

UALMA

Unit 2

GCD

CHUCHU

Positions through iterating

RISO



PS digital copy



LASER



Original



I chose the iteration project. The project was initially aimed at reproducing Riso printing, but in practice, I tried to replace Riso printing with laser printing. Through layering, halftone image overprinting, and multiple printing processes, simulated the color and misalignment effects of Riso printing.

During this process, I noticed that laser printing also exhibits uncontrollable phenomena such as misalignment, ghosting, and color variations. My research gradually shifted towards studying these results, tiny "errors," focusing on the changes in images during technology transitions.

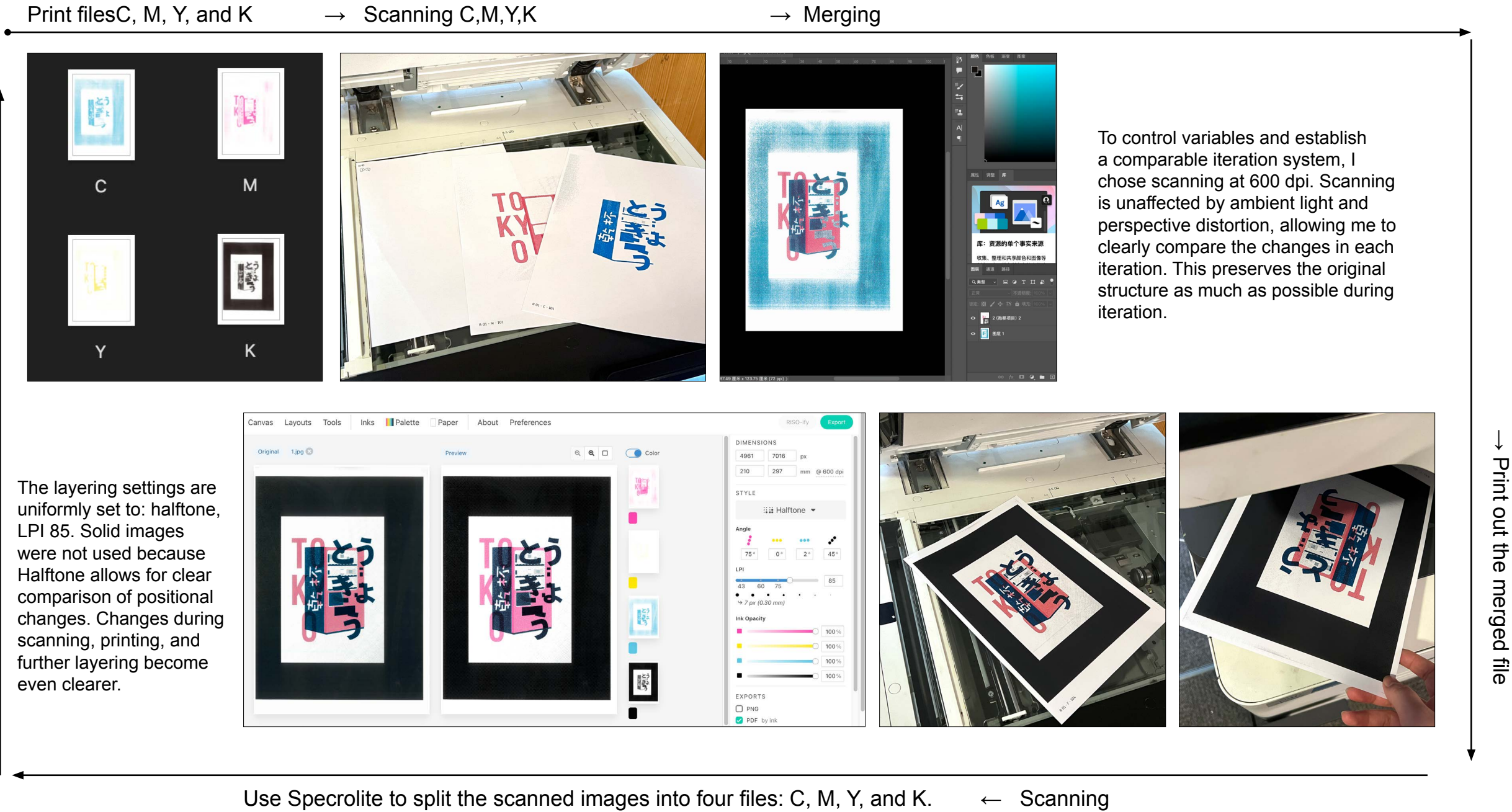
This 100-iteration project is based on this research. I repeatedly replicated the previous generation's image using the method that best reproduces the original image, amplifying the "errors" in the process through numerous iterations.

Research Questions

When replicating an image, how is information gradually shifted, lost, or reconstructed?

How does image gradually become distorted? Is this distortion systematic? How are errors continuously produced instead of being eliminated? At what stage was the error created? At what stage was it amplified?

My iterative method



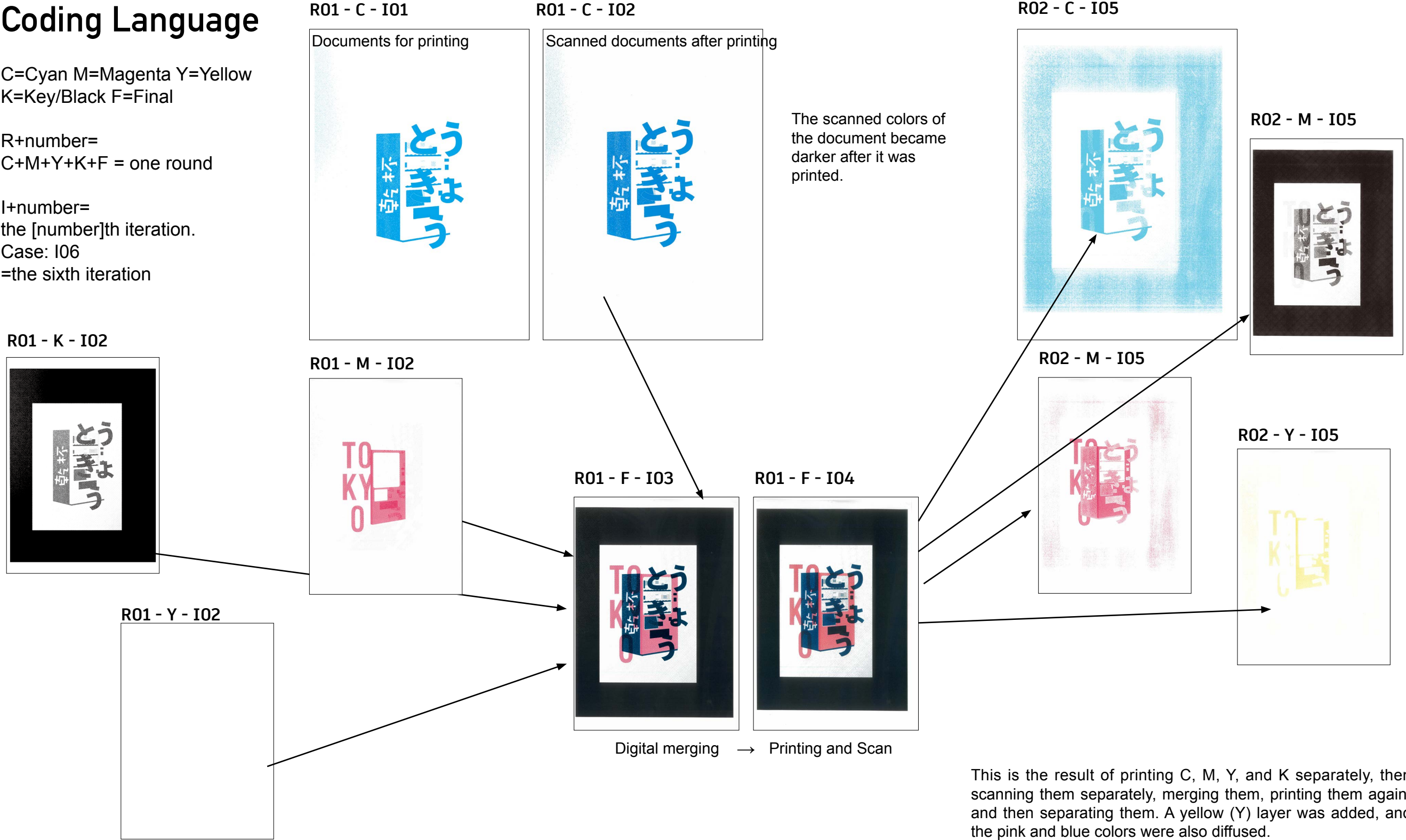
uses this cycle to transform the errors into understandable, systematic phenomena.

