

Parquet Deformations, Computational Design e IA. Esercizi grafici del passato per nuovi paradigmi

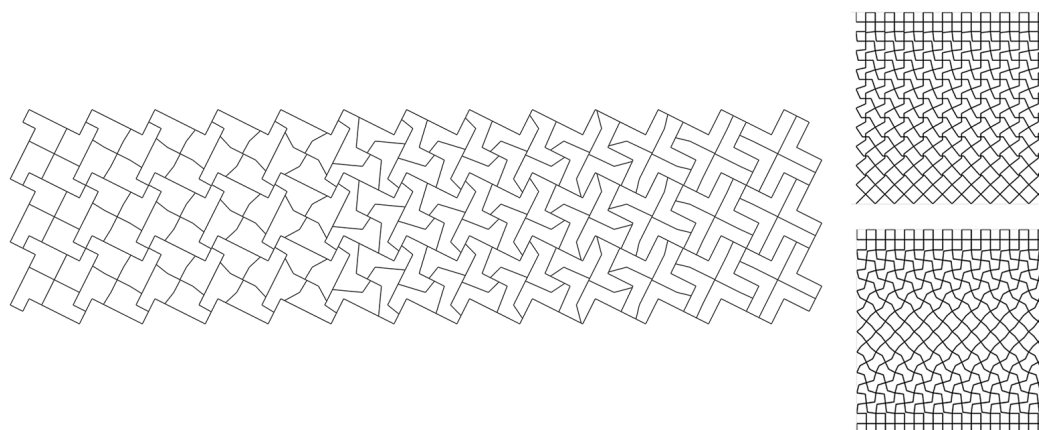
Giorgio Buratti
Giorgio Vignati

Abstract

La *Parquet Deformations* è un esercizio paradigmatico, introdotto da William Huff alla scuola di Ulm negli anni Sessanta, con lo scopo di migliorare l'attitudine degli studenti ad indagare la dimensione configurativa dello spazio, tramite la trasformazione sequenziale dei modelli. Questo scritto propone una lettura multilivello dell'esercizio per inquadrarlo in una prospettiva di sperimentazione computazionale effettuata tramite il confronto tra Intelligenza Artificiale e *Computational Design*. L'obiettivo è evidenziare la valenza pedagogica dell'esercitazione e come strumento del pensiero progettuale e come dispositivo utile alla contemporaneità del disegno, evidenziando nel contempo similitudini e differenze tra due approcci che, pur avendo una comune matrice informatica, si differenziano sensibilmente dal punto di vista formale, funzionale e applicativo.

Parole chiave

Parquet deformations, computational design, intelligenza artificiale, reti neurali, disegno algoritmico.



Confronto tra *Crossover*, esercitazione disegnata a mano da Richard Long nel 1963, qui ridisegnata algebricamente dall'autore (sopra) e quanto ottenuto tramite IA *text-to-image*.

Introduzione

La didattica del disegno per il progetto mira a sviluppare la capacità di formalizzare un problema mediante l'utilizzo di segni, col fine di analizzarne gli aspetti salienti ed identificare le proprietà risolutive necessarie. L'indagine della dimensione configurativa dello spazio, tesa ad evidenziare i sistemi relazionali, fenomenici e semantici, è stata sviluppata nei decenni dalle scuole di tutto il mondo attraverso specifici esercizi grafici. Negli anni alcune esercitazioni, e quindi alcuni punti focali disciplinari, sono decadute a favore di nuove pratiche, in un continuo adattamento alle circostanze e agli sviluppi del contesto storico. Il passaggio dalle tecnologie analogiche alla logica computazionale dell'elaboratore è stato in questo senso un passaggio cruciale. Le ricadute nell'insegnamento e nella pratica del disegno e della rappresentazione sono stati epocali, evolvendo l'ambito pedagogico e professionale.

Esercizi grafici efficaci tanto nel mondo digitale che in quello analogico, le *Parquet Deformations* di William Huff sono tassellazioni planari modulari e simmetriche ordinate secondo principi di transizione ottenuti evolvendo i tasselli lungo una o più dimensioni. Pur sviluppate alla Scuola di Ulm intorno al 1956 han mantenuto nel tempo il loro valore didattico, probabilmente grazie all'influenza di Maldonado [1], promotore dell'innesto sistematico nei corsi di disegno e composizione di discipline allora in fase emergente quali la cibernetica e la teoria dell'informazione. A prescindere dall'utilizzo di squadrette e matita o dell'elaboratore, questi esercizi esprimono infatti il potenziale del disegno nella valenza di 'programma' – dal latino *programma-matis*, 'scrivere prima' – capace di definire il percorso necessario a raggiungere un determinato obiettivo progettuale, a partire dalle condizioni al contorno sino all'organizzazione delle attività da intraprendere. La geometria è qui intesa come un codice che, sviluppandosi, genera ed induce l'invenzione della forma dello spazio, premessa operativa di un atto progettuale consapevole e apice dell'evoluzione bauhausiana del disegno che dalla libera sperimentazione di estrazione artistica, progredisce a una più precisa definizione degli elementi, delle regole e degli obiettivi.

Precedenti lavori [Buratti 2023, pp. 122-138] hanno approfondito la storia e le caratteristiche delle *Parquet Deformations*, illustrando i principi generativi degli algoritmi usati per la loro creazione, e documentando come tale approccio induca a sviluppare processi coerenti, che conducono l'allievo dall'essere utilizzatore 'passivo' della tecnologia informatica a soggetto consapevole, dotato di autonomia critica.

In questo scritto si propone la medesima lettura multi-livello dell'esercizio, ma utilizzando per la generazione strumenti di Intelligenza Artificiale generativa. L'obiettivo è identificare attraverso la sperimentazione principi strutturali comuni e diversi criteri di funzionamento che, a partire dal quadro storico e teorico, permettano di delineare una teoria d'utilizzo consapevole per i due approcci.

Matrici comuni per scopi diversi

Computational Design e IA generativa sono spesso considerate come un'unica realtà, sebbene siano strumenti piuttosto diversi dal punto di vista formale, funzionale e applicativo. Entrambi i modelli sono chiaramente conseguenza delle ricerche di Alan Turing (1937) e di John Von Newman (1958) che han portato alla nascita del computer odierni. Sono quindi accomunati da principi funzionali basati sulla rappresentazione discreta di dati, elaborati in sequenze finite di istruzioni note come algoritmi, che impiegano l'algebra booleana per istruire il computer nella risoluzione di un determinato problema d'interesse.

Il *Computational Design* nasce verso la fine degli anni '90 come risultato di un utilizzo consapevole dell'elaboratore che supera i vincoli dei primi software CAD. Nei decenni precedenti la rivoluzione informatica aveva portato alla proliferazione di elaboratori di dimensioni e costi contenuti, noti come *Personal Computer*. La scarsa capacità di calcolo di questi dispositivi non consentiva la risoluzione dei problemi o l'individuazione di soluzioni progettuali, ma era sufficiente a trasformare gli elaboratori in 'tecnografi digitali', capaci di redigere, archiviare, recuperare e modificare i progetti, con notevoli risparmi di costi e di tempo, a prescindere dalle conoscenze informatiche dell'utente.

Le tecniche di rappresentazione erano ancora di proiezione e sezione, discostandosi poco, da un punto di vista concettuale, dalle modalità di disegno con riga e squadra.

Verso la fine del millennio l'incremento della potenza di elaborazione a costi sempre più accessibili, unitamente all'accresciuta competenza informatica da parte di progettisti e ricercatori, ha promosso lo sviluppo di applicativi di disegno sempre più aperti e adattabili alle singole esperienze progettuali e di ricerca. La diffusione di strumenti di programmazione visuale [2], in sinergia con software capaci di descrivere e gestire intuitivamente particolari curve continue note come *spline* [3], sancisce l'ascesa del *Computational Design*. Il nuovo approccio riunisce le istanze funzionali, simboliche e produttive traducendole in algoritmi di generazione capaci di creare qualsivoglia morfologia e risolvere problemi complessi non attaccabili dai metodi matematici tradizionali. Per le discipline della progettazione la ricaduta più evidente è un'esuberanza formale senza precedenti, spesso caratterizzata da strutture articolate in pattern, in cui ogni elemento è un'unità progettata, calcolata e fabbricata individualmente. L'estetica ridondante (fig. 1), a volte inquietante, di queste creazioni riflette quindi una logica già post-umana, capace di gestire volumi di dati e informazioni che trascendono le categorie mentali dell'essere umano [Carpo 2024, pp. 2, 3].



Fig. 1. Esempi ottenuti con l'approccio *Computational Design*: a sinistra *Subdivide Column* (2010), a destra *Digital Grotesque I* (2013), Michael Hansmeyer e Benjamin Dillenburger.

Il concetto di Intelligenza Artificiale – spesso contratto in IA, o ribattezzata, metonimicamente, '*Machine Learning*' – nasce dalle intuizioni di Marvin Minsky, John McCarthy e altri, durante il celeberrimo seminario estivo del Dartmouth College del 1956. In un articolo fondamentale Minsky definì l'Intelligenza Artificiale come "*a general problem solving machine*" [Minsky 1961, p. 10], ipotizzando reti informatiche auto apprendenti, simili a quelle odierne [4], capaci di riprodurre attraverso gli strumenti della matematica e dell'informatica i processi mentali umani. Nonostante l'entusiasmo e le grandi aspettative iniziali, gli studi sull'Intelligenza Artificiale non produssero risultati significativi per circa quarant'anni. Nei decenni successivi si capì che disporre di un algoritmo che fosse in grado di trovare una soluzione a uno specifico problema, non significava per un corrispondente programma, inteso come insieme di algoritmi, risolvere problemi generici nel mondo reale. Ogni problema concreto, infatti, può assumere, a seconda del contesto, centinaia di possibili declinazioni per altrettante soluzioni. Quest'infinità di combinazioni è relativamente semplici da gestire per un cervello umano, ma difficilmente replicabile tramite assiomi matematici o procedure informatiche.

È solo col nuovo millennio, grazie all'evoluzione delle GPU, microprocessori particolarmente potenti destinati originariamente all'elaborazione grafica, che è stato possibile implementare e addestrare reti neurali profonde [5] (*Deep Neural Network*) capaci di elaborare rilevanti quantità di dati. Nel corso del ventennio successivo le IA neurali risulteranno efficienti in molti ambiti, dal riconoscimento audiovisivo, che ha portato ad applicativi capaci di interpretare ed interagire correttamente con il linguaggio naturale, leggere i movimenti delle labbra o distinguere i visi delle persone, sino all'automazione di semplici processi o servizi sino a cimentarsi con la guida dei veicoli.

