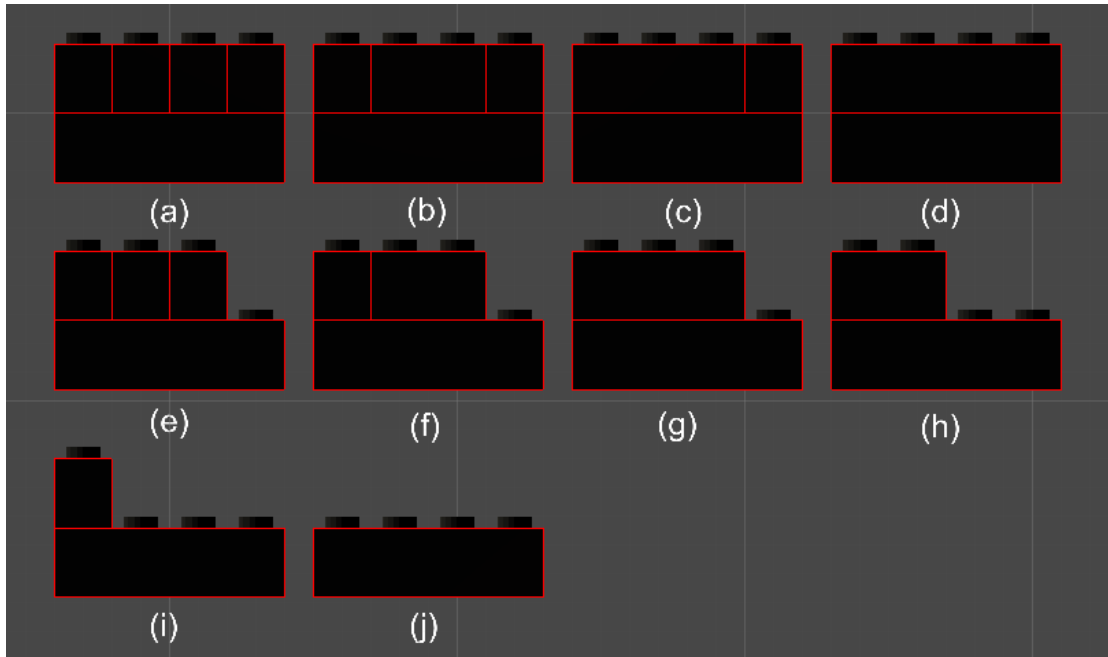


圖 25 區域性垂直連結度之示意圖。以每個結構下方的  $1 \times 4$  基本磚計算其螺柱的連結度部分： $(a) = 0$ ； $(b) = \frac{1}{7}$ ； $(c) = \frac{2}{7}$ ； $(d) = \frac{3}{7}$ ； $(e) = \frac{2}{7}$ ； $(f) = \frac{3}{7}$ ； $(g) = \frac{4}{7}$ ； $(h) = \frac{5}{7}$ ； $(i) = \frac{6}{7}$ ； $(j) = 1$ 。十個結構的  $1 \times 4$  基本磚計算其螺柱孔的連結度皆為 1。

### 區域性水平連結度

前面我們探討了樂高磚間垂直的連結度，該因子限制了每個樂高磚上下間的關係，但若樂高磚在前後左右側沒有其他相鄰的樂高磚，該樂高磚可能會屬於一個較易移動的狀態(例如旋轉)。故我們希望每個樂高磚的相鄰空間也該要有其他的樂高磚並排，以固定住樂高磚並且避免其沿著 y 軸的自體旋轉。我們令樂高磚的區域性水平連結度為：

$$E_h(B) = \frac{|EmptyCavities|}{|Cavities|}$$

以圖 26 為例，該圖中間為一個  $2 \times 8$  薄磚，其水平面具有  $2 \times (2 + 8) = 20$  個可並排樂高磚的單位空間(cavities)，同時並計算未填滿的單位空間比例做為水平的連結度。

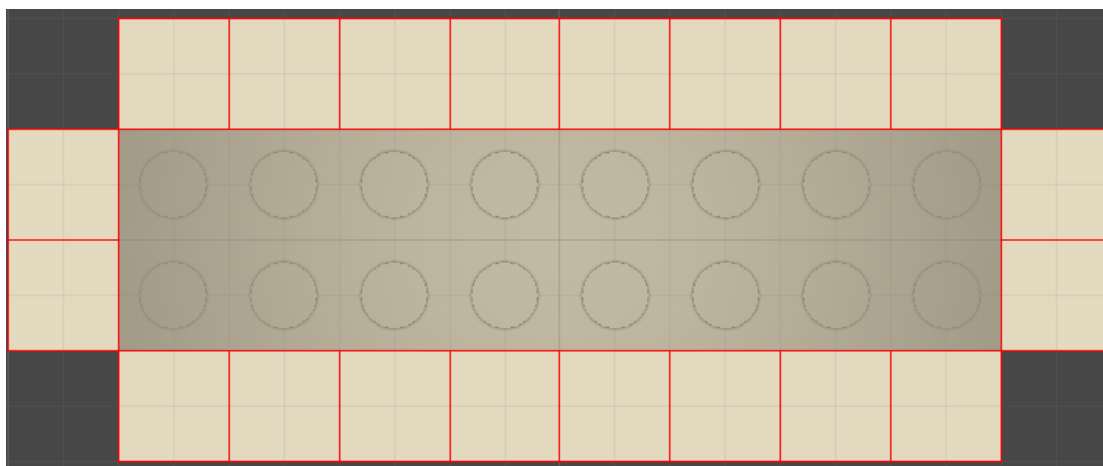


圖 26 區域性水平連結度之示意圖。以圖中灰色的  $2 \times 8$  薄磚為例，該樂高磚的前後左右側有 20 個相鄰的單位空間(如紅框)。

### 全域性連結度

前面所探討的連結度因子皆為區域性的，即使生成了每個樂高磚的區域性連結度都很優良的結構，也可能會產生全域性的不穩定結構。如圖 27 所示，其中 (a) 結構與 (b) 結構的區域性穩定度在量化上相同，但可很明顯看出 (a) 結構最右排的紅色磚即使在左側的交錯結構固定下，依然會受重力影響而整排掉落；而 (b) 結構因為排列的不同使得整個結構不會產生非預期的位移。故為了使系統傾向生成如圖 27(b) 較穩定的結構，我們使用簡單的無向圖(Undirected Graph)的概念去做全域性連結度的分析。如圖 28 所示，我們將樂高磚的中心做為圖的頂點(Vertex)，並將樂高磚彼此的連結性轉化為圖中的邊(edge)，可更清楚的看出 (a) 的結構分成了兩個沒有交集的圖，而 (b) 的結構則是全部樂高磚巧妙的連結成一個圖。我們將全域性的連結度量化為：

$$E_g(M) = 1 - \frac{|G_{max}|}{|M|}$$

其中  $|M|$  為結構的頂點數，亦等於結構的樂高磚數量；而  $|G_{max}|$  為結構中頂點數最大的圖之頂點數。用此公式可簡單的量化全域性的連結度，使系統在優化過程中傾向讓所有的樂高磚彼此間產生連結，且有效減少不穩定的獨立結構產生。