Coding Language

C=Cyan M=Magenta Y=Yellow
K=Key/Black F=Final

R+number=
C+M+Y+K+F = one round

I+number=
the [number]th iteration.
Case: I06
=the sixth iteration

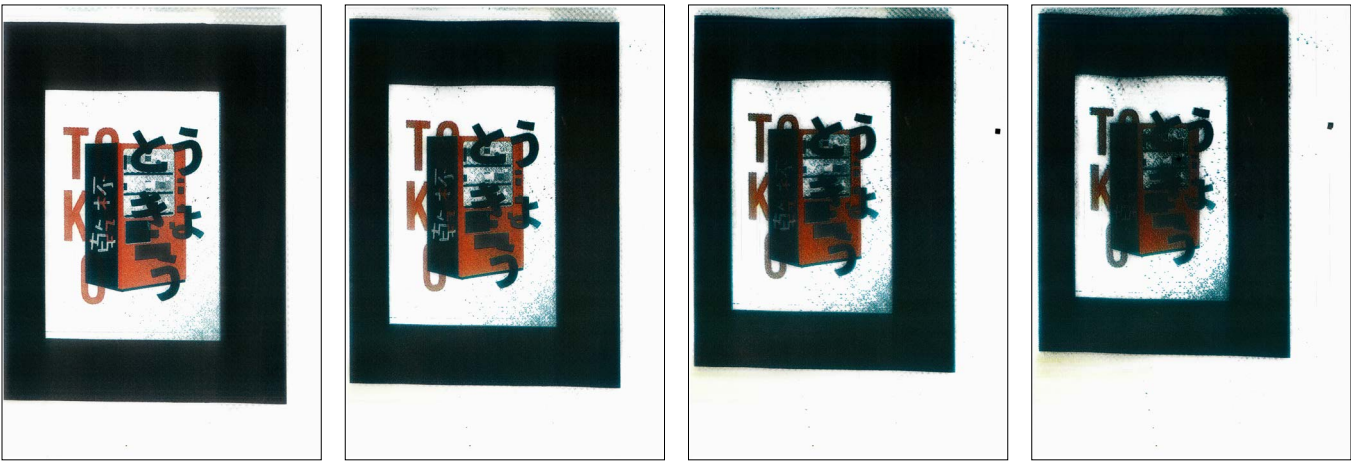


Iterative comparison of a single stage



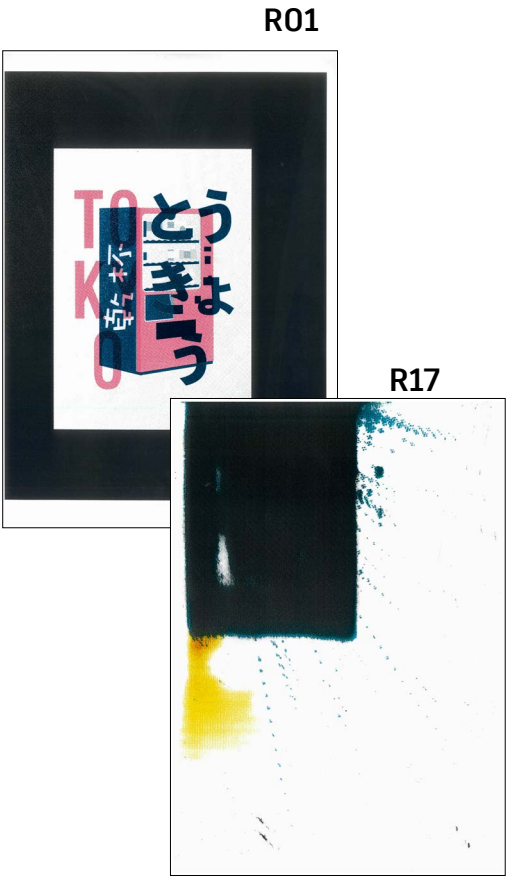
R01 → R02

In the first six iterations, the image gradually shifted from clear layering to overlapping and contamination, and color relationships and structural boundaries began to lose their independence. From R01 to R02, the CMYK layers were still clearly discernible, with misalignment and color variations, but the image structure remained intact.



R03 → R06

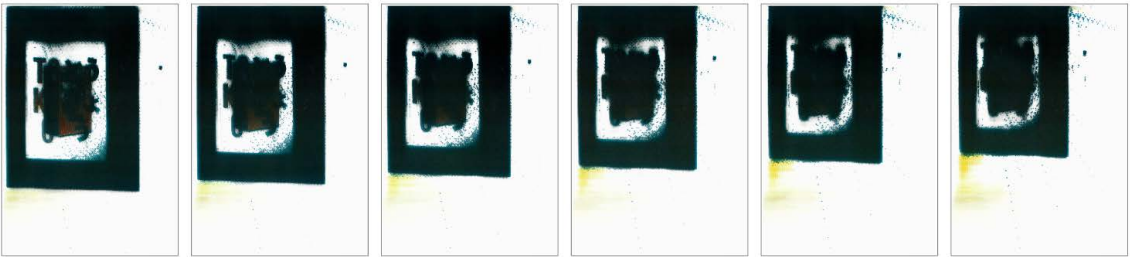
Errors begin to accumulate from R03 to R06. Printing noise is amplified during merging and layering, and the original colors are recalculated. This causes colors to gradually deepen, with black (K) gradually dominating the image. Small dots are lost or merged during scanning, and the image gradually shifts from "dots" to "blocks."



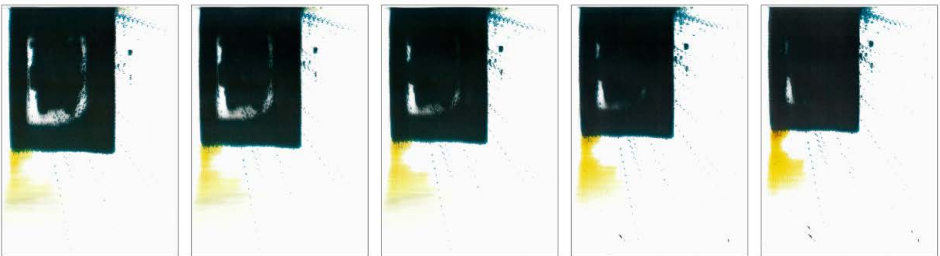
R01

R17

R07 → R12

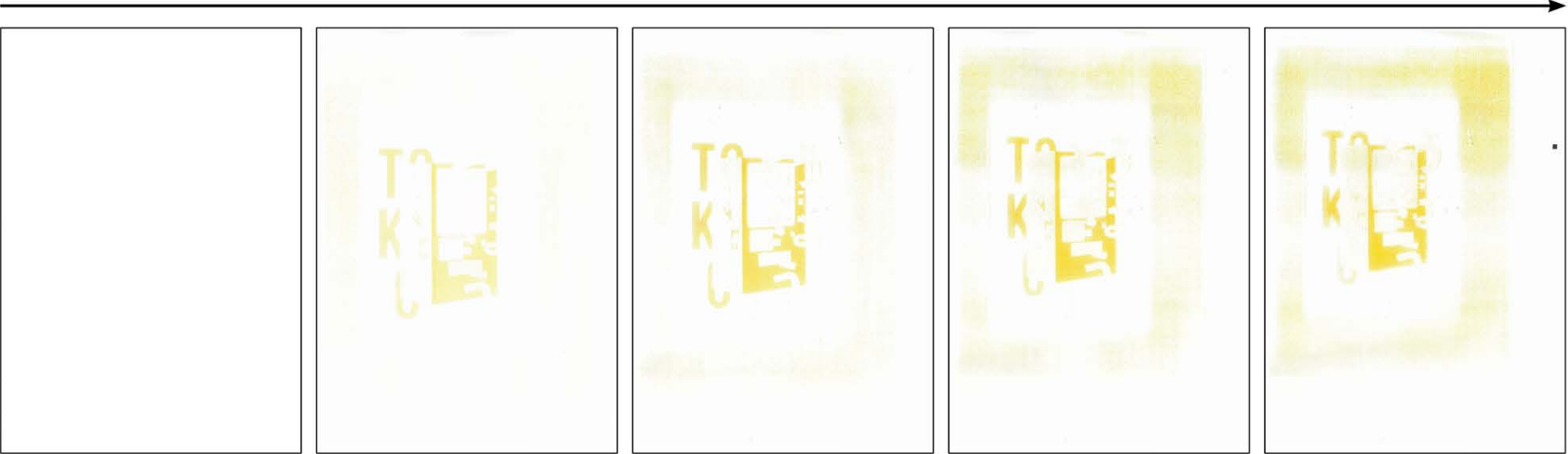


R13 → R17

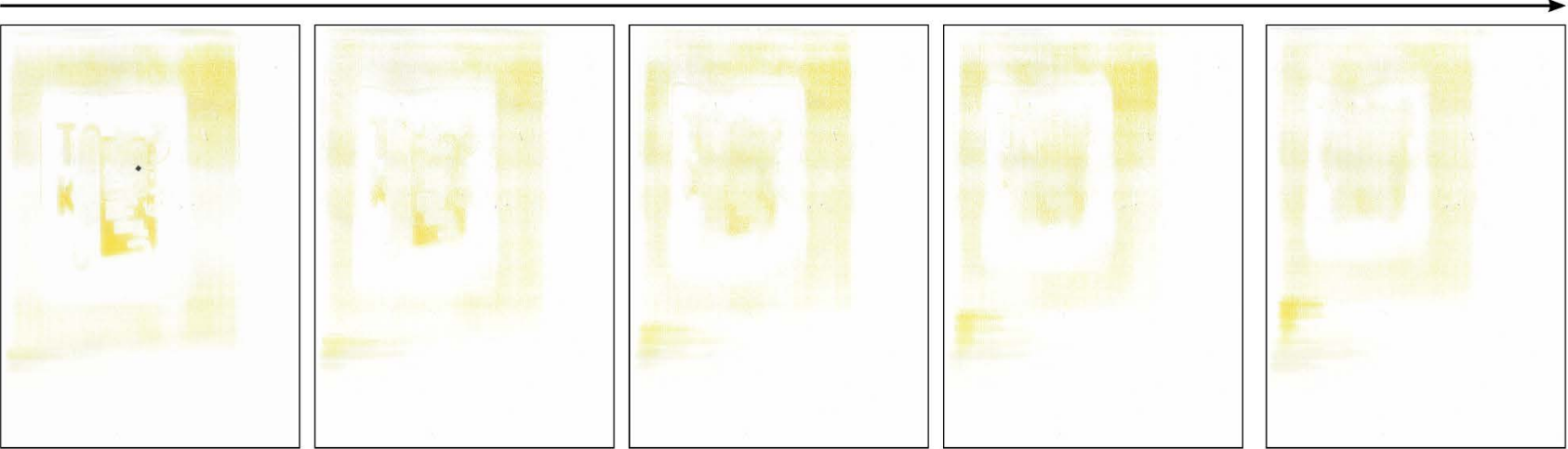


In the iterations from R07 to R017, the image shifted from local structural changes to overall image density changes. The original content was continuously compressed. Irregular color diffusion began to appear at the edges of white areas. The image edges became increasingly unstable. The focus of image change shifted from the image content to the image boundaries. Information overflowed after continuous compression.