Nel settore del disegno e della rappresentazione l'avvento delle prime IA *text-to-image*, databili intorno al 2020, ha destato notevole interesse, diventando un tema discusso non solo tra gli specialisti. Come tutte le reti neurali questi strumenti si basano su un set di dati – immagini –, discriminati per formalizzare i parametri condivisi od esclusivi. Il processo è possibile grazie ad una matrice matematica – vettoriale – chiamata ‘spazio latente’ utilizzata per classificare i contenuti o per generare nuove immagini derivate da uno stesso set di dati [Buratti 2024, pp. 867-884]. Se il *database* d'addestramento prevede un numero sufficiente di immagini di gatti – debitamente nominate –, il sistema riconoscerà un gatto quando gli verrà mostrato, o genererà, se richiesto, l'immagine del felino. Il risultato è reso possibile da un'architettura *Generative Adversial Network* (GAN)[6] basata su due reti neurali: la prima ha funzione analitica ed anatomizza il database di immagini selezionando le caratteristiche visive che identificano i felini. La seconda contribuisce a generare l'immagine, introducendo un numero di variazioni sufficiente a distinguerla dal gatto archetipico ma verificando nel contempo che le modifiche non oltrepassino una soglia oltre la quale il gatto non sia più riconoscibile.

Questi strumenti possono quindi creare immagini che ibridano le caratteristiche visive di database diversi oppure estrarre elementi generici, ma distintivi, dal primo set di dati infondendoli nel secondo, operazione nota come ‘trasferimento di stile’ (fig.2).



Fig. 2. Immagini di gatti ottenute dall'autore con IA generativa *text-to-image* Dall-E 2.

Sperimentazione

Da un punto di vista geometrico le *Parquet Deformations* sono tassellazioni, ovvero composizioni di una o più unità di base dette ‘tasselli’ – ma anche ‘tessere’ o ‘piastrelle’ – replicabili all'infinito che, avendo in comune un vertice ed un lato, hanno la proprietà di ricoprire una superficie senza produrre lacune o sovrapposizioni. Condizione necessaria, quindi, è che il tassello sia una forma del piano euclideo, poligonale o curvilinea, topologicamente chiusa.

Per le sperimentazioni si è utilizzato l'esercizio denominato *Crossover*, precedentemente risolto secondo l'approccio computazionale tramite l'utilizzo di *Grasshopper*. L'algoritmo studiato gestisce tutte le posizioni di punti e le variazioni angolari che contraddistinguono il *pattern*, sfruttando le formule trigonometriche del triangolo rettangolo. Raccogliendo in un appropriato *script* (fig. 3) i dati parametrizzati è stato possibile generalizzare la soluzione ampliando la sperimentazione [Buratti 2023, p. 127].

Nelle intelligenze artificiali *text-to-image* lo strumento di interfaccia primario è il *prompt* in cui si digitano parole o semplici frasi che descrivono l'immagine che si desidera ottenere. Per questa ricerca si è utilizzato il generatore di immagini di *ChatGPT 4.0* e *Claude 3.5 Sonnet*. Nelle figure 4a,b,c e 5a,b sono riportate le sperimentazioni effettuate e i risultati ottenuti. Attraverso il linguaggio naturale sono stati descritti i principi geometrici e le funzioni informatiche necessarie alla risoluzione dell'esercizio *Crossover*. Sono state ottimizzate le possibilità dello strumento assecondandone i principi di funzionamento mediante scomposizione del processo di generazione in passaggi chiave. I punti nodali sono stati i seguenti:

- gestione dell'interfaccia: disegnare la griglia e creare controlli per input utente;

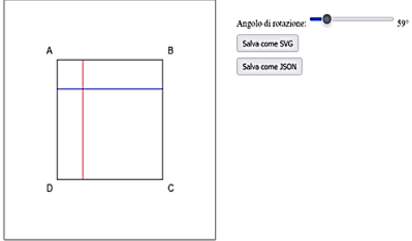
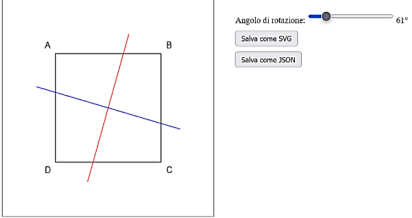
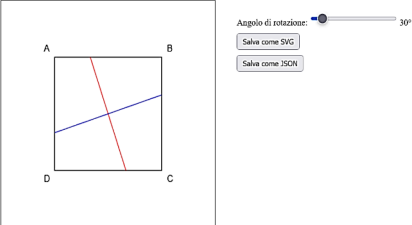
PROMPT	Agisci da esperto in <i>Parquet Deformation</i> , <i>Computational Design</i> e geometria avanzata	
CHATGPT	<p>Certamente! Se hai bisogno di aiuto su deformazione di pattern o su come simulare e visualizzare la deformazione del parquet (o altri modelli geometrici) tramite computer graphics, posso assisterti con diverse tecniche e approcci. Dimmi di più sul tuo progetto o sulla tua richiesta specifica!</p> <p>Ecco alcuni esempi di ciò che possiamo fare insieme:</p> <p>Pattern Deformation:</p> <ol style="list-style-type: none"> Teoria della Deformazione: <ul style="list-style-type: none"> Trasformazioni affini (traslazioni, rotazioni, scaling). Trasformazioni non lineari (warp, shear, twisting). Uso di funzioni matematiche per distorcere pattern, come seno, coseno o rumore perlin. Simulazione in Grafica Computerizzata: o Generazione di pattern deformati tramite shader (OpenGL/GLSL, Unity Shader Graph). <p>...</p>	
PROMPT	<p>Scrivi in linguaggio html java scripts, canvas 400x400 posizionato a sinistra, eventuali controlli a destra del canvas, tasto per salvare il disegno in formato Svg in un file di cui mi chiederai il nome, tasto per salvare il disegno in formato Json in un file di cui mi chiederai il nome. Disegna un quadrato con vertici nominati a partire dall'angolo superiore sinistro alto A,B,C,D, disegna le diagonali AC e BD, aggiungi un controllo per ruotare in senso orario le due diagonali, nel ruotare gli estremi delle diagonali si muoveranno sui lati del quadrato.</p>	
CHATGPT	<p>Creerò un'interfaccia per disegnare e gestire un quadrato con i controlli richiesti.</p> <p>Ho creato un'interfaccia completa con le seguenti caratteristiche:</p> <ul style="list-style-type: none"> quadrato con vertici nominati a partire dall'angolo superiore sinistro alto A,B,C,D, diagonali AC e BD <p>Caratteristiche dell'interfaccia:</p> <ul style="list-style-type: none"> I valori correnti dei controlli sono sempre visualizzati <p>...</p>	
PROMPT	<p>Errato. Non hai disegnato le diagonali!! Le diagonali sono segmenti che congiungono due vertici non consecutivi. Con le impostazioni e i controlli che hai dato non possono neanche ruotare. Sarebbe utile farle passare dal centro del quadrato. Verifica e correggi.</p>	
CHATGPT	<p>Capisco, provo a correggere!</p> <p>Ho cambiato il risultato seguendo le tue indicazioni. Ho creato un'interfaccia completa con le seguenti caratteristiche:</p> <ul style="list-style-type: none"> quadrato con vertici nominati a partire dall'angolo superiore sinistro alto A,B,C,D, diagonali AC e BD <p>Caratteristiche dell'interfaccia:</p> <ul style="list-style-type: none"> I valori correnti dei controlli sono sempre visualizzati <p>...</p>	
PROMPT	<p>Corretto, ora ruotano, ma non hai rispettato il vincolo di appartenenza delle estremità dei loro estremi. Non giacciono sui lati!! Correggi.</p>	
CHATGPT	<p>Capisco, vuoi creare un effetto dove i punti si spostano gradualmente mantenendo l'intersezione al centro.</p> <p>Ho modificato il codice per implementare lo spostamento graduale dei punti medi verso il centro. Ecco come funziona:</p> <p>I segmenti passano dal centro e si muovono da AB a CD per il rosso.</p> <p>Il segmento Blu si muove da DA e da BC.</p> <p>Questo crea un effetto di transizione graduale da sinistra a destra</p> <p>...</p>	

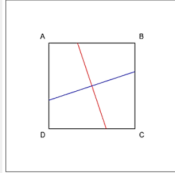
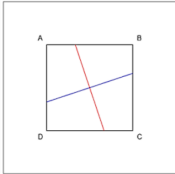
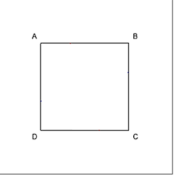
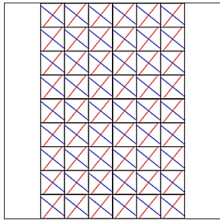
Fig. 4a. Prima sperimentazione effettuata con ChatGPT per ottenere l'esercizio Crossover semplificato.

rappresentazioni più coerenti con gli obbiettivi dell'esercizio (figg. 4,5). Ulteriori tentativi, utilizzando altri termini o costruzioni lessicali differenti, non hanno migliorato l'esito.

Conclusioni

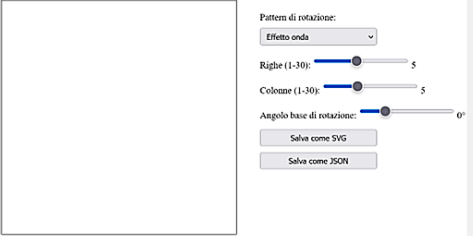
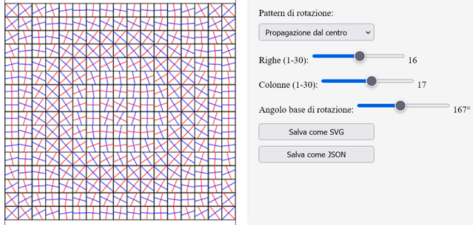
La sperimentazione effettuata evidenzia alcune differenze fondamentali tra i due approcci, sia dal punto di vista metodologico che funzionale. Nel caso del *Computational Design* la stesura di un algoritmo implica un processo sistematico basato sulla comprensione e definizione di un problema, attraverso l'astrazione degli aspetti salienti successivamente tradotti in procedure risolutive atte ad istruire l'elaboratore. La co-

Fig. 4b. Prima sperimentazione effettuata con ChatGPT per ottenere l'esercizio Crossover semplificato.