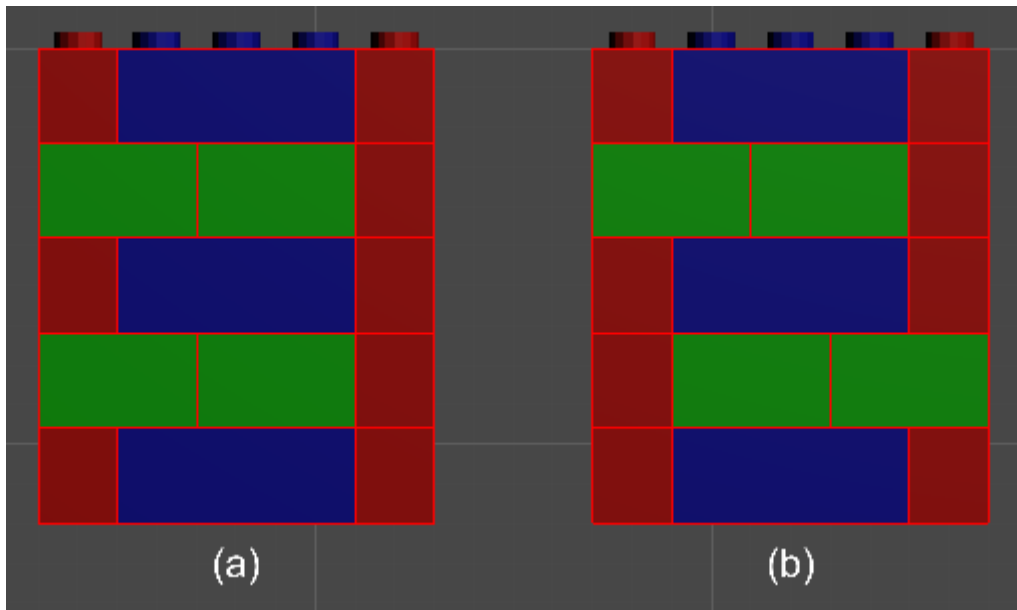


圖 27 區域性穩定度相同但結構穩定度具有差異的示意圖。其中紅色磚代表 $1 \times 1$  基本磚；綠色代表  $1 \times 2$  基本磚；藍色代表  $1 \times 3$  基本磚。

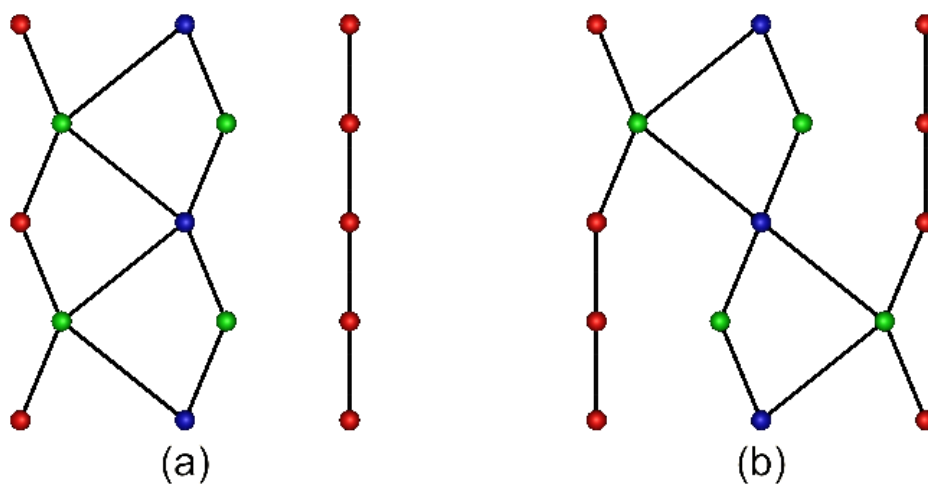


圖 28 將圖 27 之樂高磚結構轉化的無向圖。其中紅色節點代表 $1 \times 1$  基本磚；綠色節點代表  $1 \times 2$  基本磚；藍色節點代表  $1 \times 3$  基本磚。

#### 5.5.4 結構的支撐磚曝光程度(Exposed Ratio of BRACEs)

由於我們的研究是為了產出可開闔的樂高立體書設計，故結構中必然會有用



於呈現符合輸入影像的樂高磚結構以及用於支撐結構開闔的樂高磚(我們將這些樂高磚標記為 BRACE)。而這些用於支撐的樂高磚因為本身沒有任何呈現的功能，故我們希望系統在產出樂高磚設計時傾向讓這些支撐磚在使用者看的角度之下盡量是不可視(Invisible)的。所以我們量化結構的支撐磚曝光程度為：

$$E_{ERoBs}(M) = \frac{|Visible\ BRACEs|}{|BRACEs|}$$

利用此式可以檢查使用者設置的參數是否合適於輸入的影像，如圖 29，使用者不同的參數設置會影響 $E_{ERoB}$ 之值，左圖結構有47%的支撐磚在使用者的視點下是可視的(如圖灰色與棕色磚部分)；而右圖的結構只有1%的支撐磚為可視的(圖中棕色磚部分)，我們認為右圖所使用的參數生成之結構在支撐磚被遮蔽的狀況下明顯的優於左圖的結構。

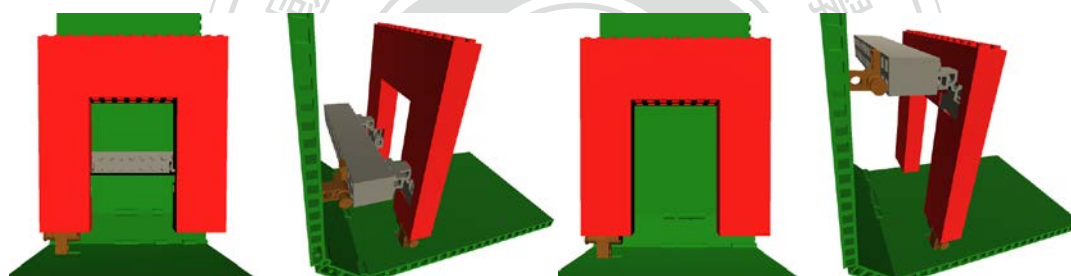


圖 29 相同輸入的不同設計產出結果。左邊二圖為同一結構，分別為正面視角與側面上方的視角，右邊同理。其中  $E_{ERoBs}(M_{left}) \cong 0.47$ ,  $E_{ERoBs}(M_{right}) \cong 0.01$ 。

