

Arte generativa computacional: exploração de arte abstrata, geométrica e florestas de grafos com python

Vitor Amadeu Souza¹; 0009-0002-1857-6799

1 – UniFOA, Centro Universitário de Volta Redonda, Volta Redonda, RJ.
vitor.amadeu@foa.org.br

Resumo: Este artigo apresenta uma metodologia para criação de arte generativa computacional utilizando a linguagem de programação Python e suas bibliotecas especializadas em visualização de dados e teoria dos grafos. O estudo explora três paradigmas distintos de arte digital: arte abstrata baseada em linhas aleatórias, arte geométrica através de polígonos regulares e florestas de grafos como representação visual de estruturas matemáticas complexas. A pesquisa demonstra como algoritmos de geração aleatória podem ser empregados para criar composições visuais únicas, combinando elementos matemáticos com princípios estéticos. Os resultados obtidos evidenciam que a programação computacional pode servir como ferramenta criativa legítima para produção artística, oferecendo possibilidades de expressão através da manipulação paramétrica de elementos visuais. A metodologia proposta utiliza as bibliotecas Matplotlib, NumPy e NetworkX para gerar três categorias distintas de arte visual, cada uma explorando diferentes aspectos da visualização computacional. Os experimentos realizados confirmam a viabilidade da abordagem algorítmica para criação artística, demonstrando que a aleatoriedade controlada pode produzir resultados esteticamente interessantes e matematicamente fundamentados.

Palavras-chave: Arte generativa. Programação criativa. Python. Visualização de dados. Algoritmos artísticos.

INTRODUÇÃO

A arte generativa representa uma das mais fascinantes intersecções entre tecnologia e criatividade, emergindo como um campo de estudo que combina princípios matemáticos, programação computacional e sensibilidade estética (Galanter, 2003). Esta forma de expressão artística utiliza sistemas autônomos ou semi-autônomos para criar obras que seriam impossíveis ou extremamente difíceis de produzir manualmente, estabelecendo novos paradigmas para a criação e apreciação artística no século XXI (Boden & Edmonds, 2009). A evolução da arte generativa está intrinsecamente ligada ao desenvolvimento da computação, tendo suas raízes nos trabalhos pioneiros de artistas como Frieder Nake, Georg Nees e Michael Noll na década de 1960, que utilizaram computadores mainframe para criar as primeiras obras de arte algorítmica (Taylor, 2014).

A linguagem de programação Python emergiu como uma ferramenta fundamental para artistas digitais e pesquisadores interessados em arte generativa devido à sua sintaxe intuitiva, vasta biblioteca de recursos e capacidade de integração com diferentes domínios do conhecimento (McKinney, 2012). As bibliotecas especializadas em visualização de dados, como Matplotlib, e em análise de redes, como NetworkX, oferecem recursos sofisticados que permitem a criação de composições visuais complexas através de código computacional (Hunter, 2007; Hagberg *et al.*, 2008). Esta convergência tecnológica possibilita que artistas e pesquisadores explorem novos territórios criativos, onde a programação se torna um meio de expressão artística tão válido quanto o pincel ou o cinzel.

O presente estudo investiga três abordagens distintas para criação de arte generativa utilizando Python: arte abstrata baseada em linhas e trajetórias aleatórias, arte geométrica construída através de polígonos regulares e visualização de florestas de grafos como manifestação artística de estruturas matemáticas. Cada uma dessas abordagens representa diferentes aspectos da criação algorítmica, desde a aleatoriedade pura até a representação visual de conceitos matemáticos complexos (Pearson, 2011). A escolha dessas três modalidades permite uma exploração abrangente das possibilidades oferecidas pela programação criativa, demonstrando como diferentes algoritmos podem produzir resultados visuais distintos e esteticamente relevantes.