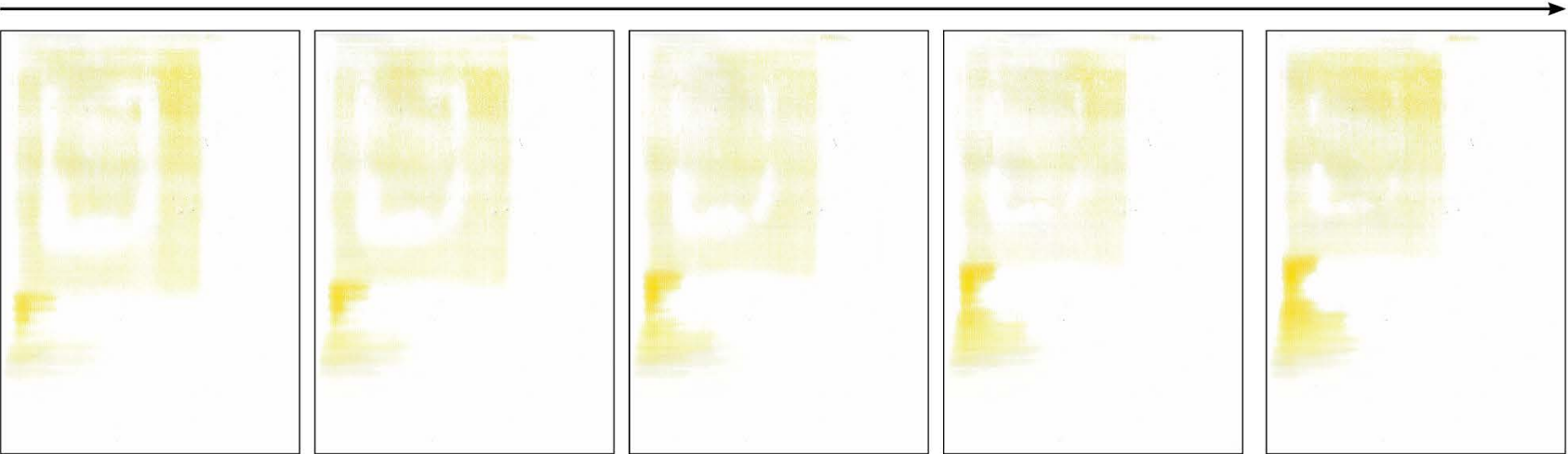
R01 → R05



R06 → R10

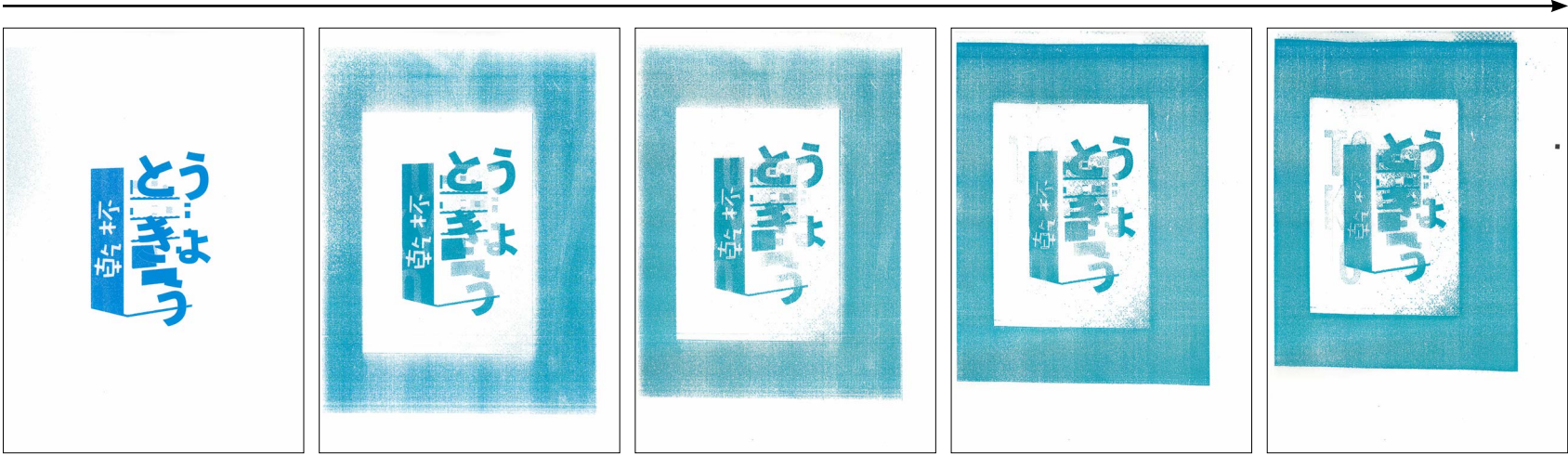


R11 → R15

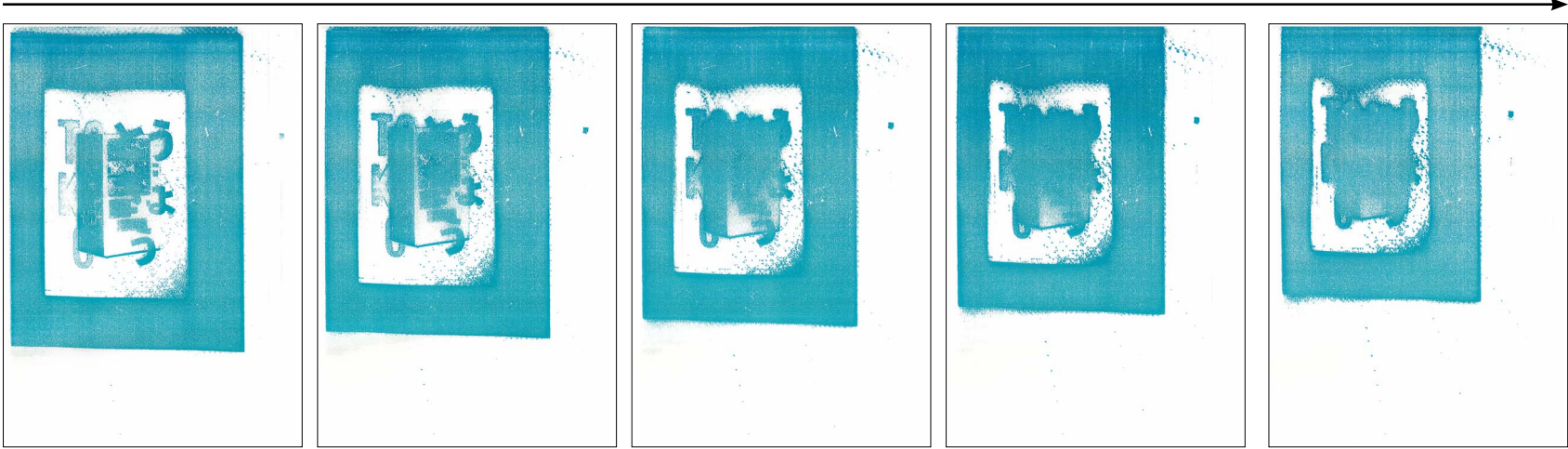


In the Y layer, during continuous iterations, yellow gradually transforms from an invisible, low-density color into a dominant image structure. The reason for this is likely due to color deviation during printing; the printer outputs a color that is not a single color of toner. Simultaneously, yellow can be observed gradually spreading from the center content towards the boundaries, suggesting that other information was captured during scanning, causing the color to expand outwards.

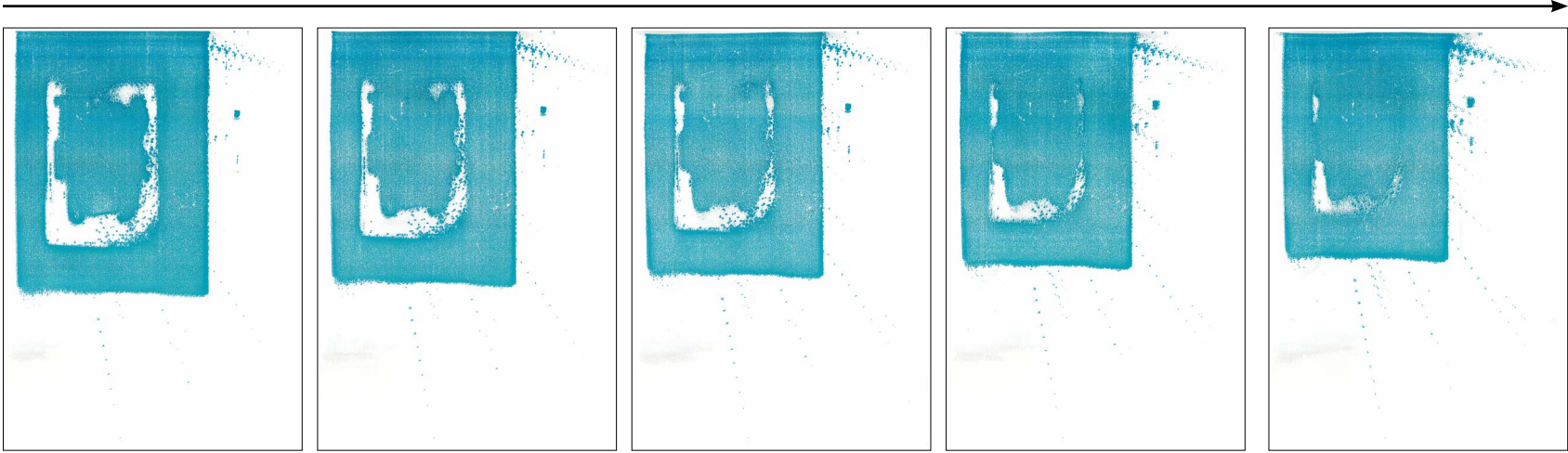
R01 → R05



R06 → R10



R11 → R15



Research Questions

When replicating an image, how is information gradually shifted, lost, or reconstructed?

How does image gradually become distorted?
Is this distortion systematic? How are errors continuously produced instead of being eliminated? At what stage was the error created?
At what stage was it amplified?



Modify LPI to 38 for testing: Test whether the image becomes more stable as the pixel size increases.