## 5.6 系統的樂高磚結構優化方法與實作細節

我們透過模擬退火(Simulated Annealing)的模型 [Kirkpatrick et al. 1983] 來解決 5.5 節提到的複合因素之優化問題。模擬退火是一個隨機性的全域優化演算法，該方法模擬了晶體冷卻的過程，透過先將晶體加熱至一定的溫度(在演算法中稱為初始溫度)，晶體中的原子在降溫的過程中能自由的排列(對應到演算法的結構變化)以至更高密度，能量更低的晶體結構(對應於演算法中的全域性最佳解)。首先，我們將 5.4 產出的結構作為初始狀況並計算初始狀況的目標函式值(能量，

Energy)，接著我們迭代地選擇一個隨機的樂高磚，進行一次隨機的區域變化後，計算新的模型之能量來做 Metropolis 演算法，若新的能量低於原先的能量則使用接受該次的變化；若新的能量高於原先的能量，則根據當下情況的溫度以及能量的變化量來判斷，有一定的概率能接受該次變化。利用 Metropolis 演算法可有效地規避區域性的最佳解(即使能量上升系統也有可能接受該次變化)。此迭代的過程將會持續到設置的溫度降到停止溫度(完成退火)或是多次的能量未發生改變之狀況(視為已找到全域最佳解)。

---

**Algorithm 3** Simulated annealing for modification of pop-ups structure

---

**Input :** Initial Layout  $M$ , Initial Temperature  $T_{start}$ , Decay Ratio  $\alpha$ , constant  $k$

```

1:  $M_{old} \leftarrow M$ 
2:  $E_{old} \leftarrow ComputeEnergy(M_{old})$ 
3:  $T \leftarrow T_{start}$ 
4:  $Fail \leftarrow 0$ 
5: while  $T > T_{stop}$  AND  $Fail < FailCrit$  do
6:    $B \leftarrow SelectBrick(M_{old})$ 
7:    $M_{new} \leftarrow LocalModifier(B)$ 
8:    $E_{new} \leftarrow ComputeEnergy(M_{new})$ 
9:    $\Delta E \leftarrow E_{new} - E_{old}$ 
10:  if  $\Delta E < 0$ 
11:     $M_{old} \leftarrow M_{new}$ 
12:     $Fail = 0$ 
13:  else
14:    if  $exp(-\Delta E/kT) > Random(0, 1)$ 
15:       $M_{old} \leftarrow M_{new}$ 
16:       $Fail = 0$ 
17:    else
18:       $Fail += 1$ 
19:   $T \leftarrow \alpha T$ 
20: end while

```

---

演算法 3 基於模擬退火模型的樂高磚結構優化之演算法。

演算法 3 為基於模擬退火模型的樂高磚結構優化之演算法，我們條列並解釋幾項用於此演算法的函數：

·**ComputeEnergy(M)**：根據給定的模型M計算我們在 5.5.1 節提到的目標函式

之數值。

·**SelectBrick(M)**：從目前的模型M中取一隨機樂高磚。此方法將會有一定的機率優先選擇沒有與 5.5.3 節全域性連結度提到的  $G_{max}$  相連結之弱節點樂高磚，來增加整體結構相互連結的概率。

·**LocalModifier(B)**：將選定的樂高磚選分割(Split)、合併(Merge)其一做變化。分割是將該樂高磚根據其高度變化成  $1 \times 1$  的基本磚或薄磚排列而成之樂高磚集合；合併則是將該樂高磚與隨機一鄰近的磚做合併形成一個樂高磚，若合併過程中無法找到鄰近的樂高磚與樂高磚B做合併，則將此次變化直接視為失敗(如演算法 3 中的第十八行)。

根據經驗法則，我們將 5.5.3 節提到的  $w_l$ 、 $w_v$ 、 $w_h$ 、 $w_g$  分別設為 0.6, 0.6, 0.4, 0.4。接著我們將模擬退火模型的初始溫度  $T_{start}$  設為 1，停止溫度  $T_{stop}$  設為 0.01，降溫比率  $\alpha$  設為 0.9995，常數  $k$  設為 0.005。

## 5.7 使用者互動介面

從圖 22 系統自動生成樂高立體書結構的流程中，使用者也可以透過與系統的互動來產生出不一樣的結果。

首先，在輸入影像之處理(5.3 節)中，我們的系統提供了基本的繪畫功能，供使用者直接在系統中進行繪畫，同時使用者也可以給定自己想要的機制以及適合的參數，來預覽 5.3.2 節生成出的像素模型，使用者可以透過預覽的像素模型大致推估生成出的結構大致輪廓，藉此確認自己所使用的參數是否適合目前的輸入影像。

在初始的設計生成(5.4 節)中，我們的系統提供使用者在調整機制與參數的同時，進行生成結構的預覽，除了使用者利用眼睛確認，此處也可以利用 5.5.4 節提到的結構因子來去評估目前使用的機制與參數適合與否。

系統生成初始設計後，在樂高磚結構優化(5.6 節)中，我們的系統提供使用者

可以自由地手動調整樂高磚的結構(如新增、刪減、移動樂高磚...等)，也可以選擇樂高磚來進行合併、分割與置換(Replace)，其中合併與分割就如 5.6 節提到的功能相同，而置換的功能則是選定的樂高磚換成一樣可以擺在相同位置且不會影響結構穩定的不同類型之樂高磚。而以上提到的調整結構之功能，不論是在結構優化前做預先的調整，在優化結束後也可以用來做後處理(Post-processing)的功能，使得生成出的結構能更符合使用者的需求。



## 六、研究結果

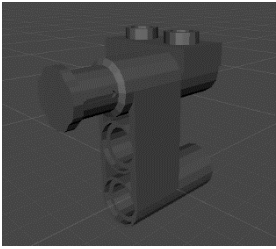
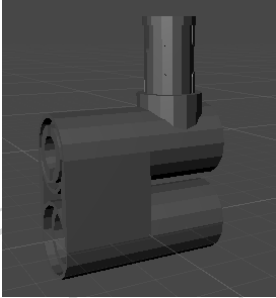
在這個章節，我們將根據提出的樂高立體書機制作研究結果的呈現，並在後續小節進行效率與優化權重的分析，以證明我們提出的系統有助於建構多樣化的樂高立體書結構。

### 6.1 階梯式結構

利用 4.2 與 4.3 節提到的公式做引導，我們實作兩種階梯式結構的呈現方式：單面呈現的結構與雙面呈現的結構。如 4.3 節我們提到，常見的階梯式結構多用單面來做呈現，並將另一面用於支撐，故我們實作了單面的結構。並且我們也實作了雙面呈現的結構以呈現不同風格與類型的輸入影像。