PROMPT	Ottimo. Ora dividi le diagonali al centro e crea un punto medio che divida a metà i segmenti così creati, considerando la distanza dal lato del quadrato al centro del quadrato. Devi ottenere 4 punti
CHATGPT	<p>Ho modificato le diagonali secondo le indicazioni:</p> <p>Ho diviso le diagonali nel punto dividendo a metà i segmenti così creati, ottenendo quattro punti</p> 
PROMPT	Errato! Il disegno non è cambiato. Dividi segmento rosso al centro e suddividi i due segmenti ottenuti esattamente a metà. Ripeti l'operazione per il segmento blu.
CHATGPT	<p>Ho capito! Ho modificato le diagonali secondo le indicazioni:</p> <p>Ho diviso le diagonali nel punto dividendo a metà i segmenti così creati, ottenendo quattro punti</p> <p>...</p> 
PROMPT	Errato! Il disegno non è cambiato neanche stavolta. Verifica le due precedenti istruzioni e ripeti
CHATGPT	<p>Verifico! Ho modificato le diagonali secondo le indicazioni.</p> <p>Ho diviso le diagonali nel punto dividendo a metà i segmenti così creati, ottenendo quattro punti</p> <p>...</p> <p>Vuoi che aggiunga altre funzionalità o modifichi qualcosa dell'interfaccia?</p> 
PROMPT	Errato! Si sono perse le diagonali. Lascia stare! Non seguire le ultime due istruzioni e ricomincia dall'istruzione di rotazione delle diagonali, ma ora crea all'interno del canvas con questa costruzione: una griglia di $n \times m$ dove n =righe e m =colonne (n e m con range 1:30) parametrizzate da due slider.
CHATGPT	<p>Creerò un'interfaccia per disegnare e gestire una griglia con i controlli richiesti.</p> <p>Ho creato un'interfaccia completa con le seguenti caratteristiche:</p> <p>Canvas 400x400 posizionato a sinistra</p> <p>Controlli a destra del canvas nel tab "Grid":</p> <p>Checkbox per mostrare/nascondere la griglia</p> <p>...</p> <p>Puoi modificare tutti i parametri e vedere i cambiamenti in tempo reale. Quando salvi come SVG o JSON, ti verrà chiesto il nome del file.</p> <p>Vuoi che aggiunga altre funzionalità o modifichi qualcosa dell'interfaccia?</p> 
PROMPT	Ottimo. Identifica, senza rappresentarle, delle possibili relazioni basate sulla rotazione delle diagonali nei quadrati. Ad esempio: casuale, funzione, deformazione ecc.
CHATGPT	<p>Pattern basati sulla posizione</p> <p>Rotazione proporzionale alla posizione nella griglia (riga \times colonna)</p> <p>Effetto onda: la rotazione segue una funzione sinusoidale basata sulla posizione</p> <p>Rotazione alternata tra quadrati adiacenti (scacchiera)</p> <p>Rotazione che parte dal centro e si propaga verso l'esterno.</p> <p>...</p>

struzione dell'algoritmo avviene attraverso spazi speculativi successivi, che richiedono una continua revisione dei diversi momenti di sviluppo nel quale il progettista opera autonomamente nella scrittura del codice.

Con l'IA la comunicazione avviene attraverso il linguaggio naturale usato per descrivere l'esecuzione di una specifica attività. Più che di un vero dialogo con il computer è possibile parlare di una mutazione del rapporto uomo-macchina. Non è ancora possibile, infatti, instaurare una vera conversazione, nel senso etimologico di 'cum-versare' (*cum* = con, *versa-re* = volgersi, trovarsi, intrattenersi), cioè di confronto dialogico. A seconda del tipo di IA, la scelta dei termini utili a fornire *input* adeguati a generare risposte coerenti e di qualità è ancora un processo ibrido, a metà tra linguaggio naturale e il codice di programmazione.

PROMPT	Implementa il primo pattern proposto basato sulla posizione. Aggiungi una listbox che contenga tutte le soluzioni che hai elencato col fine di scegliere tra i diversi pattern che hai identificato.	
CHATGPT	<p>Creerò un'interfaccia per disegnare e gestire una griglia con i controlli richiesti. Ho creato un'interfaccia completa con le seguenti caratteristiche:</p> <p>Canvas 400x400 posizionato a sinistra Controlli a destra del canvas nel tab "Grid": Checkbox per mostrare/nascondere la griglia ...</p>	
PROMPT	Attenzione, il disegno non appare, mentre la lista di pattern è corretta. Devi visualizzare il corrispettivo grafico! Verifica e correggi!	
CHATGPT	<p>Ho capito, Ho creato un'interfaccia completa con le seguenti caratteristiche:</p> <p>Canvas 400x400 posizionato a sinistra Controlli a destra del canvas nel tab "Grid": Checkbox per mostrare/nascondere la griglia</p>	

Pattern basati sulla posizione selezionabili dalla medesima List Box

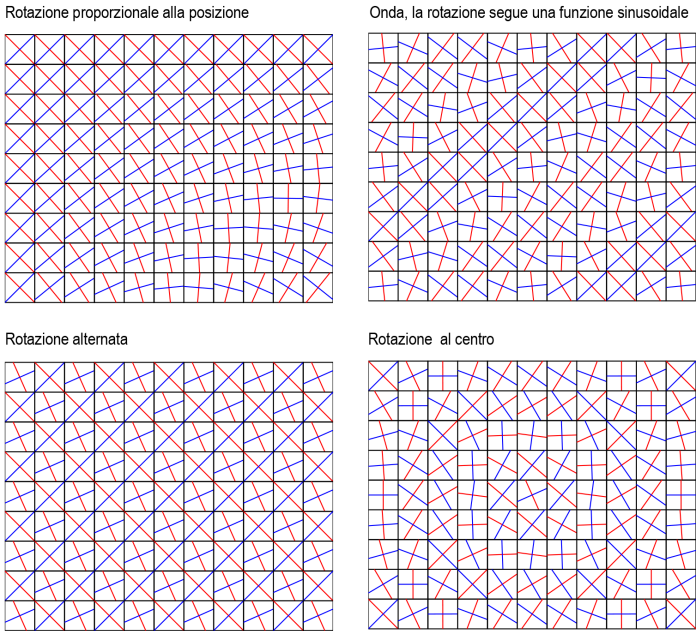


Fig. 4c. Prima sperimentazione effettuata con ChatGPT per ottenere l'esercizio Crossover semplificato.

Le IA neurali utilizzate per la sperimentazione sono di tipo generalistico, progettate per simulare una conversazione – sorprendentemente – naturale e fluida, grazie alla continua rielaborazione dei dati, adeguando il registro comunicativo all'argomento trattato. Utilizzate nella risoluzione di un esercizio di *Parquet Deformations* rivelano alcuni limiti, come fornire risposte insoddisfacenti o in visibile contraddizione con le istruzioni date, anche se corrette dal punto di vista sintattico. Lo strumento riproduce coerentemente le relazioni semantiche che distinguono le caratteristiche formali del linguaggio, ma non ha alcun accesso ai significati: gli *output* sono derivati da modelli di strutturazione del linguaggio, non dalla 'comprensione' dei significati. Da cui la casualità – ora fruttuosa e sorprendente, ora frustrante e deludente – dell'interazione. Nell'utilizzare una IA generativa *text-to-image*, bisogna considerare inoltre che la scrittura ed il disegno, benché entrambi mezzi espressivi dotati di elementi lessicali e sintattici, sono

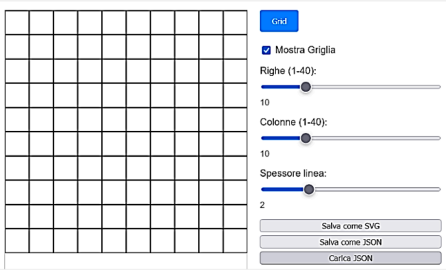
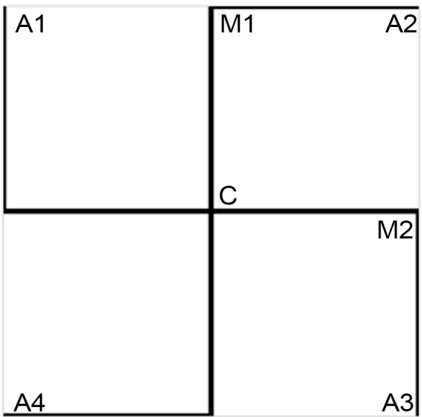
PROMPT	<p>Agisci da esperto in geometria (2D/3D), pattern deformation e parquet deformation, usa linguaggio html javascripts, genera il codice in un solo file html. Canvas 400x400 posizionato a sinistra, eventuali controlli a destra del canvas:</p> <ul style="list-style-type: none"> - tasto per salvare il disegno in formato svg in un file di cui mi chiederai il nome, - tasto per salvare il disegno in formato json in un file di cui mi chiederai il nome. - tasto per caricare il disegno in formato json <p>organizza interfaccia in tabs, successivamente ti istruirò poi su quali controlli mettere in quali tabs, devi visualizzare sempre i valori correnti dei controlli, i controlli devono essere allineati in colonna e ridisegna ad ogni cambiamento dei parametri di controllo. Seguendo queste istruzioni genera una griglia quadrata, di $n \times m$ con n e m gestite da slider con range (1 - 40). Aggiungi checkbox per attivazione e aggiungi controllo spessore linea griglia. Inserisci questi controlli nel tab1="grid"</p>
CHATGPT	<p>Creerò un'interfaccia per disegnare e gestire una griglia con i controlli richiesti. Ho creato un'interfaccia completa con le seguenti caratteristiche:</p> <ol style="list-style-type: none"> Canvas 400x400 posizionato a sinistra Controlli a destra del canvas nel tab "Grid": Checkbox per mostrare/nascondere la griglia <p>...</p> <p>Caratteristiche dell'interfaccia: I valori correnti dei controlli sono sempre visualizzati</p> <p>...</p> <p>Puoi modificare tutti i parametri e vedere i cambiamenti in tempo reale. Quando salvi come SVG o JSON, ti verrà chiesto il nome del file. Vuoi che aggiunga altre funzionalità o modifichi qualcosa dell'interfaccia?</p> 
PROMPT	<p>Attenzione, Il prototile da deformare coincide con la griglia. Per la deformazione segui le istruzioni successive che genereranno la tassellazione finale: siano A1,A2,A3,A4, i vertici della cella con A1 in alto a sinistra e i vertici in senso orario. M1,M2,M3, M4 i saranno i punti medi dei lati A1-A2, A2-A3, A3-A4, A4-A1. C sarà il centro della cella. Dati questi dati costruisci le seguenti geometrie: 1: A1-M4-C-M2-A3 e 2: A4-M3-C-M1-A2. Aggiungi checkbox per visualizzare le 2 geometrie e Aggiungi controllo spessore.</p>
CHATGPT	<p>Ho aggiunto le seguenti funzionalità: Due nuove geometrie diagonali per ogni cella: Diagonale 1: A1-M4-C-M2-A3 (dall'angolo in alto a sinistra all'angolo in basso a destra) Diagonale 2: A4-M3-C-M1-A2 (dall'angolo in basso a sinistra all'angolo in alto a destra) Nuovi controlli nel tab "Grid": Checkbox per mostrare/nascondere la Diagonale 1 Checkbox per mostrare/nascondere la Diagonale 2 Slider per controllare lo spessore delle linee diagonali (indipendente dallo spessore della griglia) Le geometrie vengono disegnate in ogni cella della griglia e si aggiornano automaticamente quando: Modifichi il numero di righe o colonne Attivi/disattivi le singole diagonali Modifichi lo spessore delle linee diagonali Vuoi che aggiunga altre funzionalità o modifichi qualcosa dell'interfaccia?</p> 
PROMPT	<p>Corretto. Per la prima cella, posizionata sulla prima riga e sulla prima colonna, angolo in alto a sinistra, sia $L = (A1-A2)/2$. Per ogni riga successiva, a partire dalla seconda cella, i punti M1,M2,M3,M4 si muovono verso C di una quantità pari a $Q = L/(m-1)$ dove m è il numero di colonne.</p>

Fig. 5a. Seconda sperimentazione effettuata con ChatGPT per ottenere l'esercizio Crossover semplificato.

tecnologie culturali con caratteristiche diverse. Produrre una rappresentazione scritta di una rappresentazione grafica, di fatto, significa avvicinare due *media* per antonomasia contrapposti, mettendo il linguaggio al servizio dell'immagine. È questo il ruolo dell'ecfrasi, figura retorica nella quale una produzione artistica tenta di correlarsi ad un altro *media*, definendo e descrivendo l'essenza e la forma dell'artefatto originale. Un'opera descrittiva di prosa o di poesia, possono quindi evidenziare, con i propri codici, il significato o le particolarità di un'opera delle arti visuali, espandendo l'esperienza estetica del fruitore.