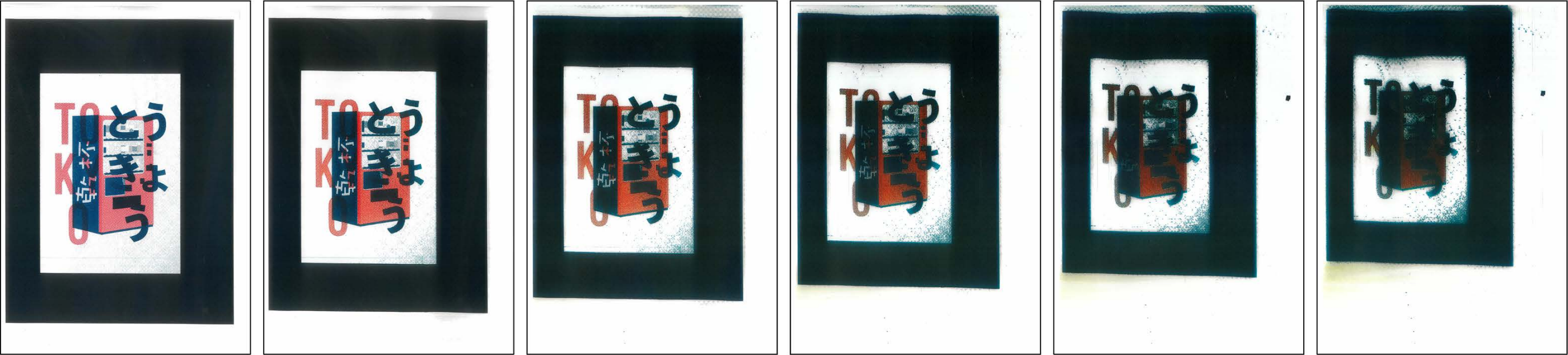
Test with image input sources of varying complexity: Does the image become more stable when using images with very little information? Does the image source affect the direction of change?

Single-color iteration test: Explore at which stage color is transferred from which color to which color.

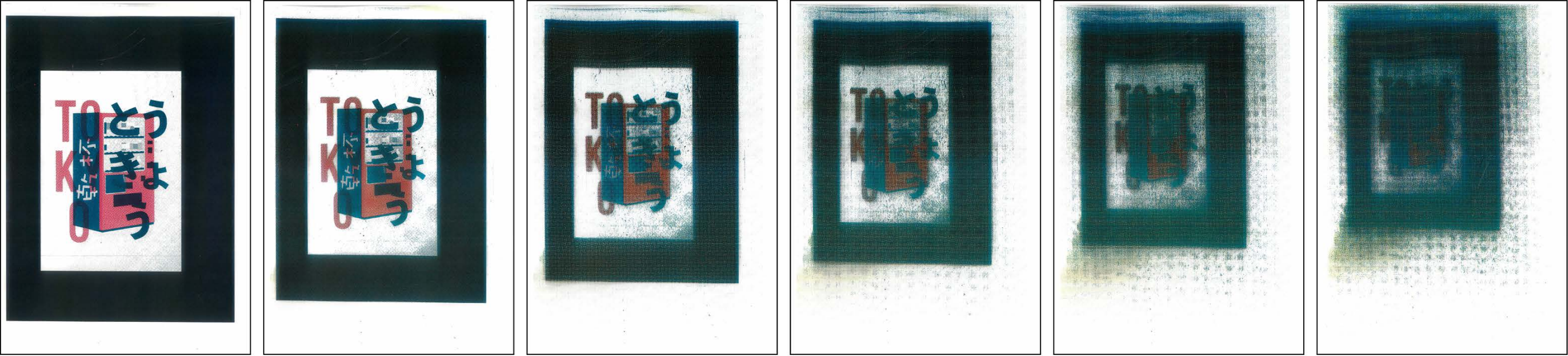
Iteration method with one stage removed: Test which part of the iteration process has the impact.

LPI 85 → LPI 38

week 1 - LPI 85 Round 01 → Round 6



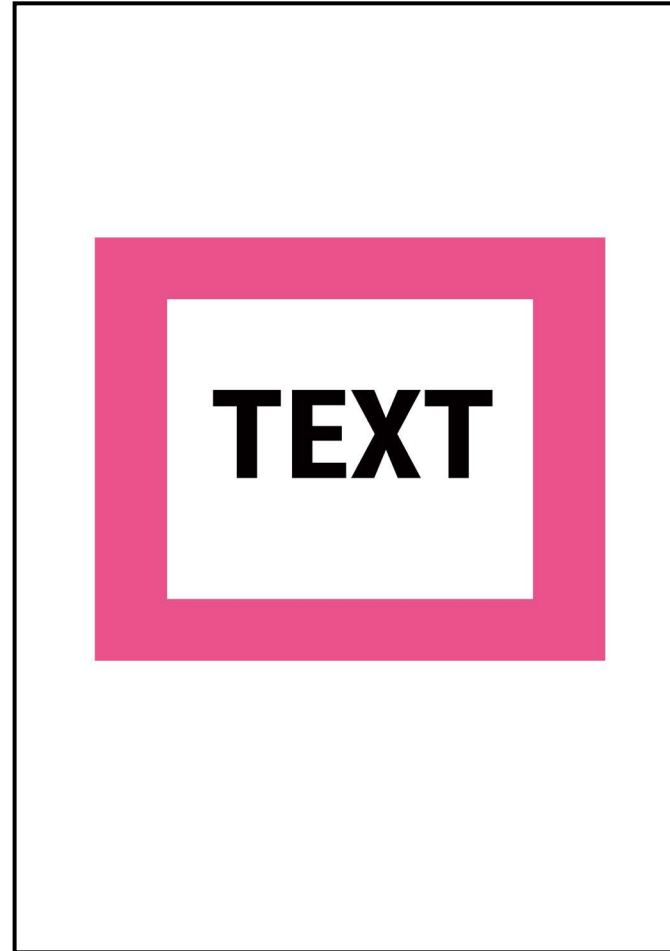
week 2 - LPI 38 Round 01 → Round 6



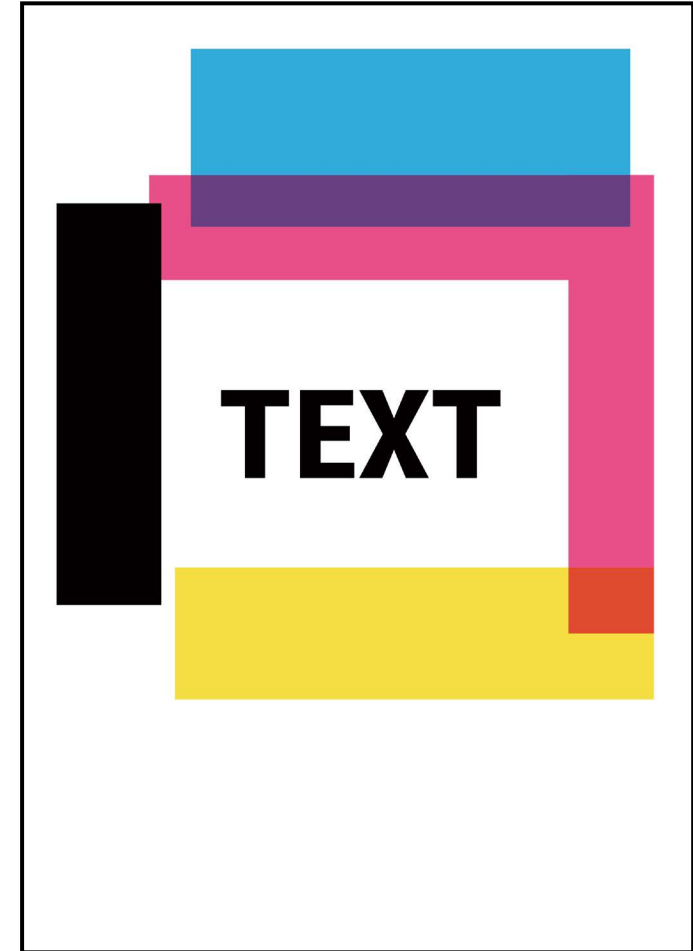
After reducing the LPI (Dots become larger), the text and image structure is simplified more quickly. Each dot becomes information, and each round of color layering reconstructs information over a large area. This leads to the image structure being dispersed more quickly, boundaries disappearing, and colors blending faster. Large dots can be more clearly identified during scanning, but repeated dotting (halftone) can also cause the image to blur rapidly because it cannot retain high-frequency details.