MÉTODOS

O primeiro paradigma implementado, denominado Arte Abstrata, utiliza algoritmos de geração de trajetórias aleatórias para criar composições lineares complexas. O algoritmo gera N elementos (onde $N=30$ para todos os experimentos), cada um consistindo em uma sequência de 10 pontos com coordenadas x e y distribuídas aleatoriamente no intervalo $[0,1]$ utilizando a função `numpy.random.rand()`. Estas sequências de pontos são conectadas por linhas cujas propriedades visuais (cor e espessura) são determinadas aleatoriamente para cada elemento. As cores são definidas através de vetores RGB tridimensionais com valores aleatórios no intervalo $[0,1]$, enquanto a espessura das linhas varia uniformemente entre 0.5 e 2.0 unidades utilizando a função `random.uniform()` (Van Rossum & Drake, 2009).

A segunda modalidade, Arte Geométrica, emprega algoritmos de geração de polígonos regulares com parâmetros aleatórios para criar composições baseadas em formas geométricas primitivas. Cada elemento é definido por um polígono regular com número de lados variando aleatoriamente entre 3 e 8, raio entre 0.1 e 0.5 unidades, e posição central aleatória no plano cartesiano. A construção matemática dos polígonos utiliza coordenadas polares convertidas para coordenadas cartesianas através das funções trigonométricas seno e cosseno, implementadas através das funções `numpy.cos()` e `numpy.sin()`. O preenchimento dos polígonos utiliza cores RGB aleatórias com transparência (alpha) fixada em 0.5 para permitir sobreposições visuais interessantes entre elementos adjacentes (Bourke, 1997).

O terceiro paradigma, Floresta de Grafos, implementa visualizações de múltiplas árvores aleatórias geradas através do algoritmo de Prüfer para criação de árvores rotuladas uniformemente distribuídas. Cada árvore possui entre 3 e 10 vértices, determinados aleatoriamente, e é posicionada no espaço através do algoritmo `spring layout`, que simula um sistema físico de molas para determinar posições otimizadas dos nós (Fruchterman & Reingold, 1991). A função `nx.random_labeled_tree()` da biblioteca `NetworkX` implementa o algoritmo de Cayley para geração uniforme de árvores rotuladas, garantindo que todas as possíveis configurações de árvore com n vértices tenham probabilidade igual de serem selecionadas (Kelly, 1957).

A validação metodológica baseou-se na execução de múltiplas iterações do algoritmo para verificar a consistência da geração aleatória e a diversidade visual dos resultados produzidos. Cada execução do programa gera uma composição única devido à natureza estocástica dos algoritmos empregados, demonstrando a capacidade do sistema de produzir variações infinitas a partir de um conjunto finito de regras algorítmicas. A reprodutibilidade científica é garantida através da documentação completa dos parâmetros utilizados e da disponibilização do código fonte implementado, cujo código-fonte está disponível para download através do link: https://github.com/vitor-souza-ime/art_agf.