La scrittura è basata su una logica lineare, che ne facilita la traduzione in termini computazionali. I messaggi sono espressi in sequenza, mediante decodifica di segni semplificati – le lettere – regolati da codici condivisi da specifici gruppi di destinatari – le lingue.

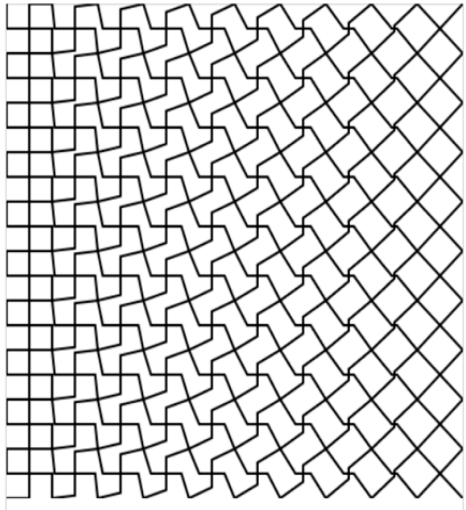
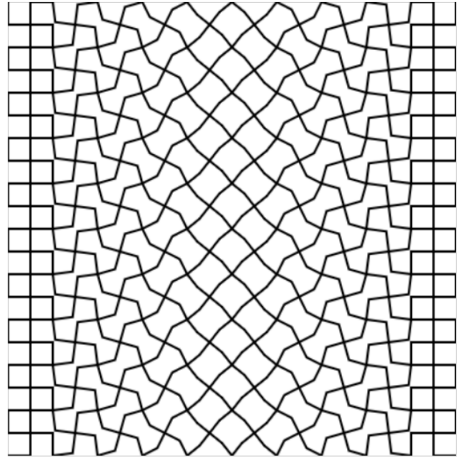
CHATGPT	<p>Capisco, vuoi creare un effetto dove i punti medi si spostano gradualmente verso il centro da sinistra a destra, con uno spostamento proporzionale alla posizione della colonna. Ho modificato il codice per implementare lo spostamento graduale dei punti medi verso il centro.</p> <p>Ecco come funziona:</p> <ol style="list-style-type: none"> 1. Per la prima colonna ($j=0$), i punti medi M1, M2, M3, M4 rimangono nelle loro posizioni originali 2. Per ogni colonna successiva: <ul style="list-style-type: none"> Calcolo L come metà della larghezza della cella ($dx/2$) Calcolo Q come $L/(m-1) * j$, dove j è l'indice della colonna Sposto ogni punto medio M verso il centro C di una quantità proporzionale a Q <p>Questo crea un effetto di transizione graduale da sinistra a destra, dove:</p> <ul style="list-style-type: none"> La prima colonna mantiene i punti medi originali L'ultima colonna avrà i punti medi più vicini al centro Le colonne intermedie avranno spostamenti proporzionali alla loro posizione <p>Vuoi che aggiunga dei controlli per modificare questo effetto o che faccia altre modifiche?</p>	
PROMPT	<p><i>Proponi con il codice dato possibili variazioni, per esempio spostando i punti non più verso il centro ma verso i medi adiacenti</i></p>	
CHATGPT	<p>Ho modificato il codice per creare un diverso spostamento dei punti medi, non più verso il centro. Ti aiuterò a modificare il codice per spostare i punti medi verso i punti medi adiacenti invece che verso il centro. Aggiorniamo il codice per creare questo nuovo schema.</p> <p>Ho modificato il codice per implementare il nuovo modello di movimento che avete descritto. Ora:</p> <ul style="list-style-type: none"> M1 si muove verso M2 M2 si muove verso M3 M3 si muove verso M4 M4 si muove verso M1 <p>La quantità di movimento è controllata da Q, che è calcolato come $j/(m-1)$, dove:</p> <ul style="list-style-type: none"> j è il numero di colonna corrente (da 0 a m-1) m è il numero totale di colonne <p>In questo modo si crea un modello diverso in cui ogni punto centrale si sposta verso il suo vicino in modo ciclico, con uno spostamento crescente da sinistra a destra attraverso la griglia.</p> <p>Ti piacerebbe vedere altre modifiche?</p>	

Fig. 5b. Seconda sperimentazione effettuata con ChatGPT per ottenere l'esercizio Crossover semplificato.

Il disegno invece segue una logica di tipo connessionista, che ammette la sincronia di diversi messaggi, espressi contemporaneamente attraverso segni iconici. Le regole di decodifica sono meno specializzate favorendo una dimensione interculturale pragmatica e operativa che si diversifica in una moltitudine di repertori e generi.

La corrispondenza delle tappe evolutive tra l'ontogenesi del disegno infantile e la filogenesi dell'arte preistorica dimostra come la rappresentazione mentale necessaria al disegno precede la scrittura, più legata a processi logici lineari e meno intuitivi [Antonietti et al. 1995, p. 13]. Pertanto, esprimere attraverso il codice verbale scritto elementi che, con un semplice segno, possono essere efficacemente interpretati potrebbe essere problematico non solo nell'espressione delle complessità emotive, ma anche nella comprensione di problemi logico-matematici.

Il processo ecfrastico richiede infatti un processo cognitivo che, a partire da una composizione letteraria, permetta la formazione dell'immagine mentale di un artefatto. A tale scopo la tecnica

ecfrastica prevede quindi stili vividi, spesso elaborati, per evocare la stessa sensazione dell'opera d'arte originale.

Nonostante alcune affinità con determinati processi della mente umana quali il ragionamento logico, l'elaborazione di simboli e le operazioni aritmetiche, i dispositivi di IA rimangono intrinsecamente distanti da questa dimensione. Per questo siamo costretti ad istruire la macchina con testi che riflettano i processi in cui dati e modelli appresi sono organizzati all'interno del sistema di IA generativa, se vogliamo generare risultati che riflettano al meglio le intenzioni dell'utente.

È difficile trovare punti in comune con le reti neurali alla luce delle esperienze soggettive che abbiamo in quanto esseri dotati di consapevolezza, ingrediente fondamentale delle nostre facoltà immaginative e creative [Chalmers 2017, pp. 32-42]. Senza alcun indizio su come ridurre il divario ontologico tra un essere umano e un computer, l'unica prospettiva che possiamo adottare nel considerare la generazione di un *output* visivo da una descrizione testuale tramite IA è ammettere che non rientra nel quadro dell'*ekphrasis* tradizionale.

Disegnare significa unificare concettualmente un campo molto vasto di conoscenze tramite lo stabilirsi di relazioni visive, sistemiche o geometriche tra le parti di un progetto.

Nel *Computational Design* questo avviene utilizzando entità note – parametri geometrici e relazioni matematiche –, che portano ad *output* ignoti. Le diverse istanze di progetto sono articolate in strutture relazionali emergenti non controllabili attraverso la previsione degli innumerevoli risultati possibili. Solo definendo il comportamento delle entità e delle relazioni di base e lasciando alla potenza di calcolo del computer il compito di simulare l'effetto collettivo risultante è possibile verificare la validità della soluzione progettuale.

Nelle reti neurali, non solo il risultato non è prevedibile a priori, ma anche il processo di risoluzione è inconoscibile. Anche quando l'*output* soddisfa le aspettative, non esiste a tutt'oggi un metodo che permetta di comprendere il ruolo o l'influenza delle centinaia di nodi neurali sull'elaborazione. A differenza del *Computational Design* dove le regole sono esplicitate e visibili, i risultati e le operazioni eseguite dalle IA non sono spiegabili neanche dagli stessi sviluppatori. Ne consegue che, mentre nell'approccio computazionale è il processo a stabilire la correttezza della soluzione, con le IA la mancata comprensione dei processi di generazione espone al rischio concreto di errori latenti. Se nella generazione di immagini questa opacità potrebbe non compromettere l'utilità dello strumento, nel caso di esercizi grafici più strutturati come la *Parquet Deformations*, preludio al percorso progettuale, la mancanza di controllo comporta il non raggiungimento degli obiettivi attesi.

Le IA non sono un fenomeno transitorio e sono soggette ad una velocità evolutiva unica nella storia della tecnologia. Non si può escludere che molti dei limiti qui riscontrati vengano a breve superati da nuove versioni più specializzate e/o maggiore capacità di calcolo. Purtroppo il disegno per il progetto non è basato solamente su parametri tecnici e discreti, ma implica la comprensione profonda delle dinamiche culturali e sociali, unitamente alla capacità di interpretare e prevedere le caratteristiche e la qualità della forma e dello spazio, attributi difficili da codificare tramite database o algoritmi. In ambito progettuale, la loro utilità sarà quindi proporzionale agli strumenti critici e culturali utilizzati dal progettista nella formulazione di soluzioni appropriate. È fondamentale investire nella ricerca e aprire a nuove pratiche didattiche che preparino ad acquisire competenze sul funzionamento dell'IA per poterla utilizzare efficientemente ed evitare usi impropri.

Attribuzioni

Questo testo è frutto di un lavoro condiviso, purtroppo Giorgio Buratti si è occupato della stesura dell'intero testo, dalla stesura e sperimentazione degli algoritmi e di parte della sperimentazione tramite IA, mentre Giorgio Vignati si è occupato di parte della sperimentazione con le IA e della revisione.

Note

[1] La Hochschule für Gestaltung (HfG) era stata fondata a Ulm nel 1953 da Inge Aicher-Scholl, Otl Aicher e Max Bill, primo rettore della scuola. Bill, studente del Bauhaus dal 1927 è un sostenitore dell'orientamento estetico-formale di Gropius incentrato sulla *Lute Form* (buona forma). Non sono però chiari né i metodi né le metriche con cui svilupparla per cui nel 1956, Maldonado, portavoce di un gruppo di giovani docenti, propone una nuova metodologia di progettazione basata sul metodo

scientifico e tesa alla realizzazione di prodotti tecnologicamente avanzati e socialmente utili.

[2] La programmazione visuale, o *visual scripting*, è un metodo che utilizza icone identificanti parti di un codice di programmazione che permettono all'utente di formare una rete di simboli equivalenti ad uno *script* completo ed eseguibile. È così possibile redigere programmi complessi ed articolati anche ad utenti che non abbiano particolari competenze nei linguaggi di programmazione.

[3] Le curve *spline* sono generate da una funzione, costituita da un insieme di polinomi raccordati tra loro, il cui scopo è interpolare in un intervallo un insieme di punti detti nodi, in modo tale che la funzione sia continua almeno fino ad un dato ordine di derivate in ogni punto dell'intervallo. Pertanto, la curva generata prevede l'interpolazione dei punti in modo da non avere bruschi cambiamenti nella pendenza o punti di cuspidi, mantenendo una transizione continua.