在單面呈現的結構中，立體結構的樂高磚堆疊方向與書頁的方向垂直，使用者在輸入連結磚結構與書脊的距離 $d_L$ 、 $d_R$ 後，系統會自動計算適合的 $H_L$ 、 $H_R$ ，以讓結構在展開時，立體的結構夾角能逼近 $90^\circ$ ；而不同於在單面結構，雙面呈現的結構中，立體結構的樂高磚堆疊方向與書頁相同，故在使用者輸入 $d_L$ 、 $d_R$ 後，系統將會自動計算適合的 $W_L$ 、 $W_R$ 來取代 $H_L$ 、 $H_R$ 。而由於堆疊方向的相同，使得 4.2.4 節提到的，單面結構的 $d_L$ 、 $d_R$ 相差必須為偶數個樂高磚的寬度，在雙面結構中沒有這個限制也能讓立體書正常地開闔(表格 2 最後一列)。

表格 1 與表格 2 為兩種實作結構的參數表，使用者可以自由定義表格中結構圖以左的設定，來產生結構圖以右的對應參數。結構厚度 $D$ 的最低值受限於 $H'$ 之值，若 $H' = 40$ (兩個樂高磚寬)，則結構厚度 $D$ 至少也要是相同之值才能避免闔上時兩側連結磚結構會直接碰撞。而在設計時 $D$ 也盡量不可超過 $d_L$ 、 $d_R$ ，假設 $D > d_R$ ，則右側的結構很有可能直接與左書頁產生碰撞。我們的系統在單面的結構中提供了七種連結磚的結構，在雙面的結構中提供了四種連結磚的結構，使用者也可以自行定義合適的連結磚結構來生成不一樣的階梯式立體書。

D	$d_L$	$d_R$	$H'$	連結磚結構示意圖	$H_L$	$H_R$	結構夾角
40	200	200	40		220	220	$88.2^\circ$
60	200	200	40		228	228	$88.3^\circ$
60	120	160	40		188	148	$87.6^\circ$
20	120	120	20		132	132	$86.8^\circ$
40	160	120	20		140	180	$87.2^\circ$
100	280	80	20		80	280	$87.5^\circ$

表格 1 單面呈現的階梯式結構的參數值列表(單位：LDU)。結構圖以左為使用者定義之參數，以右為系統自動計算適合的對應參數。

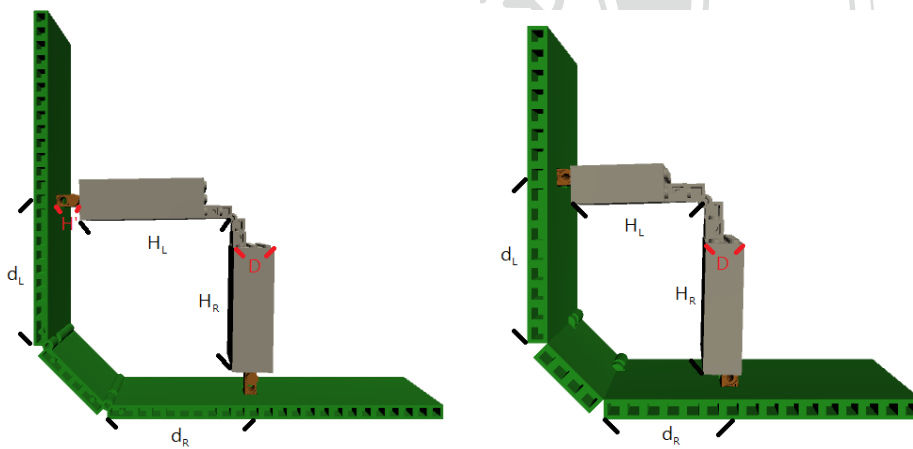
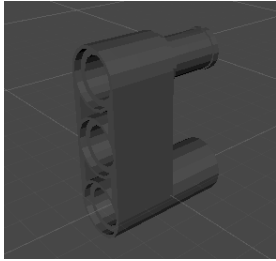
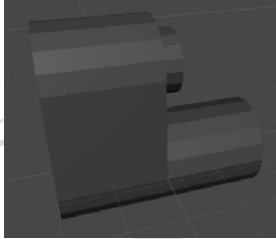


圖 30 左圖為表格 1 第二列參數生成的結構圖，右圖為第五列的結構圖。

D	$d_L$	$d_R$	$H'$	連結結構示意圖	$W_L$	$W_R$	結構夾角
40	200	200	40		240	240	$88.2^\circ$
60	200	200	40		240	240	$92.8^\circ$
60	120	160	40		200	160	$94.1^\circ$
20	120	120	20		160	160	$86.8^\circ$
40	160	80	20		120	200	$86.5^\circ$
60	180	160	20		200	220	$93.8^\circ$

表格 2 雙面呈現的階梯式結構的參數值列表(單位：LDU)。結構圖以左為使用者定義之參數，以右為系統自動計算適合的對應參數。

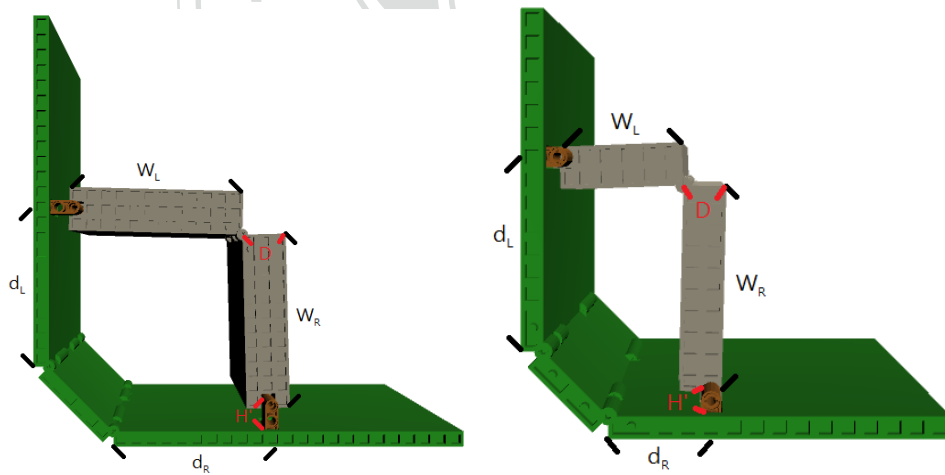


圖 31 左圖為表格 2 第二列參數生成的結構圖，右圖為第五列的結構圖。

### 單面呈現之結構

單面結構可用於呈現如立體卡片中常見的立體文字設計(如圖 32)，也可用於呈現建築物的正面結構(如圖 22 的例子)。雖然階梯式機制在紙立體書的相關研究中常用於呈現建築物的正面(front)與上面(up)，但根據 3.3 節提到的限制，以及 4.2、4.3 節提到的樂高立體書階梯式機制之公式可得知，背面的結構高度會受限於正面的結構以及連結的科技磚在書頁上的位置...等因素。故即便我們的系統也可以將背面的結構(圖 32、圖 33 灰色磚部分)用於呈現建築物的上面，但呈現的整體建築結構在厚度方面有很大的機率是被過度壓縮的狀態。