Image complexity

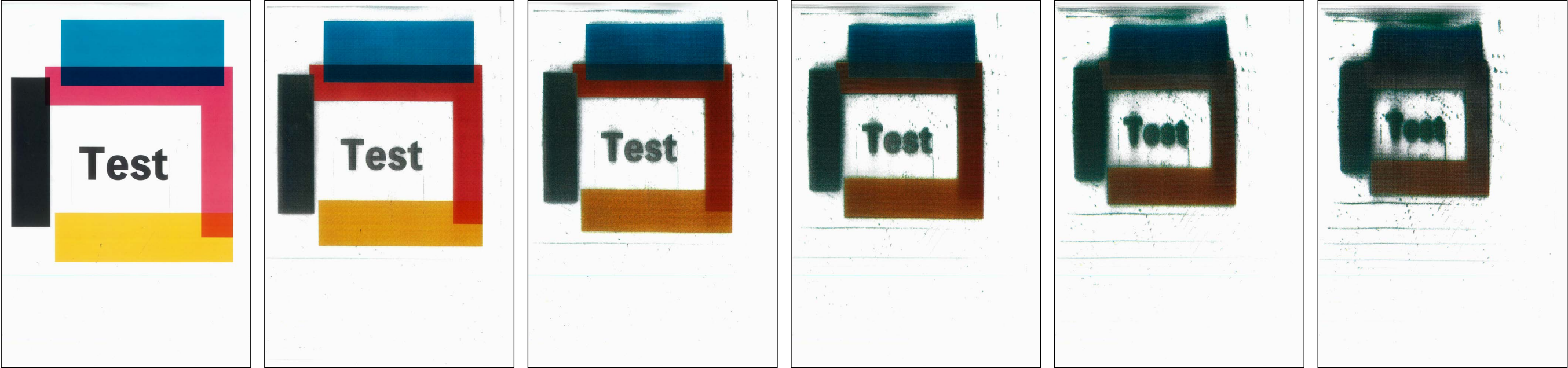
Low complexity LPI 85



Medium complexity LPI 85



LPI 85 Round 01 → Round 6 Medium complexity



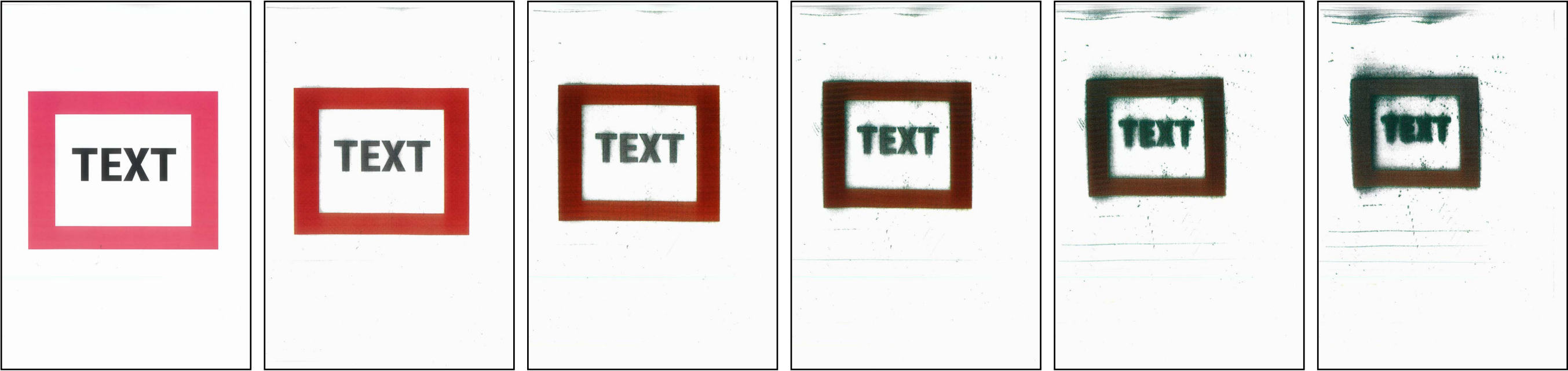
week 1 - LPI 85 Round 01 → Round 6



The complexity of the image has some impact on the speed of image degradation, but it is very small. Even if the black content is small, the color will become very dark in the sixth round. It can be observed that during layering, the system tends to enhance the midtones, causing the overall color to become "dirty". Large areas (boxes) are more stable than details (text).

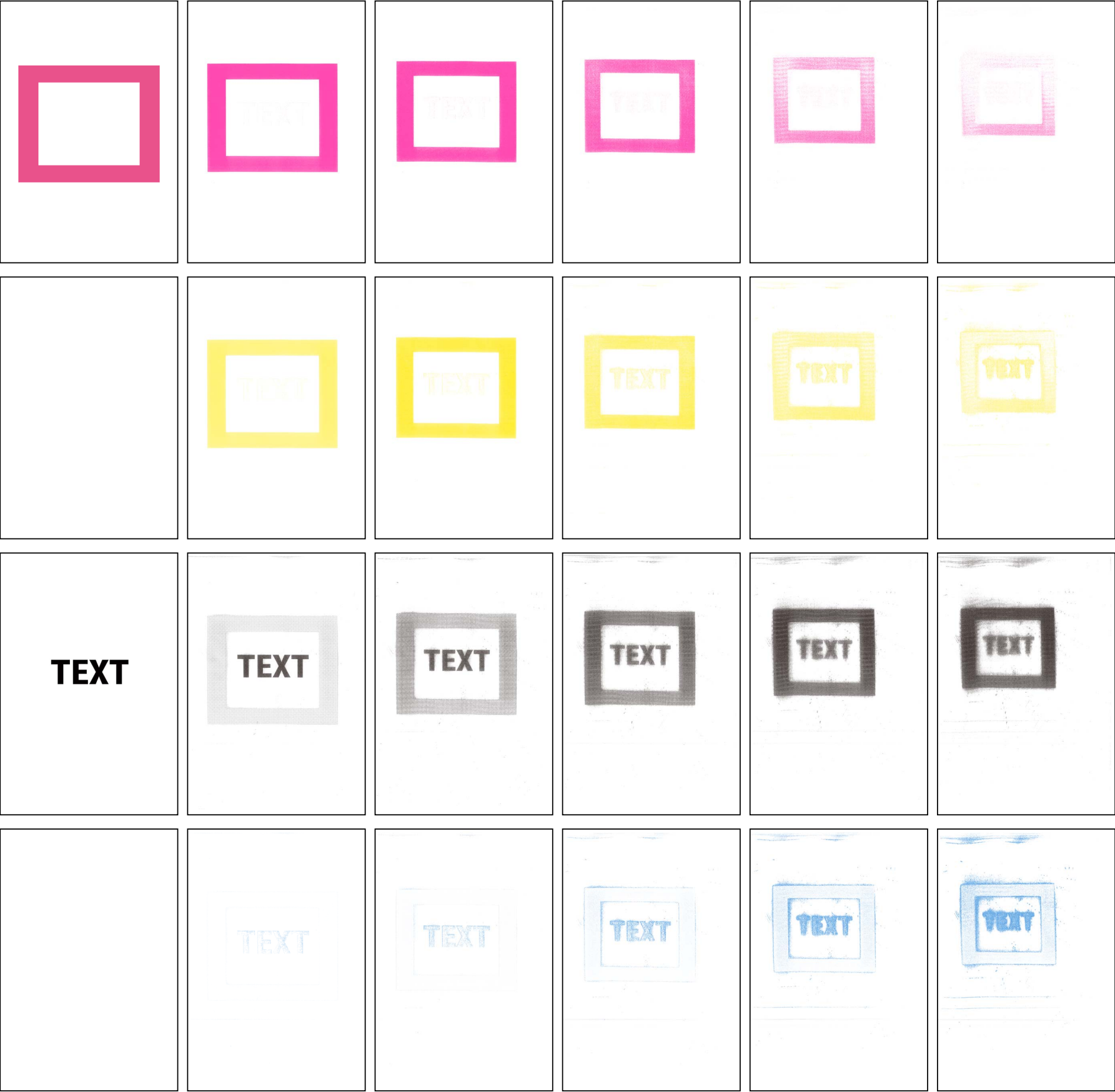
Because of their simple structure, low-resolution images are more likely to maintain their integrity during iteration.

LPI 85 Round 01 → Round 6 Low complexity





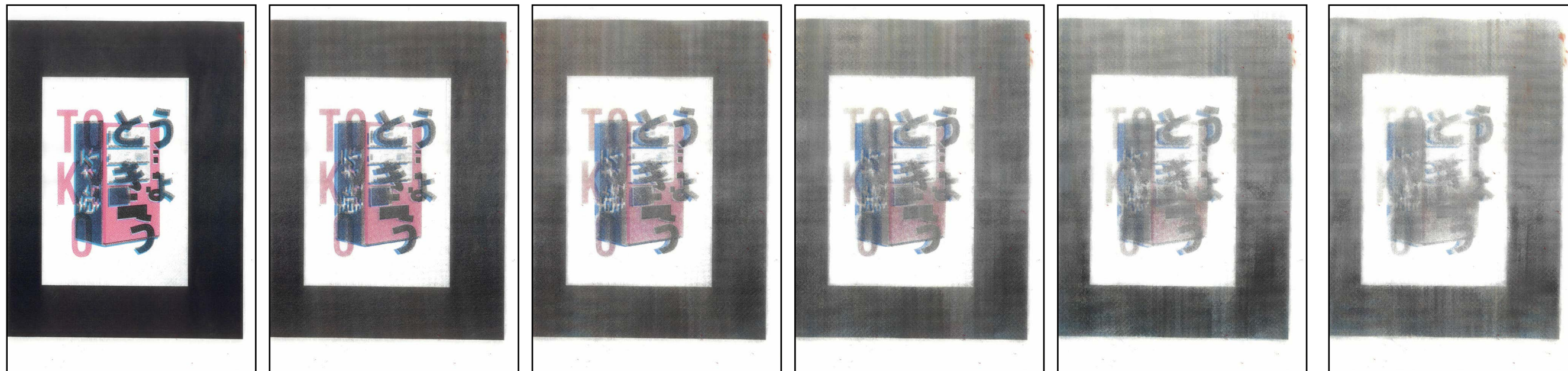
Comparing the color distribution changes of the four layers C, M, Y, and K in the two groups, we can observe that the changes are very similar. The M layer (pink) gradually decreases during the iteration process, the Y layer (yellow) is evenly distributed throughout the image, and the areas of the C and K layers (blue and black) gradually increase with each iteration.