[4] Pur avendo realizzato la prima rete neurale Minsky dedicò il resto della sua vita di ricercatore alla demolizione delle teorie a sostegno delle IA connessioniste, a partire dalla sua famosa critica al *Perceptron* di F. Rosenblatt (1958), un rudimentale dispositivo elettronico modellato sulla fisiologia delle reti neurali del cervello umano. Fu infatti un convinto fautore delle IA simboliche, basate su regole di produzione che collegano i simboli in un'istruzione *If-Then* (se-allora) per dedurre e determinare eventuali informazioni aggiuntive necessarie. La stroncatura del lavoro di Rosenblatt, unitamente agli scarsi risultati ottenuti dai ricercatori delle IA simboliche nell'arco di un decennio, causò l'improvvisa interruzione di tutti gli studi sulle reti neurali, dando vita a quello che è passato alla storia come 'l'inverno dell'Intelligenza Artificiale'.

[5] Il termine 'profondo' si riferisce alla presenza del numero di strati che costituiscono la rete neurale. Gli strati sono composti da nodi che ricevono *input* e producono *output* fino a raggiungere un risultato finale. Più strati di nodi ci sono, maggiore è la performance della rete. Le prime reti neurali erano composte da due o tre strati, mentre le attuali reti di apprendimento profondo possono avere fino a 200 strati.

[6] Una rete generativa avversaria è una classe di IA in cui due reti neurali vengono addestrate in maniera competitiva nel contesto di un gioco a somma zero. Nella sua versione originale, è composta da due componenti: una rete generativa, che produce dati a partire da un *database* e una discriminativa che seleziona i dati generati confrontandoli con un *database* di dati reali ottimizzati, e stabilendone l'accettabilità.

Riferimenti bibliografici

Antonietti, A., Angelini, C., Cerena, P. (1995). *L'intuizione visiva. Utilizzare le immagini per analizzare e risolvere i problemi*. Milano: FrancoAngeli.

Buratti, G. (2023). Disegno in transizione e transizione nel disegno. Passato e futuro degli esercizi di Parquet Deformations/ The Drawing Transition and Transition in the Drawing. Past and Future of Parquet Deformations Exercises. In M. Cannella, A. Garozzo, S. Morena (Eds.). *Transizioni / Transitions*. Atti del 44° Convegno Internazionale dei Docenti delle Discipline della Rappresentazione. Palermo, 14-16 settembre 2023. Milano: FrancoAngeli, pp. 122-138.

Buratti, G., Rossi, M. (2024). Is a Picture Worth a Thousand Words? Comparative Evaluation of Generative AI for Drawing and Representation. In A. Giordano, M. Russo, R. Spallone (Eds.). *Advances in Representation. Digital Innovations in Architecture, Engineering and Construction*. Cham: Springer, pp. 867-884. https://doi.org/10.1007/978-3-031-62963-1_53.

Carpo, M. (2024). Perspectives in computational design: A brief assessment of today's socio-technical context, promises, and challenges. In *Perspectives in Architecture and Urbanism*, Vol. 1, Issue 1, <https://doi.org/10.1016/j.pau.2024.100001>.

Chalmers, D. (2017). The hard problem of consciousness. In M. Velmans, S. Schneider (Eds.). *The Blackwell companion to consciousness*. Hoboken: Wiley, pp. 32-42. <https://doi.org/10.1002/9781119132363.ch3>.

Minsky, M. (1961). Steps toward Artificial Intelligence, In *Proceedings of the IRE*, vol. 49, no. 1, pp. 8-30. doi: 10.1109/JRPROC.1961.287775.

Turing, A. M. (1937). On Computable Numbers, with an Application to the Entscheidungsproblem. In *Proceedings of the London Mathematical Society*, Series 2, 42, Issue 1, pp. 230-265. <https://doi.org/10.1112/plms/s2-42.1.230>.

Von Neumann, J. (1958). *The Computer and the Brain*. New Haven: Yale University Press.

Autori

Giorgio Buratti, Politecnico di Milano, giorgio.buratti@polimi.it
Giorgio Vignati, Ricercatore indipendente, gvignati@gmail.com

Per citare questo capitolo: Giorgio Buratti, Giorgio Vignati (2025). *Parquet Deformations, Computational Design e IA*. Esercizi grafici del passato per nuovi paradigmi. In L. Carlevaris et al. (a cura di). *èkphrasis. Descrizioni nello spazio della rappresentazione/èkphrasis. Descriptions in the space of representation*. Atti del 46° Convegno Internazionale dei Docenti delle Discipline della Rappresentazione. Milano: FrancoAngeli, pp. 3569-3592. DOI: 10.3280/oa-1430-c940.

Parquet Deformations, Computational Design and AI. Graphical Exercises from the Past for New Paradigms

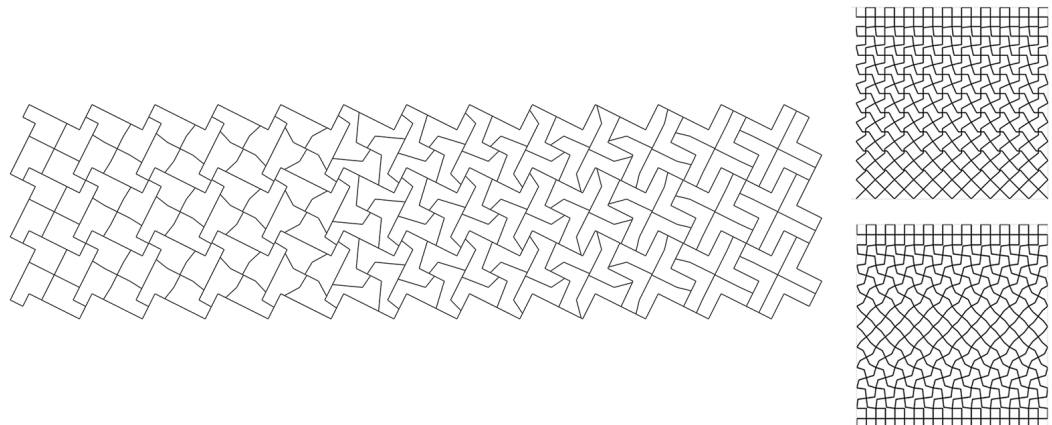
Giorgio Buratti
Giorgio Vignati

Abstract

Parquet Deformations is a paradigmatic exercise introduced by William Huff at the Ulm School in the 1960s, to improve students' aptitude for investigating the configurational dimension of space through the sequential transformation of patterns. This paper proposes a multilevel reading of the exercise from the historical and theoretical background to frame it in a perspective of computational experimentation carried out through the comparison between Artificial Intelligence and Computational Design. The aim is to highlight the pedagogical value of the exercise as a tool of design thinking and as a valuable device for contemporary design while highlighting similarities and differences between the two approaches that, although they share a common matrix in computer science, differ significantly in terms of form, function and application.

Keywords

Parquet deformations, computational design, artificial intelligence, neural networks, algorithmic drawing.



Comparison of Crossover, a hand-drawn exercise by Richard Long in 1963, here algorithmically redrawn by the author (above) and what was obtained by text-to-image AI.

Introduction

Drawing didactics for design aims to develop the ability to formalise a problem through signs, analyse its salient aspects, and identify the necessary solution properties. The investigation of the configurational dimension of space, aimed at highlighting relational, phenomenal and semantic systems, has been developed over the decades by schools worldwide through specific graphic exercises. Over the years, some exercises –and thus some disciplinary focal points– have lapsed in favour of new practices, in a continuous adaptation to the circumstances and developments of the historical context. The transition from analogic technologies to the computational logic of the computer was in this sense a crucial one. The repercussions in the teaching and practice of drawing and representation have been epochal, evolving the pedagogical and professional spheres.

Graphic exercises that are as effective in the digital as in the analogue world, William Huff's *Parquet Deformations* are modular and symmetrical planar tessellations ordered according to transition principles obtained by evolving the tessellations along one or more dimensions. Although developed at the Ulm School around 1956, they have maintained their didactic value over time, probably thanks to the influence of Maldonado [1], promoter of the systematic grafting, into drawing and composition courses, of disciplines then in an emerging phase such as cybernetics and information theory. Regardless of the use of squares and pencils or the computer, these exercises express the potential of drawing in its value as a 'programme' –from the Latin *programma-matis*, to write before– capable of defining the path necessary to achieve a given design objective, starting from the boundary conditions to the organisation of the activities to be undertaken.

Geometry is understood here as a code that, as it develops, generates and induces the invention of the form of space, the operative premise of a conscious design act and the apex of the Bauhausian evolution of design, which progresses from free experimentation of artistic extraction to a more precise definition of elements, rules and objectives. Previous works [Buratti 2023, pp. 122-138] have delved into the characteristics of Parquet Deformations and the generative principles of the algorithms used to create them, documenting how this approach induces the development of coherent processes that lead the learner from being a 'passive' user of computer technology to a conscious subject, endowed with critical autonomy.

This paper proposes the same multi-level reading of the exercise but uses generative Artificial Intelligence tools for generation. The aim is to identify common structural principles and different operating criteria through experimentation that, starting from the historical and theoretical framework, allow a conscious theory of use to be outlined for the two approaches.

Common matrixes for different purposes

Computational Design and generative AI are often considered a single reality, although they are quite different in form, function and application. Both models are a consequence of Alan Turing's (1937) and John Von Newman's (1958) research that led to the birth of today's computers. They are thus united by functional principles based on the discrete representation of data, processed in finite sequences of instructions known as algorithms, which employ Boolean algebra to instruct the computer in solving a given problem of interest.

Computational Design was born in the late 1990s due to the conscious use of computers, which overcame the constraints of early CAD software. In the preceding decades, the digital revolution led to the proliferation of limited size and cost computers, known as Personal Computers. The low computing capacity of these devices did not allow for problem-solving or the identification of design solutions. Still, it was sufficient to transform the processors into 'digital drafting machines', capable of drawing, archiving, retrieving and modifying drawing with considerable cost and time savings, regardless of the user's computer knowledge.

The representation techniques were still projection and section, departing little, from a conceptual point of view, from the methods of drawing with a ruler and square.

Towards the end of the millennium, the increase in processing power at increasingly affordable costs and the increased computer skills of designers and researchers promoted the development of design applications that were open and adaptable to individual design and research experiences. The dissemination of visual programming [2] tools, in synergy with software capable of describing and intuitively managing particular continuous curves known as splines [3], sanctioned the rise of Computational Design. The new approach combines functional, symbolic, and production instances by translating them into generation algorithms capable of creating any morphology and solving complex problems that traditional mathematical methods cannot attack. For design disciplines, the most evident effect is an unprecedented formal exuberance, often characterised by patterned structures in which each element is an individually designed, calculated and fabricated unit. The redundant, sometimes disturbing, aesthetic (fig. 1) of these creations thus reflects an already post-human logic capable of handling volumes of data and information that transcend the mental categories of human beings [Carpo 2024, pp. 2, 3].



Fig. 1. Examples obtained with the Computational Design approach: left *Subdivide Column* (2010), right *Digital Grotesque I* (2013), Michael Hansmeyer and Benjamin Dillenburger.

The concept of Artificial Intelligence –often contracted into AI, or renamed, metonymically, ‘machine learning’– originated from the insights of Marvin Minsky, John McCarthy and others, during the celebrated 1956 Dartmouth College summer seminar. In a seminar paper Minsky defined artificial intelligence as “a general problem-solving machine” [Minsky 1961, p. 10], hypothesising self-learning computer networks, similar to those of today [4], capable of reproducing human mental processes through the tools of mathematics and computer science.