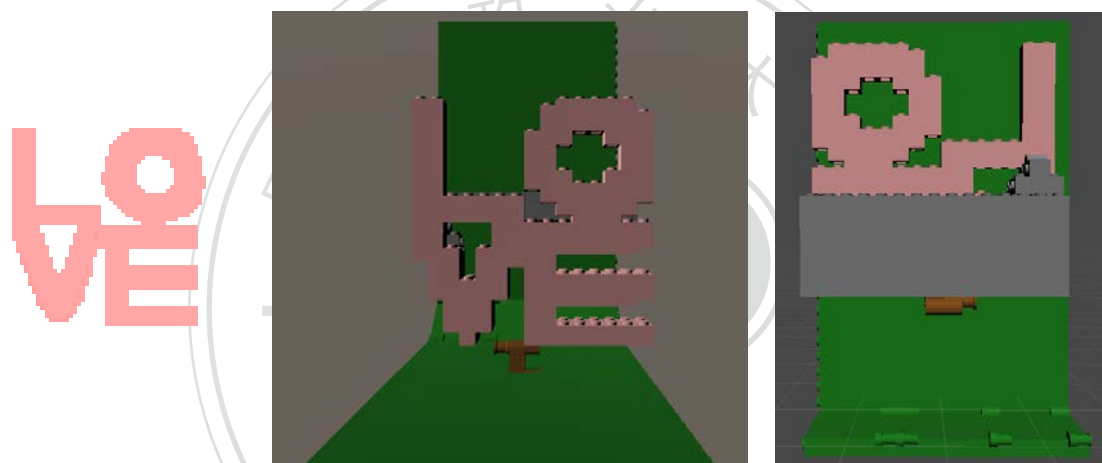


圖 32 呈現立體文字的樂高書結構。左圖為輸入影像，使用者設置結構的寬度為 15，中間圖為系統產出之設計，右圖為將立體書闔上後的結構圖(切除背面的書頁)。

我們的系統也供使用者針對同一影像使用複數個相同的立體書機制去生成結構。如圖 33 所示，使用者將輸入影像的白宮分成中間與左右兩側共三個部分，分別使用相同機制以及不同參數去做結果的生成，使得結構間增加了前後的關係，在使用者正面視角的視覺效果上增加了整體結構的厚度，簡單的複合設計也可讓結構看起來更加地立體。





圖 33 利用三個相同機制產生的樂高書結構。輸入影像與圖 22 相同，使用者分別將三個結構的結構寬度分別設置為 10,9,10(從左到右)，左圖為結構的正面視角，中間圖為結構的斜上方視角，右圖為將立體書闔上後的結構圖(切除背面的書頁)。

### 雙面呈現之結構

雙面的階梯式結構可以用來呈現任意的輸入影像，系統會以影像的垂直線做為分界，界線以左(右)的內容生成左(右)側的立體結構。例如使用者將圖 34 的骷髏作為輸入影像，並輸入相同的 $d_L$ 、 $d_R$ 後，系統將會產出左右對稱的骷髏結構，若 $d_L$ 、 $d_R$ 分別輸入不同的值，系統將會計算適合的分界線，以生成出左右不對稱的結構。

圖 35 為使用者利用四個相同機制去生成的結構，因為系統在這個結構都是以垂直線作為分界，故使用者若要產出如該圖的結果，可先將輸入影像逆時針旋轉 90 度後再進行計算與生成。

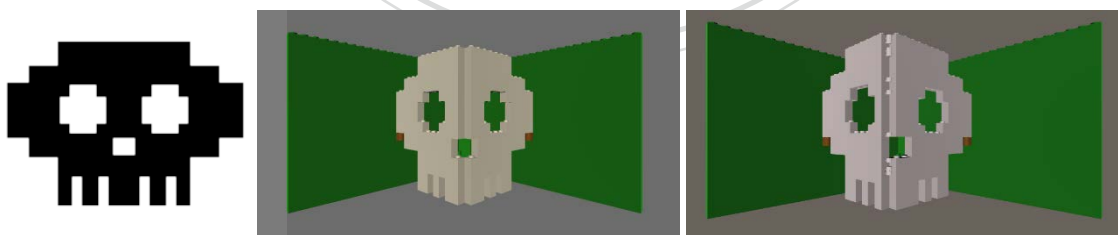


圖 34 使用雙面呈現的階梯式機制生成之骷髏結構。左圖為輸入影像，中間圖為 $d_L = d_R$ 時系統生成的對稱設計，右圖則為 $d_L > d_R$ 時生成的左右非對稱之設計。我們將輸入影像的白色部分轉為黑色以方便觀察。

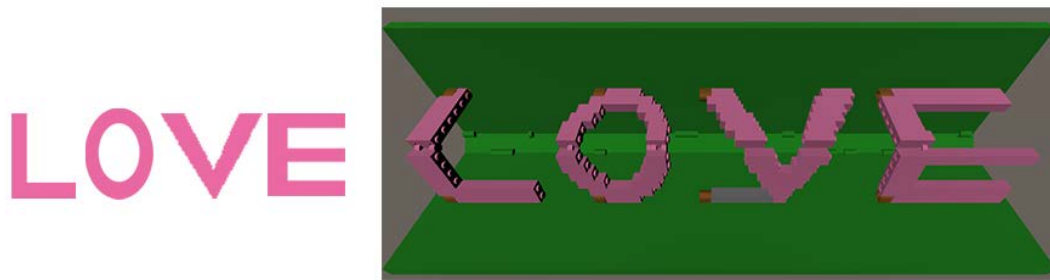


圖 35 使用四個雙面呈現的階梯式機制生成的立體文字結構。左圖為輸入影像，右圖為系統生成的設計。

## 6.2 V 型結構

利用 4.1 節提到的公式與限制做為引導，我們實作了 V 型的結構。使用者透過自訂結構寬度  $W$  與與結構的夾角  $\theta$ ，接著系統會計算合適的连接磚與書脊之距離  $d$ ，接著微調  $\theta$  至適合的角度  $\theta'$ ，接著利用這些參數生成初始的設計。表格 3 為 V 型結構的參數值表，左三欄的參數為使用者定義，而後三欄則是透過系統計算出來的參數。我們的系統會根據使用者定義與計算得出的參數來去計算適合插在帶插銷孔的薄磚中的十字軸磚。由於目前系統僅提供一種連結磚結構，故 V 型結構的  $D$  最小也必須為兩個樂高磚寬，且因為十字軸磚的種類只有 13 種 (2~12、16 與 32，單位為樂高磚寬)，為了方便計算，我們在系統只使用前十一種，而主要的影響因素為厚度  $D$  與角度  $\theta$ ，故我們系統目前在 V 型結構的厚度最大容許值約為 10 個樂高磚寬 (200LDU) 並搭配較大的角度  $\theta$ ，在較大的結構夾角  $\theta$  中，結構隨著立體書闔上時移動的距離較小，系統就可以選擇使用較小的十字軸磚 (表格 3 第 3~5 列)。