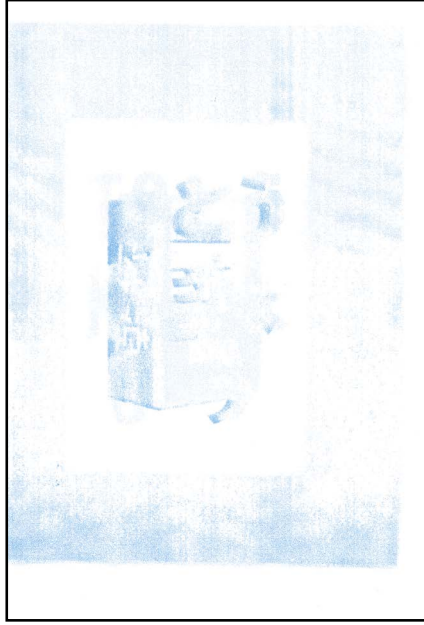
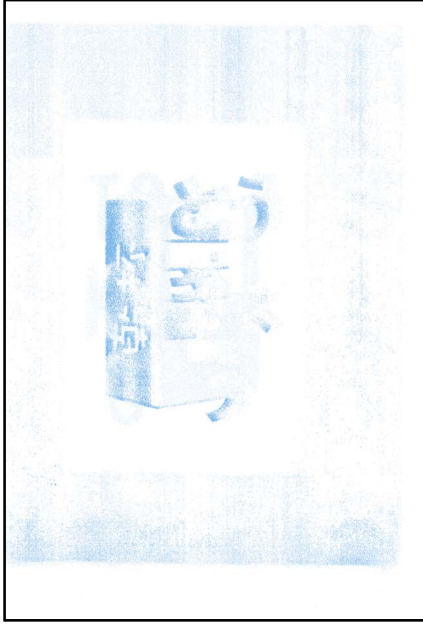
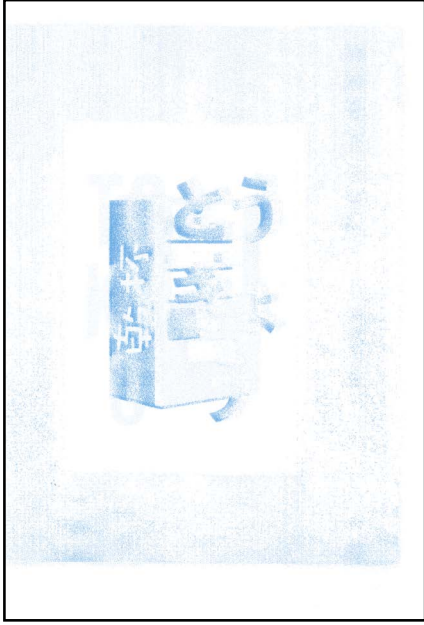
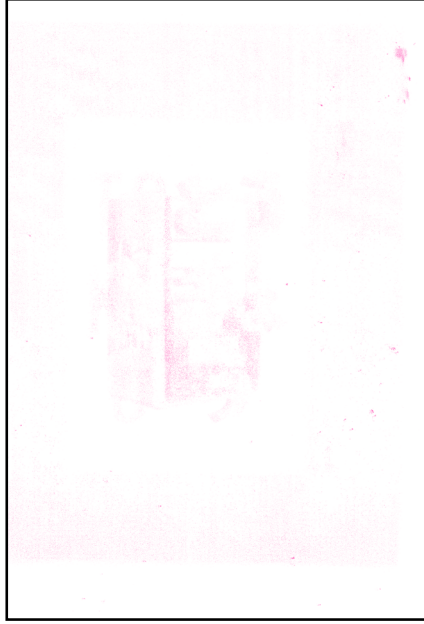
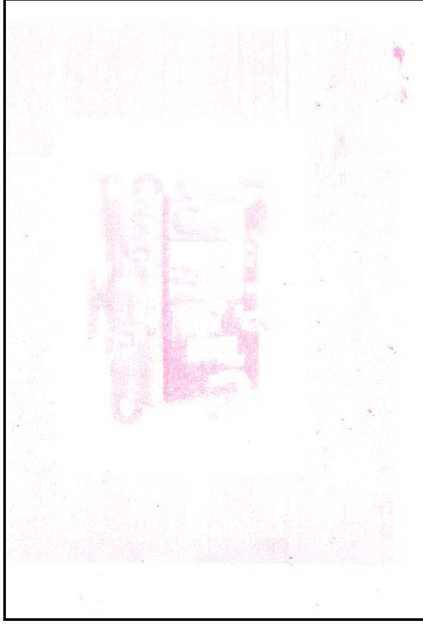
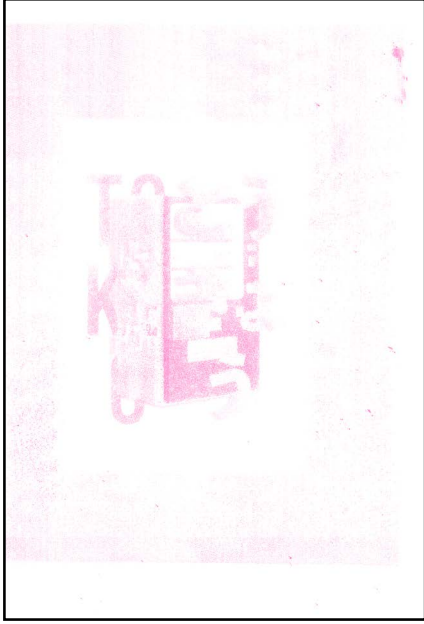
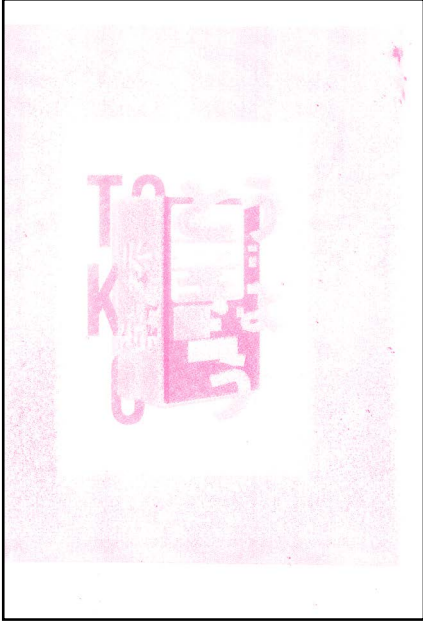
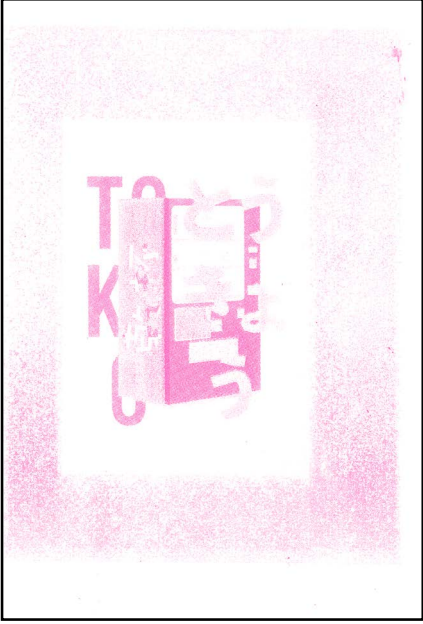
As can be seen in low-complexity images, even incomplete color systems are completed in the loop.

Print - Scan - Layer - Print

The electronic merging process was converted into four physical printing steps.



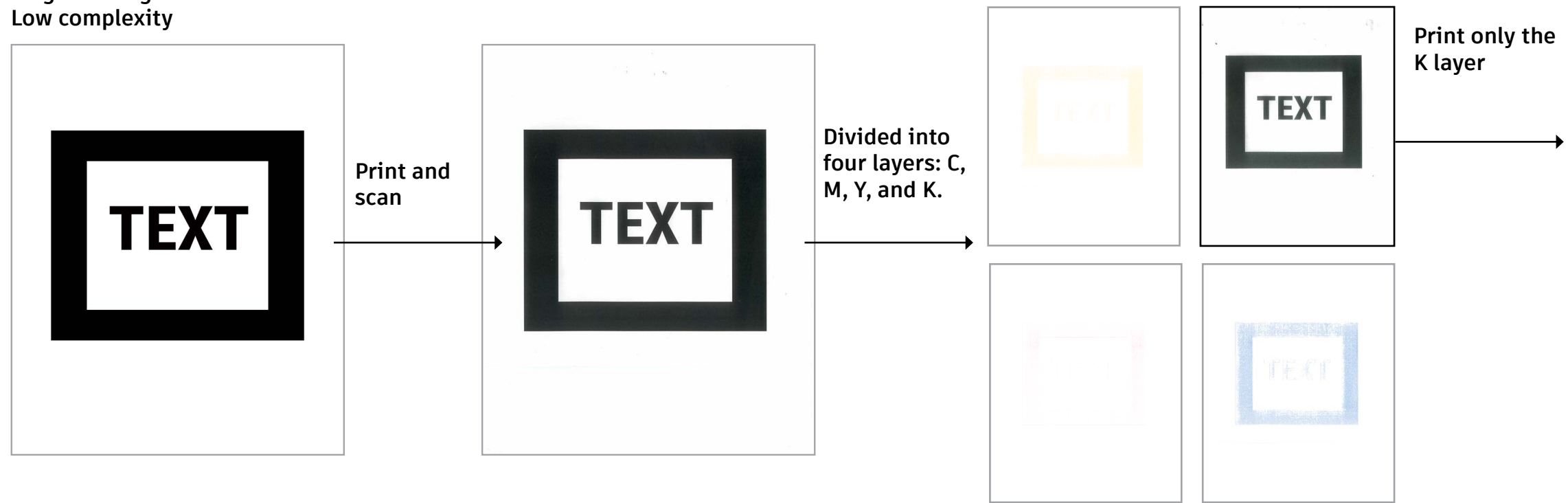
By replacing the digital merging step with four physical prints, it can be observed that the color deepening originates from the layer transitions during digital merging.