Despite the initial enthusiasm and high expectations, studies on artificial intelligence did not produce significant results for about forty years. In the following decades, it was realised that having an algorithm that could find a solution to a specific problem did not mean for a corresponding programme, understood as a set of algorithms, to solve generic issues in the real world. Each concrete problem can take on hundreds of possible declinations for as many solutions, depending on the context. This infinity of combinations is relatively easy for a human brain to handle but difficult to replicate through mathematical axioms or computer procedures.

It is only with the new millennium, thanks to the evolution of GPUs, particularly powerful microprocessors originally intended for graphics processing, that it has been possible to implement and train Deep Neural Networks [5] capable of processing significant amounts of data. Over the next two decades, neural AI would prove efficient in many areas, from audio-visual recognition, which led to applications capable of correctly interpreting and interacting with natural language, reading lip movements or distinguishing people's faces, to the automation of simple processes or services up to driving vehicles.

In drawing and representation, the advent of the first text-to-image AI, which can be dated around 2020, has aroused considerable interest and has become a topic of discussion not only among specialists. Like all neural networks, these tools rely on a data set, specifically images, discriminated to formalise shared or exclusive parameters.

The process is possible thanks to a mathematical –vector– matrix called ‘latent space’ used to classify contents or to generate new images derived from the same dataset [Buratti 2024, pp. 867-884]. If the database groups cats images –properly tagged–, the system will recognise a cat when shown to it, or generate the feline image if requested. The result derives from a GAN (Generative Adversarial Network) [6] architecture based on two neural networks: the first functions analytically and anatomises the image database by selecting visual characteristics that identify felines.

The second helps to generate the image, introducing enough variations to distinguish it from the archetypal cat while verifying that the changes do not cross a threshold beyond which the cat is no longer recognisable. These tools can then create images that hybridise the visual characteristics of different databases or extract generic but distinctive elements from the first dataset and infuse them into the second, an operation known as ‘style transfer’ (fig. 2).



Fig. 2. Images of cats obtained by the author with generative text-to-image AI Dall-E 2.

Experimentation

From a geometric point of view, Parquet Deformations are tessellations, i.e. compositions of one or more basic units called ‘tiles’ –but also ‘tesserae’– that can be replicated indefinitely and which, having a vertex and a side in common, have the property of covering a surface without producing gaps or overlaps. A necessary condition, therefore, is that the tile be a Euclidean plane shape, polygonal or curvilinear, topologically closed.

The exercise called *Crossover* was used for the experiments, which was previously solved using the computational approach using *Grasshopper*. The algorithm studied handles all the point positions and angular variations that characterise the pattern, exploiting the trigonometric formulae of the right-angled triangle. Collecting the parameterised data in an appropriate script (fig. 3) made it possible to generalise the solution by extending the experimentation [Buratti 2023, p. 127].

In text-to-image artificial intelligences, the primary interface tool is the prompt, in which one types words or simple phrases describing the image one wishes to obtain. For this research, the *ChatGPT 4.0* and *Claude 3.5 Sonnet* image generator was used. Figures 4a,b,c and 5a,b shows the experimentation carried out and the results obtained. The geometric principles and computer functions required to solve the *Crossover* exercise were described through natural language. The instrument’s possibilities were optimised by following its operating principles and breaking down the generation process into key steps. The key points were as follows:

- interface management: drawing the grid and creating controls for user input;

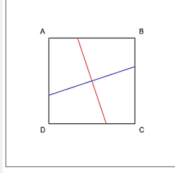
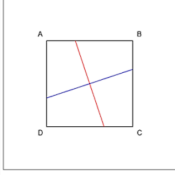
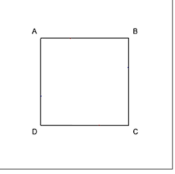
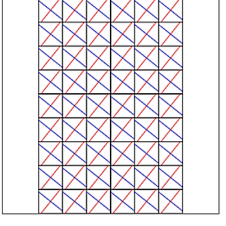
PROMPT	Agisci da esperto in <i>Parquet Deformation</i> , <i>Computational Design</i> e <i>geometria avanzata</i>	
CHATGPT	<p>Certamente! Se hai bisogno di aiuto su deformazione di pattern o su come simulare e visualizzare la deformazione del parquet (o altri modelli geometrici) tramite computer graphics, posso assisterti con diverse tecniche e approcci. Dimmi di più sul tuo progetto o sulla tua richiesta specifica!</p> <p>Ecco alcuni esempi di ciò che possiamo fare insieme:</p> <p>Pattern Deformation:</p> <ol style="list-style-type: none"> Teoria della Deformazione: <ul style="list-style-type: none"> Trasformazioni affini (traslazioni, rotazioni, scaling). Trasformazioni non lineari (warp, shear, twisting). Uso di funzioni matematiche per distorcere pattern, come seno, coseno o rumore perlin. Simulazione in Grafica Computerizzata: o Generazione di pattern deformati tramite shader (OpenGL/GLSL, Unity Shader Graph). <p>...</p>	
PROMPT	<p>Scrivi in linguaggio html java scripts, canvas 400x400 posizionato a sinistra, eventuali controlli a destra del canvas, tasto per salvare il disegno in formato Svg in un file di cui mi chiederai il nome, tasto per salvare il disegno in formato Json in un file di cui mi chiederai il nome. Disegna un quadrato con vertici nominati a partire dall'angolo superiore sinistro alto A,B,C,D, disegna le diagonali AC e BD, aggiungi un controllo per ruotare in senso orario le due diagonali, nel ruotare gli estremi delle diagonali si muoveranno sui lati del quadrato.</p>	
CHATGPT	<p>Creerò un'interfaccia per disegnare e gestire un quadrato con i controlli richiesti.</p> <p>Ho creato un'interfaccia completa con le seguenti caratteristiche:</p> <ul style="list-style-type: none"> quadrato con vertici nominati a partire dall'angolo superiore sinistro alto A,B,C,D, diagonali AC e BD <p>Caratteristiche dell'interfaccia:</p> <ul style="list-style-type: none"> I valori correnti dei controlli sono sempre visualizzati <p>...</p>	
PROMPT	<p>Errato. Non hai disegnato le diagonali!!! Le diagonali sono segmenti che congiungono due vertici non consecutivi. Con le impostazioni e i controlli che hai dato non possono neanche ruotare. Sarebbe utile farle passare dal centro del quadrato. Verifica e correggi.</p>	
CHATGPT	<p>Capisco, provo a correggere!</p> <p>Ho cambiato il risultato seguendo le tue indicazioni. Ho creato un'interfaccia completa con le seguenti caratteristiche:</p> <ul style="list-style-type: none"> quadrato con vertici nominati a partire dall'angolo superiore sinistro alto A,B,C,D, diagonali AC e BD <p>Caratteristiche dell'interfaccia:</p> <ul style="list-style-type: none"> I valori correnti dei controlli sono sempre visualizzati <p>...</p>	
PROMPT	<p>Corretto, ora ruotano, ma non hai rispettato il vincolo di appartenenza delle estremità dei loro estremi. Non giacciono sui lati!! Correggi.</p>	
CHATGPT	<p>Capisco, vuoi creare un effetto dove i punti si spostano gradualmente mantenendo l'intersezione al centro.</p> <p>Ho modificato il codice per implementare lo spostamento graduale dei punti medi verso il centro. Ecco come funziona:</p> <p>I segmenti passano dal centro e si muovono da AB a CD per il rosso.</p> <p>Il segmento Blu si muove da DA e da BC.</p> <p>Questo crea un effetto di transizione graduale da sinistra a destra</p> <p>...</p>	

Fig. 4a. First experimentation performed with *ChatGPT* to obtain the simplified Crossover exercise.

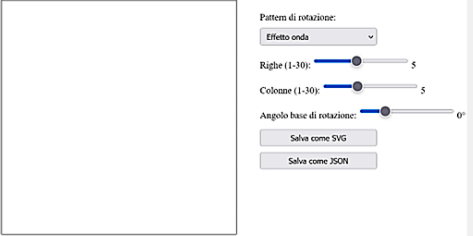
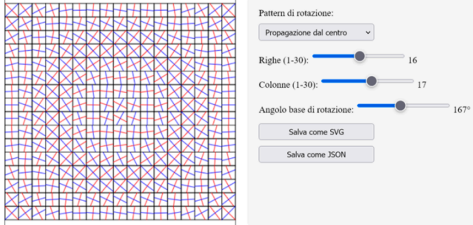
Conclusion

The experimentation highlights some fundamental differences between the approaches, methodologically and functionally. In the case of computational design, the writing of an algorithm involves a systematic method of resolution that consists of understanding and defining a problem through the abstraction of the salient aspects, which are subsequently translated into solving procedures designed to instruct the processor. The construction of the process takes place through successive speculative spaces, requiring continuous review of the different moments of development in which the designer operates independently in writing the code. With AI, communication occurs through the natural language used to describe the execution of a specific task.

Fig. 4b. First experimentation performed with *ChatGPT* to obtain the simplified Crossover exercise.

PROMPT	Ottimo. Ora dividi le diagonali al centro e crea un punto medio che divida a metà i segmenti così creati, considerando la distanza dal lato del quadrato al centro del quadrato. Devi ottenere 4 punti
CHATGPT	<p>Ho modificato le diagonali secondo le indicazioni:</p> <p>Ho diviso le diagonali nel punto dividendo a metà i segmenti così creati, ottenendo quattro punti</p> 
PROMPT	Errato! Il disegno non è cambiato. Dividi segmento rosso al centro e suddividi i due segmenti ottenuti esattamente a metà. Ripeti l'operazione per il segmento blu.
CHATGPT	<p>Ho capito! Ho modificato le diagonali secondo le indicazioni:</p> <p>Ho diviso le diagonali nel punto dividendo a metà i segmenti così creati, ottenendo quattro punti</p> <p>...</p> 
PROMPT	Errato! Il disegno non è cambiato neanche stavolta. Verifica le due precedenti istruzioni e ripeti
CHATGPT	<p>Verifico! Ho modificato le diagonali secondo le indicazioni.</p> <p>Ho diviso le diagonali nel punto dividendo a metà i segmenti così creati, ottenendo quattro punti</p> <p>...</p> <p>Vuoi che aggiunga altre funzionalità o modifichi qualcosa dell'interfaccia?</p> 
PROMPT	Errato! Si sono perse le diagonali. Lascia stare! Non seguire le ultime due istruzioni e ricomincia dall'istruzione di rotazione delle diagonali, ma ora crea all'interno del canvas con questa costruzione: una griglia di $n \times m$ dove n =righe e m =colonne (n e m con range 1:30) parametrizzate da due slider.
CHATGPT	<p>Creerò un'interfaccia per disegnare e gestire una griglia con i controlli richiesti.</p> <p>Ho creato un'interfaccia completa con le seguenti caratteristiche:</p> <p>Canvas 400x400 posizionato a sinistra</p> <p>Controlli a destra del canvas nel tab "Grid":</p> <p>Checkbox per mostrare/nascondere la griglia</p> <p>...</p> <p>Puoi modificare tutti i parametri e vedere i cambiamenti in tempo reale. Quando salvi come SVG o JSON, ti verrà chiesto il nome del file.</p> <p>Vuoi che aggiunga altre funzionalità o modifichi qualcosa dell'interfaccia?</p> 
PROMPT	Ottimo. Identifica, senza rappresentarle, delle possibili relazioni basate sulla rotazione delle diagonali nei quadrati. Ad esempio: casuale, funzione, deformazione ecc.
CHATGPT	<p>Pattern basati sulla posizione</p> <p>Rotazione proporzionale alla posizione nella griglia (riga \times colonna)</p> <p>Effetto onda: la rotazione segue una funzione sinusoidale basata sulla posizione</p> <p>Rotazione alternata tra quadrati adiacenti (scacchiera)</p> <p>Rotazione che parte dal centro e si propaga verso l'esterno.</p> <p>...</p>