D	W	$\theta$	$\theta'$	d	使用的十字軸磚(樂高磚寬)
40	240	$15^\circ$	$15.6^\circ$	260	6
40	240	$30^\circ$	$27.3^\circ$	240	6
40	240	$45^\circ$	$42.2^\circ$	200	6
60	240	$15^\circ$	$15.6^\circ$	260	7
60	240	$45^\circ$	$42.2^\circ$	200	6
200	220	$45^\circ$	$43.9^\circ$	180	12

表格 3 V 型結構的參數值列表(單位:LDU)。左三欄為使用者定義之參數；後三欄為系統自動計算適合的對應參數(單位:D、W、d 為 LDU； $\theta$ 、 $\theta'$ 為度)。

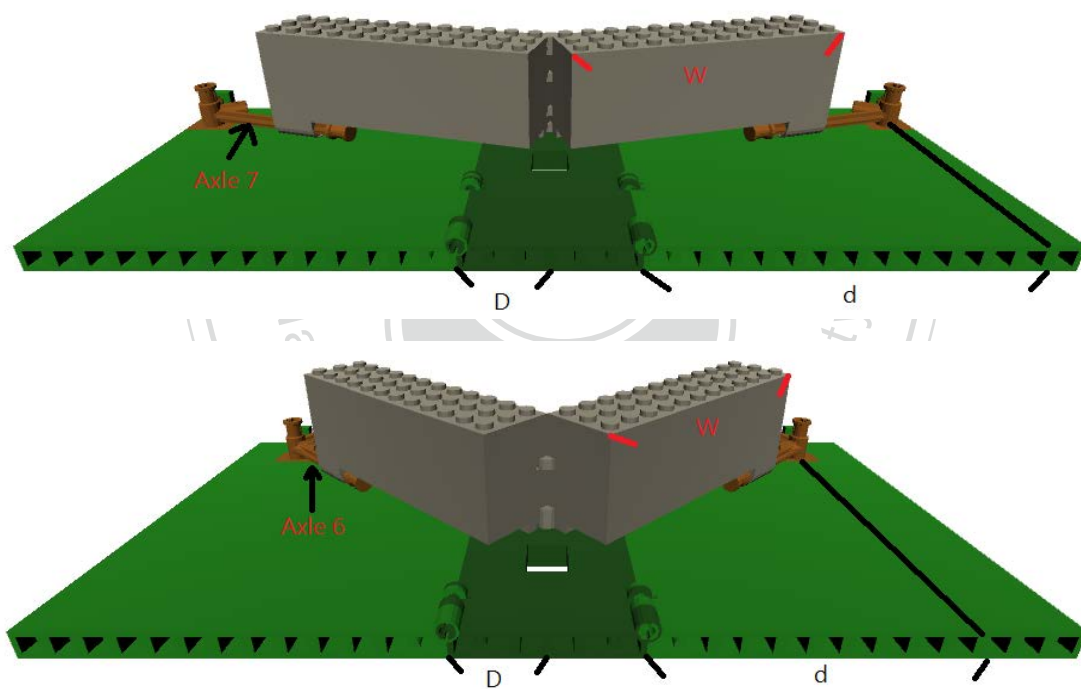


圖 36 上圖為表格 3 第四列參數生成的結構圖，右圖為第五列的結構圖。可看出較大的結構角度所使用的十字軸磚長度較短。

V 型的結構可用於呈現建築物的正面(圖 38)，也可用於輸入呈現立體的風景。常見的立體風景通常包含遠近不同的景色，例如雲會在山前或山後、而天空則會在景色最遠處等距離的資訊。如圖 37 所示，我們的系統也允許使用者輸入數張影像作為輸入，並讓使用者決定每個輸入的影像所要對應的深度，以供系統產出有層次變化的結果。



圖 37 呈現山、雲與天空的風景之 V 型結構。左邊三張圖為輸入的影像，從下到上分別為近景、中景與遠景。右圖為輸出的結果，我們將輸入影像中白色的雲轉為黑色以方便觀察。

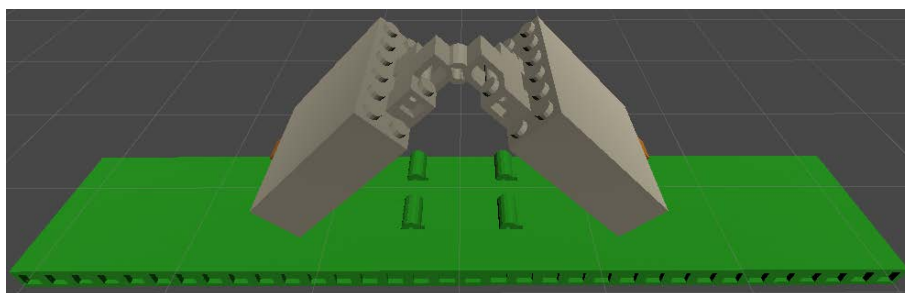


圖 38 呈現簡單房屋的 V 型結構。左圖為輸入影像(參考圖 1)，中間圖為生成之設計，右圖為將立體書闔上後的結構圖(切除左書頁)。

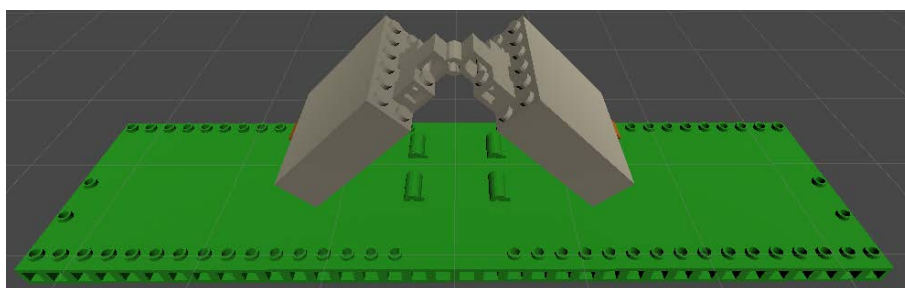
相對於階梯式的單向結構只能呈現建築物的單面或是被壓縮的雙面，V 型結構由於使用者可以較自由地選擇結構的角度 $\theta$ ，故除了用小角度呈現建築物的單面外，也可用 $45^\circ$ 去呈現建築物的正面與左面(left)或右面(right)等，在建築相關的輸入影像上比階梯式更具有彈性。