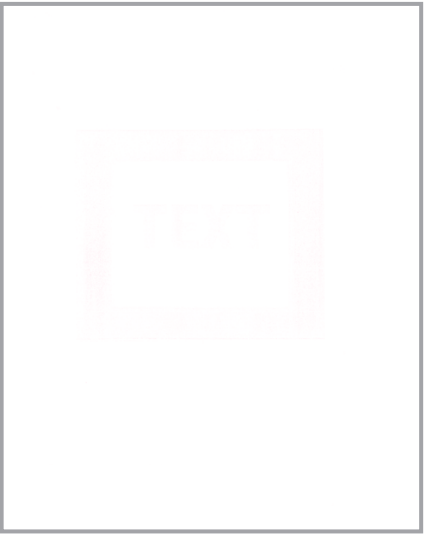
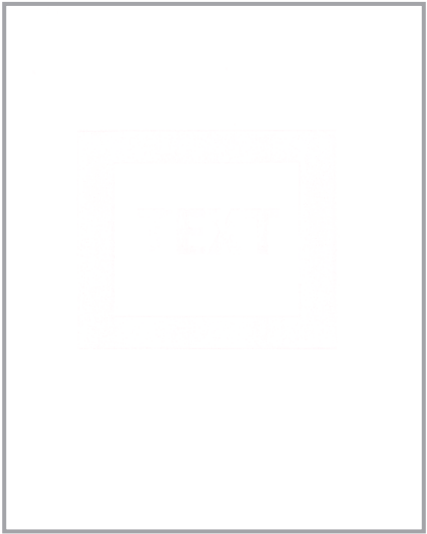
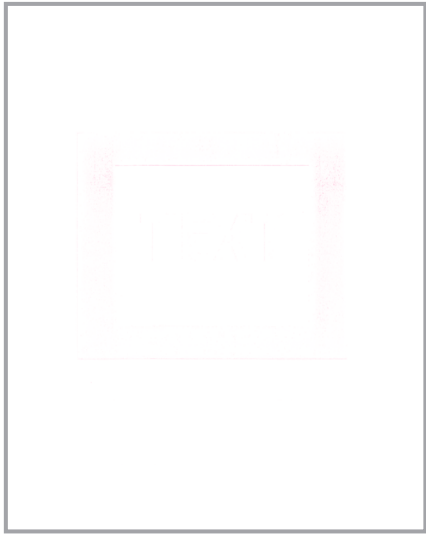
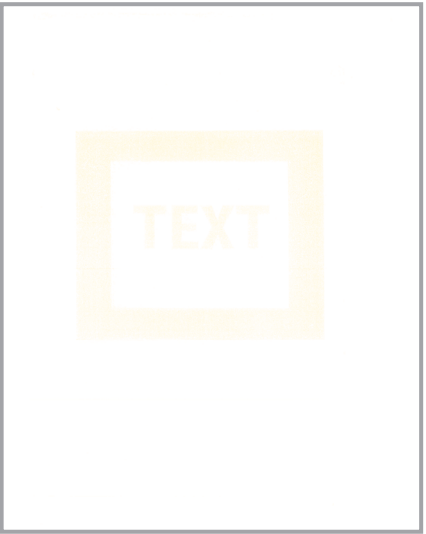
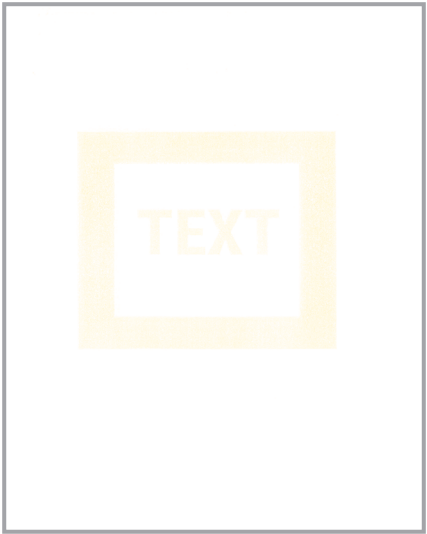
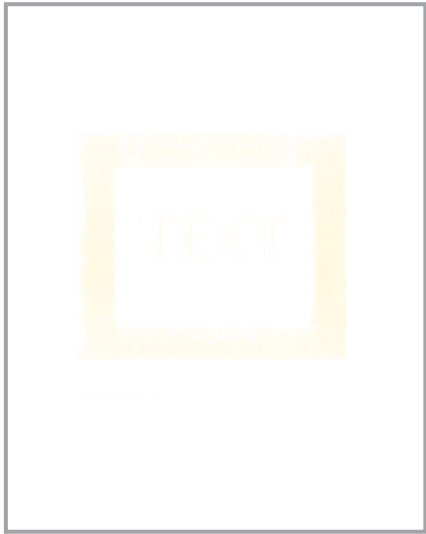
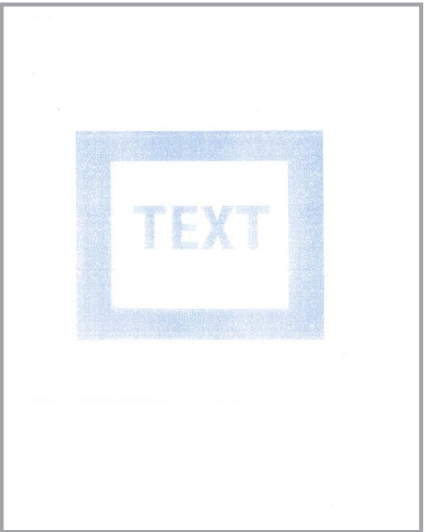
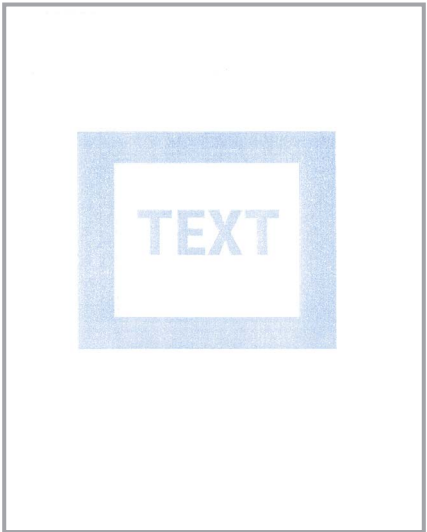
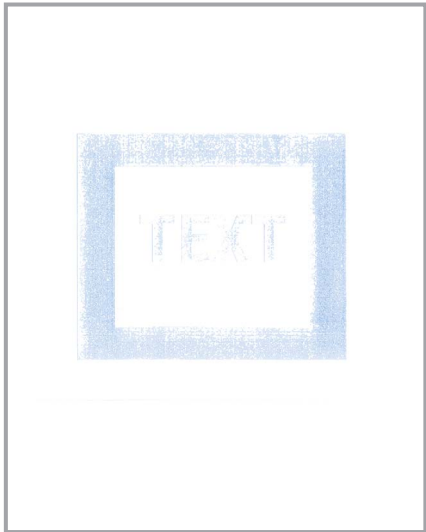
K-layer single-layer printing test

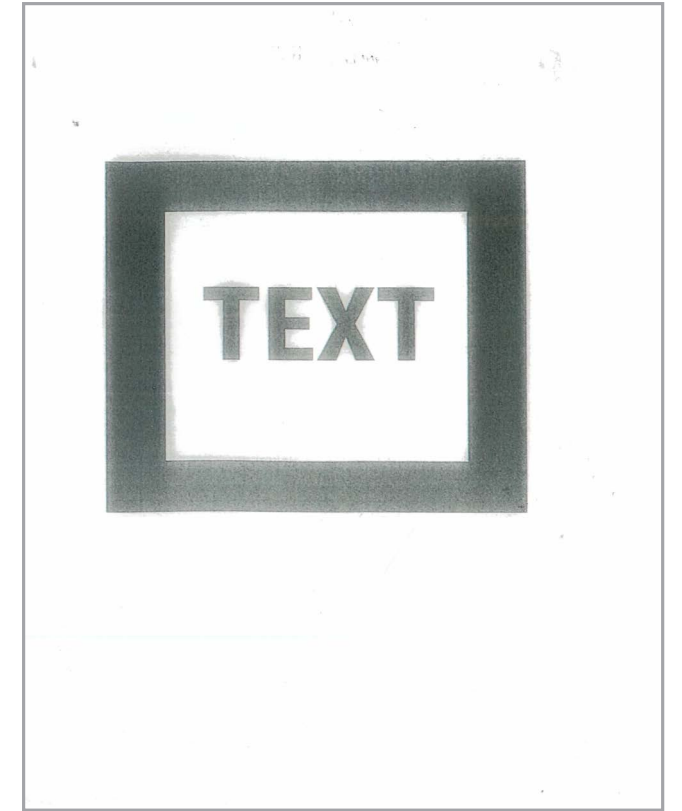
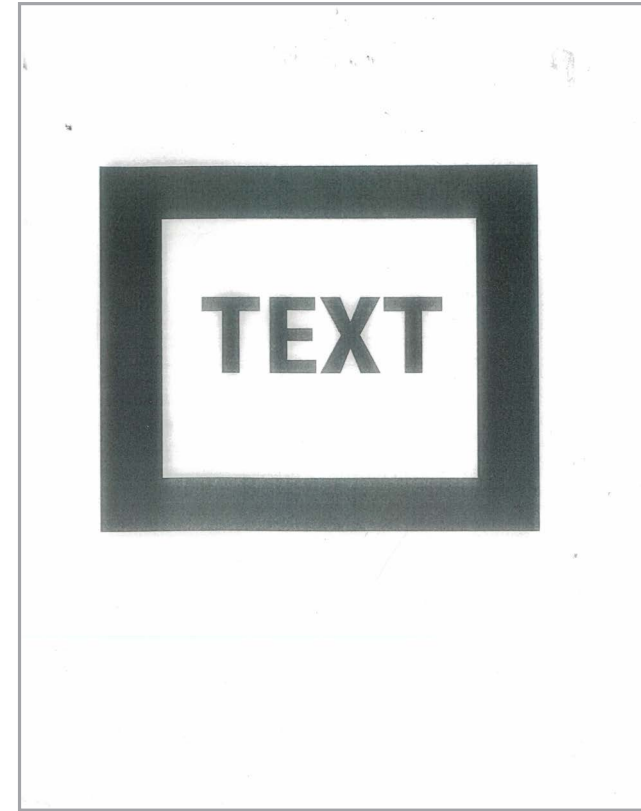
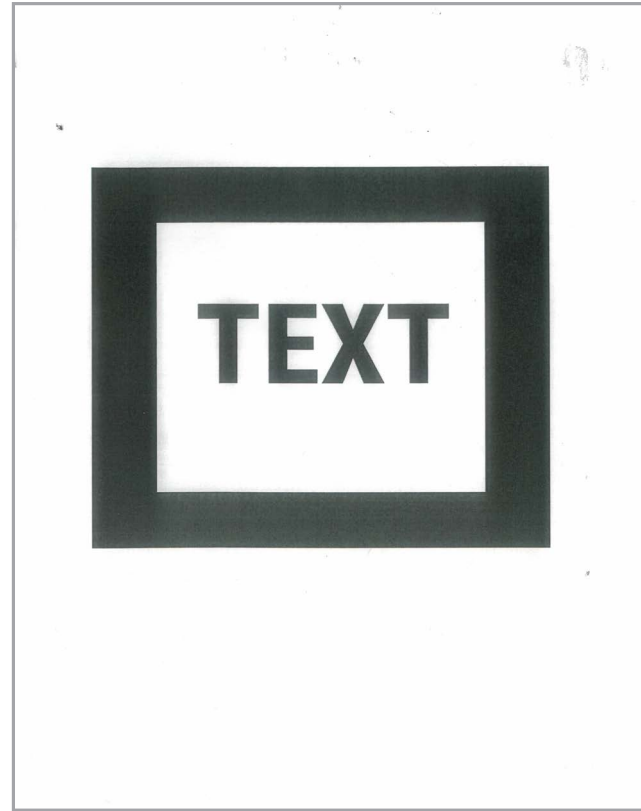
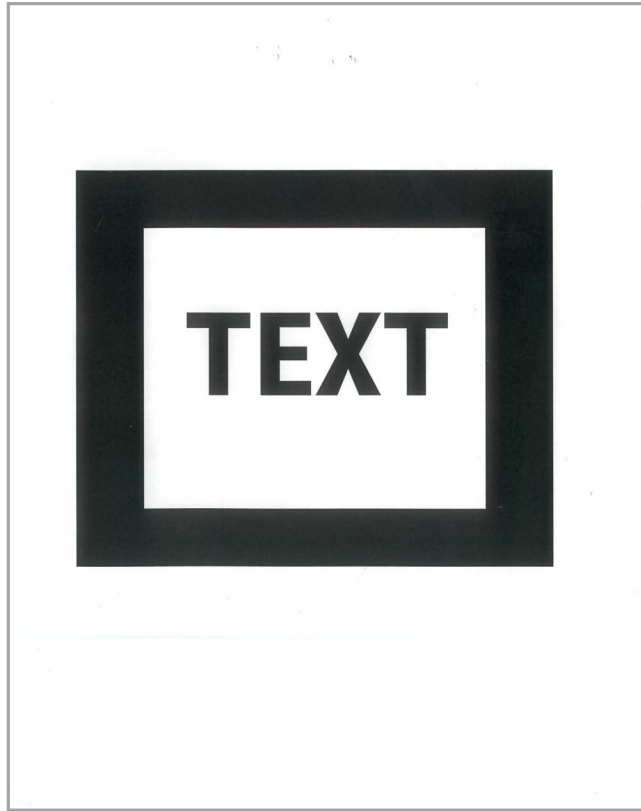
Objective: From the previous observations, we can see that colors inevitably shift towards darker shades during iteration. So, if we initially use only black for printing, how will the colors be allocated? If we only print a single layer (K layers) during iteration, will the image structure remain stable?