Rather than an authentic dialogue with the computer, speaking of a mutation of the human-machine relationship is possible. In fact, it is not yet possible to establish an actual conversation, in the etymological sense of 'cum-versare' (*cum*=to, *versare*=to turn around, to find oneself, to entertain oneself), that is, of dialogical confrontation. Depending on the type of AI used, choosing valid terms to provide adequate input to generate consistent and quality responses is still a hybrid process, somewhere between natural language and programming code. The neural AIs used for experimentation are general-purpose, designed to simulate a –surprisingly– natural and fluid conversation by continuously reprocessing the data, adapting the communicative register to the subject matter. When used in solving a Parquet Deformations

PROMPT	Implementa il primo pattern proposto basato sulla posizione. Aggiungi una listbox che contenga tutte le soluzioni che hai elencato col fine di scegliere tra i diversi pattern che hai identificato.	
CHATGPT	<p>Creerò un'interfaccia per disegnare e gestire una griglia con i controlli richiesti. Ho creato un'interfaccia completa con le seguenti caratteristiche:</p> <p>Canvas 400x400 posizionato a sinistra Controlli a destra del canvas nel tab "Grid": Checkbox per mostrare/nascondere la griglia ...</p>	
PROMPT	Attenzione, il disegno non appare, mentre la lista di pattern è corretta. Devi visualizzare il corrispettivo grafico! Verifica e correggi!	
CHATGPT	<p>Ho capito, Ho creato un'interfaccia completa con le seguenti caratteristiche:</p> <p>Canvas 400x400 posizionato a sinistra Controlli a destra del canvas nel tab "Grid": Checkbox per mostrare/nascondere la griglia</p>	

Pattern basati sulla posizione selezionabili dalla medesima List Box

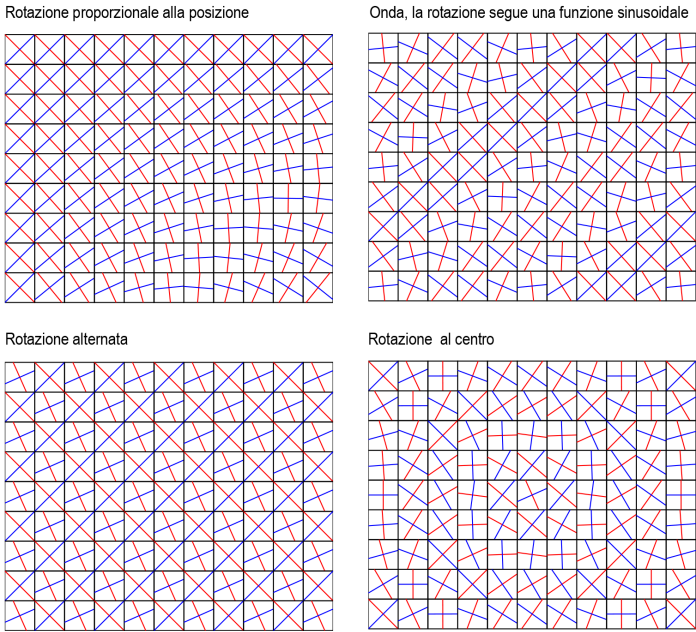


Fig. 4c. First experimentation performed with ChatGPT to obtain the simplified Crossover exercise.

exercise, they reveal some limitations, providing unsatisfactory answers or in visible contradiction to the instructions, even if they are syntactically correct. The tool reproduces the semantic relations that distinguish formal language features but has no access to meanings; they are derived from patterns of language structuring, not patterns of understanding. Hence the randomness –now fruitful and surprising, now frustrating and disappointing– of interaction. In using a text-to-image generative AI, one must also consider that writing and drawing, although both expressive media endowed with lexical and syntactic elements, are cultural technologies with different characteristics. Producing a written representation of a graphic representation, in fact, means bringing two quintessentially opposing media closer together; putting language at the service of the image. This is the role of *èkphrasis*, a rhetorical figure in which an artistic production attempts to relate itself to another medium, defining and describing the essence and form of the

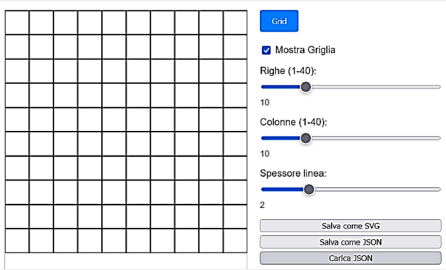
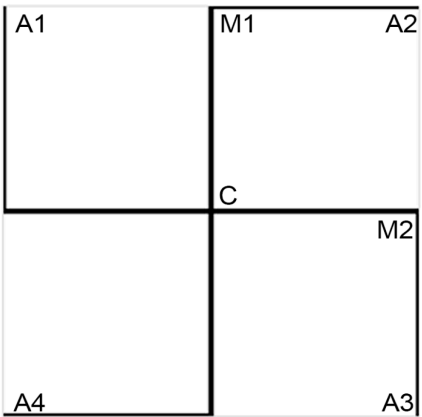
PROMPT	<p>Agisci da esperto in geometria (2D/3D), pattern deformation e parquet deformation, usa linguaggio html javascripts, genera il codice in un solo file html. Canvas 400x400 posizionato a sinistra, eventuali controlli a destra del canvas:</p> <ul style="list-style-type: none"> - tasto per salvare il disegno in formato svg in un file di cui mi chiederai il nome, - tasto per salvare il disegno in formato json in un file di cui mi chiederai il nome. - tasto per caricare il disegno in formato json <p>organizza interfaccia in tabs, successivamente ti istruirò poi su quali controlli mettere in quali tabs, devi visualizzare sempre i valori correnti dei controlli, i controlli devono essere allineati in colonna e ridisegna ad ogni cambiamento dei parametri di controllo. Seguendo queste istruzioni genera una griglia quadrata, di $n \times m$ con n e m gestite da slider con range (1 - 40). Aggiungi checkbox per attivazione e aggiungi controllo spessore linea griglia. Inserisci questi controlli nel tab1="grid"</p>
CHATGPT	<p>Creerò un'interfaccia per disegnare e gestire una griglia con i controlli richiesti. Ho creato un'interfaccia completa con le seguenti caratteristiche:</p> <ol style="list-style-type: none"> Canvas 400x400 posizionato a sinistra Controlli a destra del canvas nel tab "Grid": Checkbox per mostrare/nascondere la griglia <p>...</p> <p>Caratteristiche dell'interfaccia: I valori correnti dei controlli sono sempre visualizzati</p> <p>...</p> <p>Puoi modificare tutti i parametri e vedere i cambiamenti in tempo reale. Quando salvi come SVG o JSON, ti verrà chiesto il nome del file. Vuoi che aggiunga altre funzionalità o modifichi qualcosa dell'interfaccia?</p> 
PROMPT	<p>Attenzione, Il prototipo da deformare coincide con la griglia. Per la deformazione segui le istruzioni successive che genereranno la tassellazione finale: siano A1,A2,A3,A4, i vertici della cella con A1 in alto a sinistra e i vertici in senso orario. M1,M2,M3, M4 i saranno i punti medi dei lati A1-A2, A2-A3, A3-A4, A4-A1. C sarà il centro della cella. Dati questi dati costruisci le seguenti geometrie: 1: A1-M4-C-M2-A3 e 2: A4-M3-C-M1-A2. Aggiungi checkbox per visualizzare le 2 geometrie e Aggiungi controllo spessore.</p>
CHATGPT	<p>Ho aggiunto le seguenti funzionalità: Due nuove geometrie diagonali per ogni cella: Diagonale 1: A1-M4-C-M2-A3 (dall'angolo in alto a sinistra all'angolo in basso a destra) Diagonale 2: A4-M3-C-M1-A2 (dall'angolo in basso a sinistra all'angolo in alto a destra) Nuovi controlli nel tab "Grid": Checkbox per mostrare/nascondere la Diagonale 1 Checkbox per mostrare/nascondere la Diagonale 2 Slider per controllare lo spessore delle linee diagonali (indipendente dallo spessore della griglia) Le geometrie vengono disegnate in ogni cella della griglia e si aggiornano automaticamente quando: Modifichi il numero di righe o colonne Attivi/disattivi le singole diagonali Modifichi lo spessore delle linee diagonali Vuoi che aggiunga altre funzionalità o modifichi qualcosa dell'interfaccia?</p> 
PROMPT	<p>Corretto. Per la prima cella, posizionata sulla prima riga e sulla prima colonna, angolo in alto a sinistra, sia $L = (A1-A2)/2$. Per ogni riga successiva, a partire dalla seconda cella, i punti M1,M2,M3,M4 si muovano verso C di una quantità pari a $Q = L/(m-1)$ dove m è il numero di colonne.</p>

Fig. 5a. Second experimentation performed with ChatGPT to obtain the simplified Crossover exercise.

original artefact. A descriptive work of prose or poetry can thus highlight, with their own codes, the meaning or particularities of a work of visual art, expanding the user's aesthetic experience. Writing is based on linear logic, facilitating its translation into computational terms. Messages are expressed sequentially, by decoding simplified signs –the letters– governed by codes shared by specific groups of recipients –the languages.

On the other hand, drawing follows a connectionist logic, which admits the synchrony of several messages, expressed simultaneously through iconic signs. Decoding rules are less specialised, favouring a pragmatic and operational cross-cultural dimension that diversifies into many repertoires and genres.