### 6.3 書框的生成

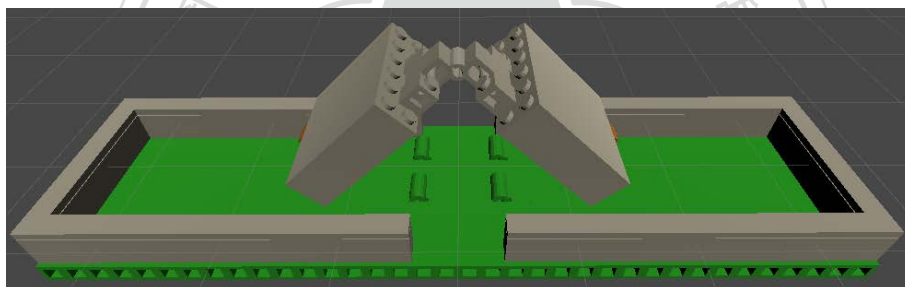
實務上的立體書需要在闔上時，利用書框將書裡的「內容」遮住(如圖 1 的白色磚部分)，而前面所生成的各種立體書結構中，書頁內的立體結構皆為暴露在外的狀態(如圖 19)，故我們的系統也提供在生成立體書後計算並生成適合的書框結構。圖 39 為生成書框的示意圖，我們在書頁上引入變形基本磚，讓書框可以合理的座落在書頁上。我們的系統在產生書框前需要在書頁的外圍生成薄磚、基本磚與變形基本磚組成的書框底座，如 3.3 節提到，書頁中每兩個變形基本磚中間所夾的  $1 \times 1$  薄磚數只能是  $2 + 5 \times n$ ,  $n \in \mathbb{N}$  個，而每三個薄磚可以用一個基本磚代替，而我們的書框底座就用此條件進行隨機的生成(圖 39 b)。接著我們的系統會在書框底座上利用各種薄磚生成書框的樂高磚結構(圖 39 c)，其中書框的最上一層需要使用平滑磚代替薄磚，使得左右書頁的書框不會產生螺柱的碰撞。我們將書框的結構往上移動，可清楚的看出書框底部的螺柱孔有與書框底座的螺柱對齊(圖 39 d)。將圖 c 的立體書闔上後的狀態就如圖 e 所示。



(a)



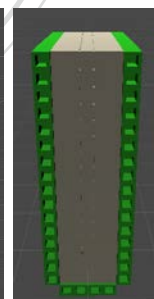
(b)



(c)



(d)



(e)

圖 39 書框生成的示意圖。(a)系統產出之樂高立體書結構；(b)系統於書頁外加入一層由變形基本磚、基本磚與薄磚組成的書框底座；(c)生成合適高度的書框磚；(d)將書框抬起後，從下方視角可看到書頁上每個變形基本磚的螺柱皆有對準書框樂高磚的螺柱孔；(e)將包含書框的樂高立體書結構闔上後之示意圖。

## 6.4 研究分析

我們在表格 4 整理了數個結果進行優化前後的結果，其中分別列出該結果圖使用的權重，右半部細分之上列為優化前樂高結構的目標函式各項之值 (Energy/Loss，越高則相對越差；反之則越佳)，下列為優化流程結束後樂高結構的目標函式各項之值。可明顯看出透過 5.5 節我們所提出的結構調整之目標函式與優化方法，各項因素之值皆有明顯的下降，可以理解為結構組裝的容易程度以及結構的穩定性都得到了顯著的提升。

由於我們在系統優化的過程中僅允許水平面向的合併與分割，故單純系統優化後的結構都是由薄磚組成的，並沒有垂直方向的薄磚再合併成基本磚的情況發生，故結構的 $E_{Simp}$ 在優化後皆落在 0.25 左右，若要優化至更佳的結構複雜度，通常會需要使用者在互動介面預先將部分的薄磚透過垂直合併為基本磚再進行系統優化。而在結構的穩定性方面，優化後其全域性連結度 $E_g$ 大多皆有效的達到 0(結構的磚彼此相連，沒有單獨的樂高磚)。但由於我們的結果圖多用較低的結構厚度去生成，故其區域性水平連結度 $E_h$ 因為水平向周圍的磚較少的關係，使得結構平均的區域性連結度( $w_v E_v + w_h E_h$ )皆在 0.5 至 0.6 間，最終結構整體的穩定性 $E_{Stab}$ 約落在 0.35 左右。若需要優化至更加的結構穩定性，我們傾向使用較大的結構厚度來提升結構的區域性水平連結度，或著是直接調整 $w_{Simp}$ 與 $w_{Stab}$ 的權重比，令優化過程更傾向穩定性優先的結構，在表格 4 可以看到我們使用兩種不同的權重來優化圖 38 的初始設計，而不同的權重經過優化後 $E_{Simp}$ 與 $E_{Stab}$ 也有明顯的改變。

	圖/時間	$W_{\text{simp}}$	$W_{\text{stab}}$	$E_{\text{Simp}}$	$E_{\text{Stab}}$	Energy
優化前	圖 22	2	3	0.982	0.505	3.479
後	357 秒			0.252	0.342	1.530
優化前	圖 32			0.986	0.500	3.472
後	422 秒			0.226	0.362	1.538
優化前	圖 35			0.963	0.506	3.444
後	372 秒			0.452	0.368	2.008
優化前	圖 38	4	1	0.971	0.478	3.380
後	567 秒			0.291	0.312	1.518
優化前	圖 38			0.979	0.476	4.396
後	519 秒			0.236	0.357	1.305

表格 4 優化權重的設計與優化前後的目標函式值之變化表。



## 七、結論

### 7.1 結論

我們提出一套系統，將二維的輸入影像透過選擇適合的機制來產生樂高立體書的結構。我們利用樂高的基本磚建構立體的模型，並利用基本磚、樞紐磚與科技磚的複合結構組裝出可在書頁的開闔當中正常運作、移動的連接結構。我們提出的系統在生成立體書的流程中支援多種類的樂高磚類型，使用者也可以另外引入自己所需要的樂高磚；生成的結果也說明我們的系統可以適用於多種類型的輸入影像。三維的樂高模型在組裝與設計的自由度是無窮的，我們提出了結構優化的目標與流程，來迭代地計算出容易組裝且更加穩定的樂高立體書模型。

我們的主要貢獻有：一，我們提出了三個樂高立體書的機制，透過公式分析合理的樂高立體書之結構；二，我們提出一套完整的系統，將二維的輸入影像轉化為合理的樂高立體書結構；三，我們提出一套優化目標函式與優化流程，使結構趨於更容易組裝且更穩定的狀態。

### 7.2 未來展望

我們的系統在計算樂高立體書時僅將輸入的影像根據輸入的機制與參數轉化為像素模型並用 $1 \times 1$ 的薄磚以填滿的方式生成，並未考慮輸入影像與輸出結構的相似度(Shape Matching)等因素。同時，目前系統在建構結構上皆以基本磚與薄磚為主，並沒有引入斜向磚、弧形磚等樂高磚，故若輸入影像的邊界區域變化較大時，生成出的設計將會遺失邊界變化的特徵。