Original image used -
Low complexity



In the first round, I observed that black added blue and yellow during layering. I believe this phenomenon is because the light inside the machine during scanning is cool-toned, so color deviation is unavoidable during the digitization of the printed entity. Therefore, during redistribution, the K color is partially allocated to the C and Y layers. Each scan is a color redistribution.





Testing with pure K values revealed that the color overflow at the frame edges occurs during the scanning stage. Or rather, scanning significantly enhances the overflow effect. Toner printing uses heating, positive and negative charges, and rollers; the color on the rollers isn't 100% accurate and overflow-free. Therefore, this phenomenon also occurs in basic printing, but it's too weak for the human eye to detect.



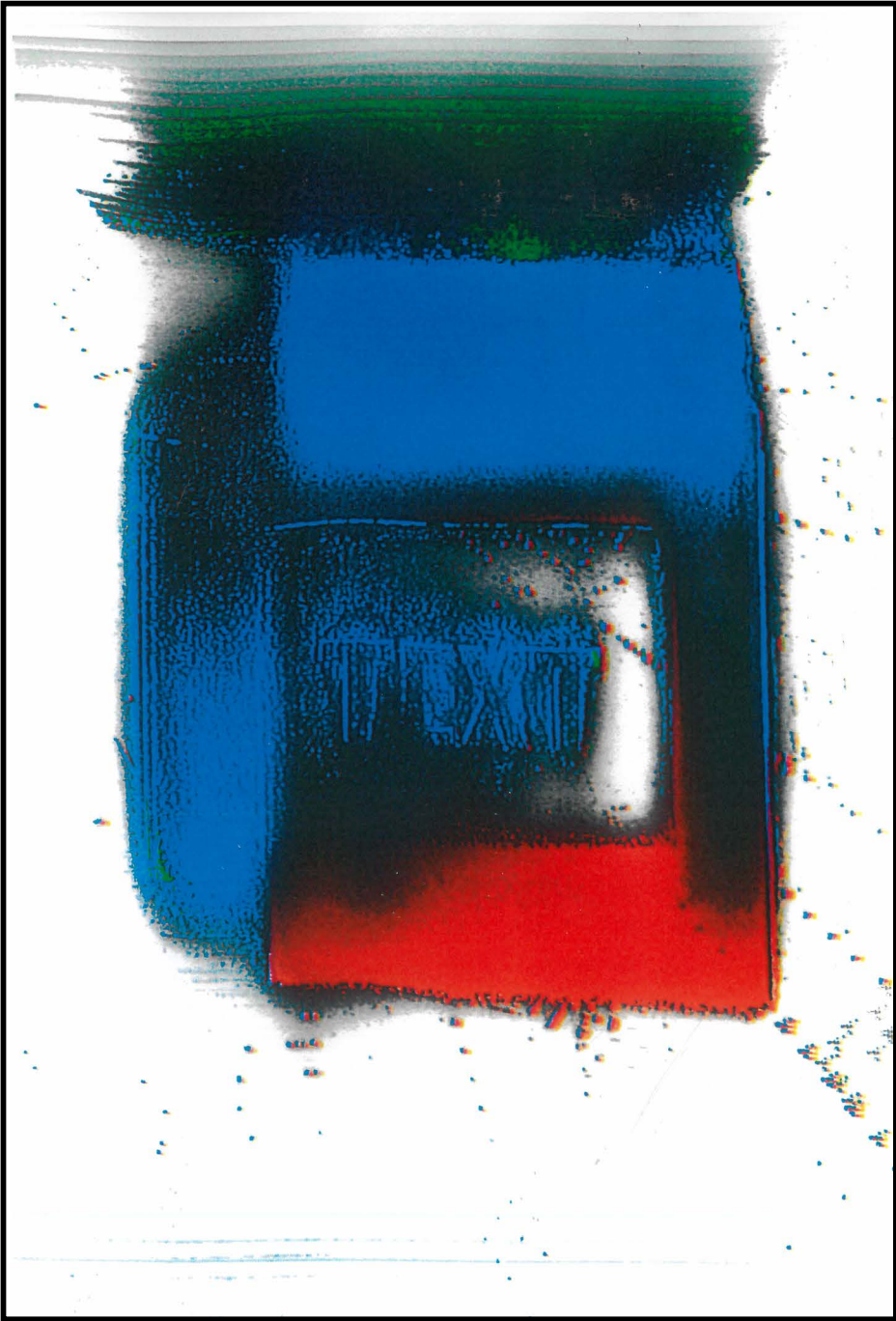
Print - Scan - Print

Remove layering

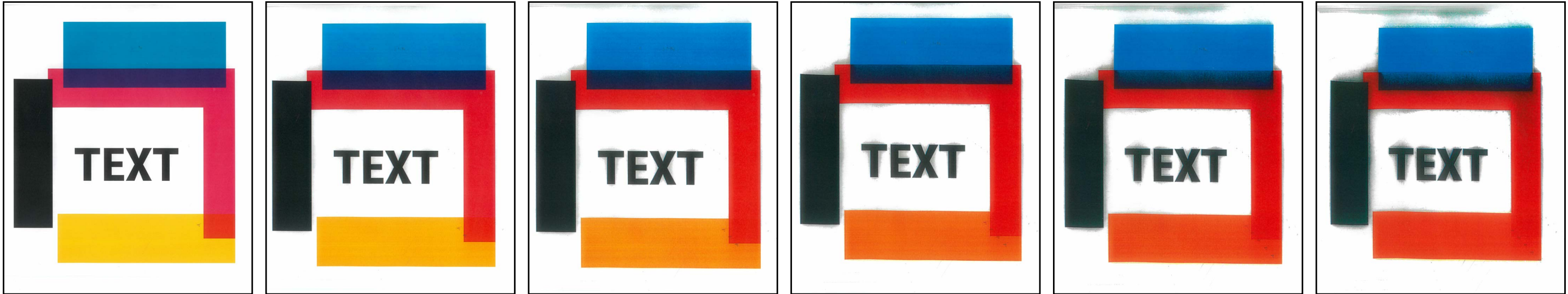


I discovered that the scanning process not only supplemented the cool light but also resampled the toner output from CMYK to RGB. The RGB → CMYK → RGB iterations continued uninterrupted. Yellow and pink eventually stabilized as red. Blue remained stable, continuously diffusing.

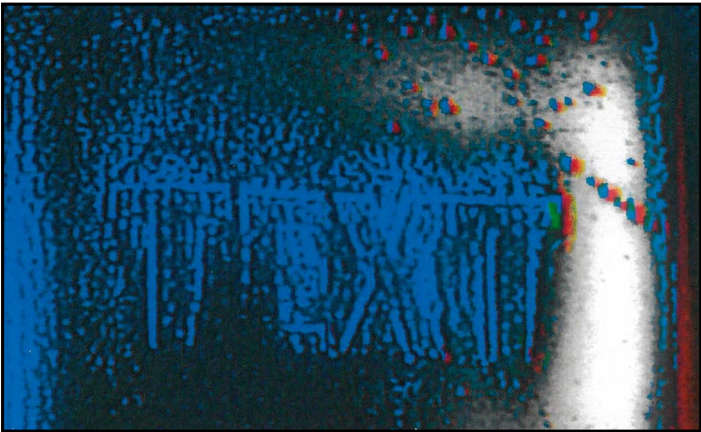
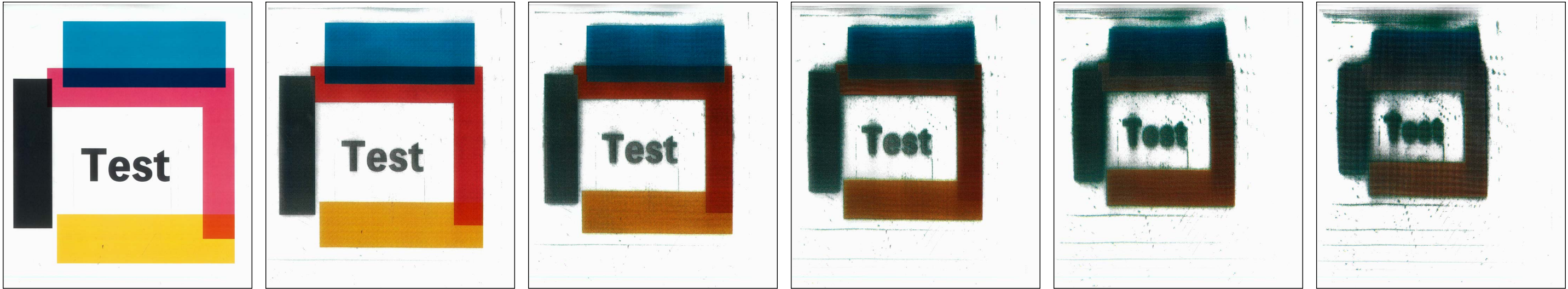
Cyan (C), Magenta (M), Yellow (Y), Black (K) → → → → → → → → → Red (R), Green (G), Blue (B)



LPI 85 Round 01 → Round 6 Print, scan, print, scan



LPI 85 Round 01 → Round 6 Print, scan, layer, print, scan, merge



It can be observed that during the process of removing halftone dots, the error becomes a continuous superposition, and the image continues to be distorted, gradually spreading outward from a single point.