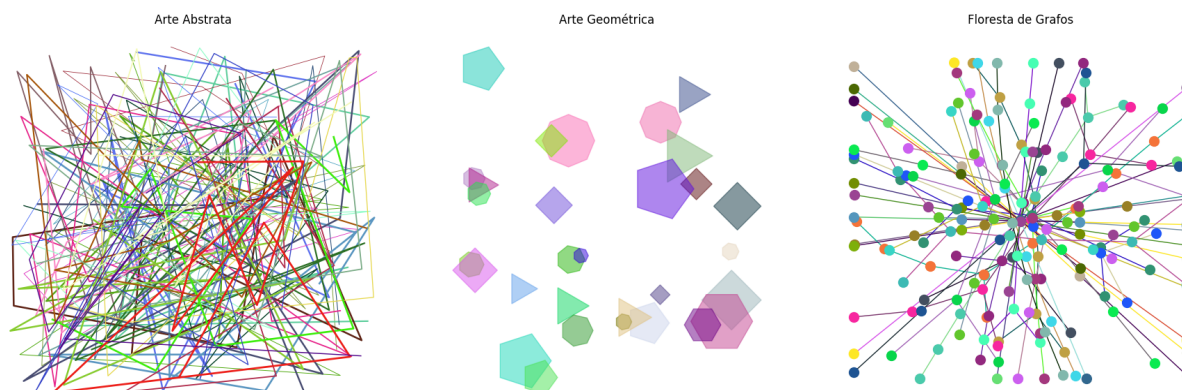
RESULTADOS E DISCUSSÃO

A implementação dos algoritmos de arte generativa produziu resultados visuais que demonstram claramente as distintas características de cada paradigma artístico investigado. A análise dos resultados revela padrões emergentes que surgem da interação entre aleatoriedade controlada e estrutura algorítmica, confirmando a hipótese de que sistemas computacionais podem gerar complexidade visual esteticamente relevante através de regras relativamente simples (Mitchell, 2009). A composição final apresenta três modalidades artísticas claramente diferenciadas, cada uma explorando aspectos únicos da visualização computacional e demonstrando as vastas possibilidades oferecidas pela programação criativa, como pode ser apreciado na Figura 1.

A Arte Abstrata, primeiro paradigma implementado, gerou composições caracterizadas por complexidade visual emergente resultante da sobreposição de múltiplas trajetórias lineares aleatórias. A análise visual revela a formação de regiões de alta densidade onde múltiplas linhas se interceptam, criando pontos focais que direcionam a atenção do observador e estabelecem hierarquias visuais não planejadas explicitamente no algoritmo (Alexander, 2002). A variação aleatória de cores produz paletas cromáticas únicas a cada execução, demonstrando como a aleatoriedade pode servir como ferramenta de exploração do espaço de possibilidades estéticas. A espessura variável das linhas contribui para a percepção de profundidade e movimento, criando uma sensação de dinamismo que ecoa as obras do expressionismo abstrato americano, particularmente os trabalhos de Jackson Pollock, que também utilizava processos semi-aleatórios em sua criação artística (Hunter *et al.*, 1956).



Figura 1 - Arte abstrata, geométrica e floresta de grafos



Fonte: O autor.

A Arte Geométrica apresentou características fundamentalmente diferentes, organizando-se através de formas primitivas claramente definidas que mantêm sua identidade individual mesmo quando sobrepostas. A utilização de transparência ($\alpha = 0.5$) permitiu a criação de efeitos de cor aditiva onde polígonos sobrepostos geram novas tonalidades através de mistura ótica, fenômeno que ecoa técnicas utilizadas por movimentos artísticos como o neoimpressionismo de Georges Seurat e Paul Signac (Brettell, 1999). A distribuição aleatória de formas geométricas no espaço cartesiano criou composições que remetem simultaneamente aos trabalhos de Piet Mondrian e Wassily Kandinsky, combinando precisão geométrica com distribuição aparentemente casual (Düchting, 2000).

A Floresta de Grafos representou o paradigma mais inovador, transformando estruturas matemáticas abstratas em composições visuais que revelam a beleza inerente à teoria dos grafos. A utilização do algoritmo de posicionamento de Fruchterman-Reingold produziu configurações espaciais onde a estrutura topológica de cada árvore é claramente visível, permitindo que observadores sem conhecimento matemático formal apreciem a elegância das estruturas arbóreas (Di Battista *et al.*, 1998). A sobreposição de múltiplas árvores com diferentes tamanhos e configurações cria uma "floresta" visual onde cada elemento mantém sua identidade estrutural enquanto contribui para uma composição coletiva mais complexa.

A comparação entre os três paradigmas revela características complementares que demonstram a versatilidade da abordagem algorítmica para criação artística. Enquanto a

Arte Abstrata explora a expressividade das linhas e a complexidade emergente, a Arte Geométrica investiga as possibilidades compositivas das formas primitivas, e a Floresta de Grafos revela a dimensão estética de estruturas matemáticas abstratas. Esta diversidade confirma que a programação criativa não se limita a um único estilo ou abordagem artística, mas oferece um meio capaz de explorar múltiplas tradições e inventar novas modalidades expressivas (Kay & Goldberg, 1977).

CONCLUSÕES

A análise dos resultados revela que a aleatoriedade controlada desempenha um papel fundamental na geração de complexidade visual interessante, funcionando como catalisador criativo que permite a emergência de padrões e estruturas não antecipados explicitamente pelos algoritmos implementados. Esta descoberta tem implicações importantes para compreensão dos processos criativos em geral, sugerindo que a criatividade pode ser conceptualizada não apenas como produto da intencionalidade consciente, mas também como resultado da interação dinâmica entre estrutura e acaso, controle e espontaneidade. O sucesso da abordagem algorítmica em produzir diversidade visual a partir de regras finitas demonstra o potencial transformador da programação criativa como meio de expressão artística.