另外，在樂高磚結構的優化過程中我們並沒有針對樂高磚進行垂直方向的合併與切割(薄磚 $\rightarrow$ 基本磚、基本磚 $\rightarrow$ 薄磚)，由於提出的結構優化流程是完全隨機的，將結構中的樂高磚進行垂直方向的合併容易使得結構雖然穩定，但卻參差不齊難以組裝與拆解(如圖 40)。針對此問題，我們的系統提供圖形使用者介面讓使用者自行做垂直方向的合併與切割，使得圖 40 左圖的結構能先從使用者互動轉

變成由上而下分別為一層的薄磚、一層的基本磚與一層的薄磚，最後將這三層樂高磚進行目前系統提供的結構優化方式以產出類似圖 40 中間圖的結構。

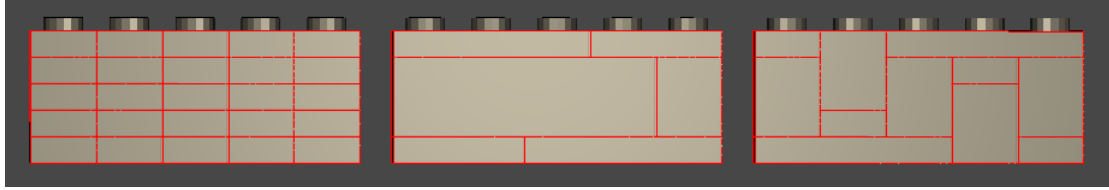


圖 40 使用垂直方向的合併可能產生之結構比較圖。左圖為初始設計，為 $5 \times 5$ 的薄磚空間，中間圖為目前系統優化中可能產生的結構圖，右圖則為使用垂直方向合併可能產生的結構圖。可看出即便中間圖與右圖的結構皆有足夠的穩定性，但右圖的結構相對更難組裝，使用者看起來也更加零碎。

我們的系統僅針對一個或複數個二維的輸入影像做樂高立體書的計算與生成，但現實中的樂高結構更多是展現複雜的三維立體模型，故我們希望能將二維的輸入影像推廣至三維的輸入模型，並且在系統中引入更多類型的斜向磚與弧形磚，同時在優化的目標函式中引入輸入模型與輸出結構間的相似度計算，以用更多不同種類的樂高磚生成，來保留輸入模型的特徵。同時我們也希望能提出更新穎與更複雜的樂高立體書機制來增加建構樂高結構的趣味性。

## 參考文獻

- 宋家慶、翁瑋辰、張登凱、紀明德。具關節可動之樂高生物骨架設計。Computer Graphics Workshop 2018。
- Coros, S., Thomaszewski, B., Noris, G., Sueda, S., Forberg, M., Sumner, R. W., ... & Bickel, B. (2013). Computational design of mechanical characters. *ACM Transactions on Graphics (TOG)*, 32(4), 83.
- Demaine, E. D., & Tachi, T. (2017). Origamizer: A practical algorithm for folding any polyhedron. In *LIPICs-Leibniz International Proceedings in Informatics (Vol. 77)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.
- Glassner, A. (1998). Interactive pop-up card design. Microsoft Technical Report.
- Hong, J. Y., Way, D. L., Shih, Z. C., Tai, W. K., & Chang, C. C. (2016). Inner engraving for the creation of a balanced LEGO sculpture. *The Visual Computer*, 32(5), 569-578.
- Iizuka, S., Endo, Y., Mitani, J., Kanamori, Y., & Fukui, Y. (2011). An interactive design system for pop-up cards with a physical simulation. *The Visual Computer*, 27(6-8), 605-612.  
<https://doi.org/10.1007/s00371-011-0564-0>
- Koo, B., Li, W., Yao, J., Agrawala, M., & Mitra, N. J. (2014). Creating works-like prototypes of mechanical objects. *ACM Transactions on Graphics*, 33(6).
- Kuo, M. H., Lin, Y. E., Chu, H. K., Lee, R. R., & Yang, Y. L. (2015). Pixel2brick: Constructing brick sculptures from pixel art. In *Computer Graphics Forum (Vol. 34, No. 7, pp. 339-348)*.
- Lang, R. J. (1996). A computational algorithm for origami design. In *Proceedings of the twelfth annual symposium on Computational geometry (pp. 98-105)*. ACM.
- Lee, Y. T., Tor, S. B., & Soo, E. L. (1996). Mathematical modelling and simulation of pop-up books. *Computers & Graphics*, 20(1), 21-31. [https://doi.org/10.1016/0097-8493\(95\)00089-5](https://doi.org/10.1016/0097-8493(95)00089-5)
- Li, X. Y., Shen, C. H., Huang, S. S., Ju, T., & Hu, S. M. (2010). Popup: automatic paper architectures from 3D models. *ACM Transactions on Graphics (TOG)*, 29(4), 111.

- Luo, S. J., Yue, Y., Huang, C. K., Chung, Y. H., Imai, S., Nishita, T., & Chen, B. Y. (2015). Legolization: optimizing lego designs. *ACM Transactions on Graphics (TOG)*, 34(6), 222.
- Ono, S., Alexis, A., & Chang, Y. (2013). Automatic generation of LEGO from the polygonal data. In *International workshop on advanced image technology* (pp. 262-267).
- Ruiz Jr, C. R., Le, S. N., Yu, J., & Low, K. L. (2014). Multi-style paper pop-up designs from 3D models. In *Computer Graphics Forum* (Vol. 33, No. 2, pp. 487-496).
- Testuz, R., Schwartzburg, Y., & Pauly, M. (2013). Automatic Generation of Constructable Brick Sculptures. In *Eurographics (Short Papers)* (pp. 81-84).
- Xiao, N., Zhu, Z., Martin, R. R., Xu, K., Lu, J. M., & Hu, S. M. (2018). Computational Design of Transforming Pop-up Books. *ACM Transactions on Graphics (TOG)*, 37(1), 8.
- Yun, G., Park, C., Yang, H., & Min, K. (2017). Legorization with multi-height bricks from silhouette-fitted voxelization. In *Proceedings of the Computer Graphics International Conference*(p. 40). ACM.
- Zhu, L., Xu, W., Snyder, J., Liu, Y., Wang, G., & Guo, B. (2012). Motion-guided mechanical toy modeling. *ACM Trans. Graph.*, 31(6), 127-1.
- Zhang, M., Igarashi, Y., Kanamori, Y., & Mitani, J. (2015). Designing mini block artwork from colored mesh. In *International Symposium on Smart Graphics* (pp. 3-15). Springer, Cham.
- Zhou, J., Chen, X., & Xu, Y. (2019). Automatic Generation of Vivid LEGO Architectural Sculptures. *Computer Graphics Forum*, 38(6), 31-42. doi:10.1111/cgf.13603