The correspondence of evolutionary stages between the ontogeny of child drawing and the phylogeny of prehistoric art shows how the mental representation necessary

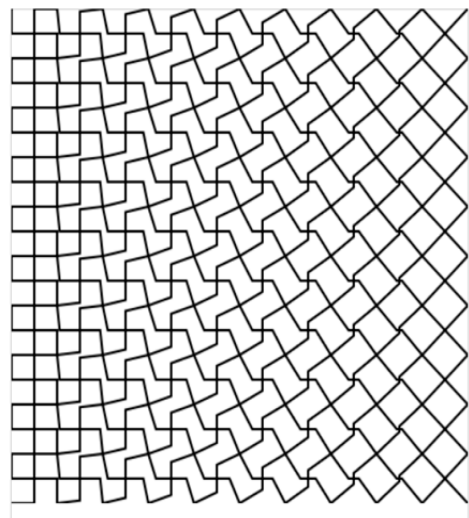
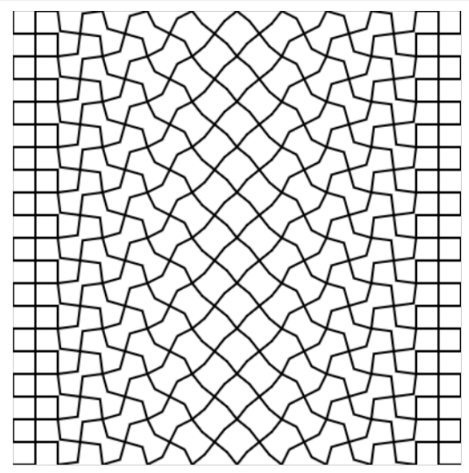
CHATGPT	<p>Capisco, vuoi creare un effetto dove i punti medi si spostano gradualmente verso il centro da sinistra a destra, con uno spostamento proporzionale alla posizione della colonna. Ho modificato il codice per implementare lo spostamento graduale dei punti medi verso il centro.</p> <p>Ecco come funziona:</p> <ol style="list-style-type: none"> 1. Per la prima colonna ($j=0$), i punti medi M1, M2, M3, M4 rimangono nelle loro posizioni originali 2. Per ogni colonna successiva: <ul style="list-style-type: none"> Calcolo L come metà della larghezza della cella ($dx/2$) Calcolo Q come $L/(m-1) * j$, dove j è l'indice della colonna Sposto ogni punto medio M verso il centro C di una quantità proporzionale a Q <p>Questo crea un effetto di transizione graduale da sinistra a destra, dove:</p> <ul style="list-style-type: none"> La prima colonna mantiene i punti medi originali L'ultima colonna avrà i punti medi più vicini al centro Le colonne intermedie avranno spostamenti proporzionali alla loro posizione <p>Vuoi che aggiunga dei controlli per modificare questo effetto o che faccia altre modifiche?</p>	
PROMPT	<p>Proponi con il codice dato possibili variazioni, per esempio spostando i punti non più verso il centro ma verso i medi adiacenti</p>	
CHATGPT	<p>Ho modificato il codice per creare un diverso spostamento dei punti medi, non più verso il centro. Ti aiuterò a modificare il codice per spostare i punti medi verso i punti medi adiacenti invece che verso il centro. Aggiorniamo il codice per creare questo nuovo schema.</p> <p>Ho modificato il codice per implementare il nuovo modello di movimento che avete descritto. Ora:</p> <ul style="list-style-type: none"> M1 si muove verso M2 M2 si muove verso M3 M3 si muove verso M4 M4 si muove verso M1 <p>La quantità di movimento è controllata da Q, che è calcolato come $j/(m-1)$, dove:</p> <ul style="list-style-type: none"> j è il numero di colonna corrente (da 0 a m-1) m è il numero totale di colonne <p>In questo modo si crea un modello diverso in cui ogni punto centrale si sposta verso il suo vicino in modo ciclico, con uno spostamento crescente da sinistra a destra attraverso la griglia.</p> <p>Ti piacerebbe vedere altre modifiche?</p>	

Fig. 5b Second experimentation performed with ChatGPT to obtain the simplified Crossover exercise.

for drawing precedes writing, which is more linked to linear and less intuitive logical processes [Antonietti et al. 1995, p. 13]. Therefore, expressing through the written verbal code elements that, with a simple sign, can be effectively interpreted, could be problematic in expressing emotional complexities and understanding logical-mathematical problems. Indeed, the ekphrastic process requires a cognitive process that, starting from a literary composition, enables the formation of the mental image of an artefact. For this purpose, the ekphrastic technique therefore involves vivid, often elaborate styles to evoke the same feeling as the original work of art. Despite some affinities with specific processes of the human mind, such as logical reasoning, symbol processing and arithmetic operations, AI devices remain inherently distant from this dimension. Therefore, we are forced to instruct the machine with text that reflects the processes in which learned data and patterns are organised within the generative AI system if we are to generate results that best reflect the user's intentions. It isn't easy to find common

ground with neural networks in light of the subjective experiences we have as beings endowed with awareness, a fundamental ingredient of our imaginative and creative faculties [Chalmers 2017, pp. 32-42]. Without any clue as to how to narrow the ontological gap between a human being and a computer, the only perspective we can take in considering generating a visual output from a textual description by AI is to admit that it does not fall within the framework of traditional *ekphrasis*.

In the design process, drawing conceptually unifies a considerable field of knowledge by establishing visual, systemic, or geometric relationships between the parts of a project. Computational Design uses known entities –geometric parameters and mathematical relationships–, leading to unknown outputs. The several design instances are articulated in emergent relational structures that cannot be controlled by predicting the countless possible outcomes. Only by defining the behaviour of the basic entities and relationships and leaving it to the computer's computational power to simulate the resulting collective effect can the design solution's validity be verified.

Using Neural Networks, not only is the outcome not predictable *a priori*, but the resolution process is also unknowable. Even when the output meets expectations, there is no method for understanding the role or influence of the hundreds of neural nodes on the processing. Unlike Computational Design, where the rules are explicit and visible, the results and operations performed by AIs are not explainable even by the developers themselves. As a result, while in the computational approach it is the process that determines the correctness of the solution, with AIs the lack of understanding of the generation processes exposes the real risk of latent errors. While in image generation this opacity may not undermine the tool's usefulness, in the case of more structured graphical exercises such as Parquet Deformations, a prelude to the design path, the lack of control results in failure to achieve the objectives.

AIs are not a transient phenomenon and are subject to rapid evolution. It cannot be ruled out that new versions and more outstanding computational capabilities will soon overcome many of the encountered limitations. Nonetheless, drawing for design is not only based on technical and discrete parameters but also involves a deep understanding of cultural and social dynamics and the ability to interpret and predict the characteristics and quality of form and space, attributes that are difficult to encode through databases or algorithms. In design, the formulation of countless alternatives is applicable only if one has the appropriate critical and cultural tools to choose appropriate solutions. It is essential to invest in education and open up new teaching practices that prepare designers to acquire skills on how AI works so that they can use it to their advantage and avoid misuse.

Attributions

This text is the result of shared work. However, G.B. was responsible for drafting the whole text, drafting and testing the algorithms, and conducting some of the experimentation using AIs. At the same time, G.V. was responsible for part of the experimentation with AIs and the revision.

Notes

[1] The Hochschule für Gestaltung (HfG) was founded in Ulm in 1953 by Inge Aicher-Scholl, Otl Aicher and Max Bill, the school's first dean. Bill, a Bauhaus student since 1927, is an advocate of Gropius' aesthetic-formal orientation centred on *Lute Form* (good form). However, neither the methods nor the metrics by which to develop it are clear; so in 1956, Maldonado, spokesperson for a group of young faculty members, proposed a new design methodology based on the scientific method and aimed at creating technologically advanced and socially useful products.

[2] Visual programming, or visual scripting, is a method that uses icons identifying parts of a programming code that allow the user to form a network of symbols equivalent to a complete, executable script. It is thus possible to write complex and articulate programs even for users with no special programming language skills.

[3] Spline curves are generated by a function consisting of a set of connected polynomials, whose purpose is to interpolate in an interval a set of points called nodes such that the function is continuous at least up to a given order of derivatives at each point in the interval. Therefore, the generated curve involves interpolating points so that there are no abrupt changes in slope or cusp points, maintaining a continuous transition.

[4] Although he built the first neural network, Minsky devoted the rest of his life as a researcher to demolishing the theories supporting Connectionist AIs, beginning with his famous critique of F. Rosenblatt's Perceptron (1958), a rudimentary electronic device modelled on the physiology of the human brain's neural networks. Indeed, he was a staunch proponent

of symbolic AI, based on production rules that link symbols in an If-Then instruction to infer and determine any additional information needed. The crushing of Rosenblatt's work, coupled with the poor results obtained by symbolic AI researchers over a decade, caused a sudden halt to all studies of neural networks, resulting in what has gone down in history as the 'winter of artificial intelligence'.

[5] The term 'deep' refers to the presence of a number of layers that make up the neural network. The layers are composed of nodes that receive input and produce output until an end result is reached. The more layers of nodes there are, the higher the performance of the network. Early neural networks were composed of two or three layers, while today's deep learning networks can have up to 200 layers.

[6] A generative adversarial network is a class of AI in which two neural networks are trained competitively in the context of a zero-sum game. Its original version consists of two components: a generative network, which produces data from a database, and a discriminative network, which selects the generated data by comparing it with a database of optimized real data and establishing its acceptability.

Reference List

- Antonietti, A., Angelini, C., Cerena, P. (1995). *L'intuizione visiva. Utilizzare le immagini per analizzare e risolvere i problemi*. Milano: FrancoAngeli.
- Buratti, G. (2023). Disegno in transizione e transizione nel disegno. Passato e futuro degli esercizi di Parquet Deformations/The Drawing Transition and Transition in the Drawing. Past and Future of Parquet Deformations Exercises. In M. Cannella, A. Garozzo, S. Morena (Eds.). *Transizioni / Transitions*. Atti del 44° Convegno Internazionale dei Docenti delle Discipline della Rappresentazione. Palermo, 14-16 settembre 2023. Milano: FrancoAngeli, pp. 122-138.
- Buratti, G., Rossi, M. (2024). Is a Picture Worth a Thousand Words? Comparative Evaluation of Generative AI for Drawing and Representation. In A. Giordano, M. Russo, R. Spallone (Eds.). *Advances in Representation. Digital Innovations in Architecture, Engineering and Construction*. Cham: Springer, pp. 867-884. https://doi.org/10.1007/978-3-031-62963-1_53.
- Carpo, M. (2024). Perspectives in computational design: A brief assessment of today's socio-technical context, promises, and challenges. In *Perspectives in Architecture and Urbanism*, Vol. 1, Issue 1, <https://doi.org/10.1016/j.pau.2024.100001>.
- Chalmers, D. (2017). The hard problem of consciousness. In M. Velmans, S. Schneider (Eds.). *The Blackwell companion to consciousness*. Hoboken: Wiley, pp. 32-42. <https://doi.org/10.1002/9781119132363.ch3>.
- Minsky, M. (1961) Steps toward Artificial Intelligence, In *Proceedings of the IRE*, vol. 49, no. 1, pp. 8-30. doi: 10.1109/JR-PROC.1961.287775.
- Turing, A. M. (1937). On Computable Numbers, with an Application to the Entscheidungsproblem. In *Proceedings of the London Mathematical Society*, Series 2, 42, Issue 1, pp. 230-265. <https://doi.org/10.1112/plms/s2-42.1.230>.
- Von Neumann, J. (1958). *The Computer and the Brain*. New Haven: Yale University Press.

Authors

Giorgio Buratti, Polytechnic of Milan, giorgio.buratti@polimi.it
Giorgio Vignati, Independent Scholar, gvignati@gmail.com

To cite this chapter: Giorgio Buratti, Giorgio Vignati (2025). Parquet Deformations, Computational Design and AI. Graphical Exercises from the Past for New Paradigms. In L. Carlevaris et al. (Eds.). *èkphrasis. Descrizioni nello spazio della rappresentazione/èkphrasis. Descriptions in the space of representation*. Proceedings of the 46th International Conference of Representation Disciplines Teachers. Milano: FrancoAngeli, pp. 3569-3592. DOI: 10.3280/oa-1430-c940.