A investigação realizada estabelece fundamentos para futuras pesquisas em arte generativa computacional, oferecendo uma metodologia testada e resultados documentados que podem servir como ponto de partida para estudos mais ambiciosos. A combinação de múltiplos paradigmas artísticos em uma única composição sugere possibilidades interessantes para exploração de hibridizações mais complexas, potencialmente incorporando elementos de inteligência artificial, processamento de imagem em tempo real, e interatividade com audiências.

REFERÊNCIAS

ALEXANDER, Christopher. The phenomenon of life: an essay on the art of building and the nature of the universe. Berkeley, Calif. Center for Environmental Structure, 2002.



BODEN, M. A.; EDMONDS, E. A. What is generative art? *Digital Creativity*, v. 20, n. 1-2, p. 21-46, 2009. Disponível em: <https://doi.org/10.1080/14626260902867915>. Acesso em: 15 ago. 2025.

BOURKE, P. Polygons, meshes. *Geometry*, 1997. Disponível em: <http://paulbourke.net/geometry/polygonmesh/>. Acesso em: 15 ago. 2025.

KELLY, Paul J. A congruence theorem for trees. 1957.

DI BATTISTA, G. et al. *Graph Drawing: Algorithms for the Visualization of Graphs*. Upper Saddle River: Prentice Hall, 1998.

DÜCHTING, Hajo; KANDINSKY, Wassily. *Wassily Kandinsky, 1866-1944: A revolution in painting*. Taschen, 2000.

FRUCHTERMAN, T. M.; REINGOLD, E. M. Graph drawing by force-directed placement. *Software: Practice and Experience*, v. 21, n. 11, p. 1129-1164, 1991. Disponível em: <https://doi.org/10.1002/spe.4380211102>. Acesso em: 15 ago. 2025.

GALANTER, Philip. What is generative art? Complexity theory as a context for art theory. In: *GA2003–6th Generative Art Conference*. 2003.

HAGBERG, Aric; SWART, Pieter J.; SCHULT, Daniel A. *Exploring network structure, dynamics, and function using NetworkX*. Los Alamos National Laboratory (LANL), Los Alamos, NM (United States), 2008.

BRETTELL, Richard R. Neo-Impressionism and the Search for Solid Ground: Art, Science, and Anarchism in Fin-de-siecle France/Pissarro, Neo-Impressionism, and the Spaces of the Avant-Garde. *The Art Bulletin*, v. 81, n. 1, p. 168, 1999.

HUNTER, J. D. Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, v. 9, n. 3, p. 90-95, 2007. Disponível em: <https://doi.org/10.1109/MCSE.2007.55>. Acesso em: 15 ago. 2025.

KAY, A.; GOLDBERG, A. Personal dynamic media. *Computer*, v. 10, n. 3, p. 31-41, 1977. Disponível em: <https://doi.org/10.1109/C-M.1977.217672>. Acesso em: 15 ago. 2025.

MCKINNEY, Wes. *Python for data analysis: Data wrangling with Pandas, NumPy, and IPython*. " O'Reilly Media, Inc.", 2012.

MITCHELL, M. *Complexity: A Guided Tour*. Oxford: Oxford University Press, 2009. Disponível em: <https://doi.org/10.1093/oso/9780195124415.001.0001>. Acesso em: 15 ago. 2025.

PEARSON, Matt. Generative art: a practical guide using processing. Simon and Schuster, 2011.

TAYLOR, Grant D. When the machine made art. 2014.

VAN ROSSUM, G.; DRAKE, F. L. Python 3 Reference Manual. Scotts Valley: CreateSpace, 2009. Disponível em: <https://docs.python.org/3/reference/>. Acesso em: 15 ago. 2025.

HUNTER, Sam; POLLOCK, Jackson; KARPEL, Bernard. Jackson Pollock. The Bulletin of the Museum of Modern Art, v. 24, n. 2, p. 3-36, 1956.