

What are the main causes of poor print quality?



Contents

Introduction	<u>3</u>
Looking back	<u>4</u>
External influences	<u>5</u>
Possible causes	<u>6</u>
Needle in a haystack	<u>7</u>
Conclusion	<u>8</u>
About AutoCoding Systems	<u>9</u>

Introduction

We hear a lot these days from customers wanting to automatically check the print quality of the online printed date/batch codes used on packaging in food, beverage, pharmaceutical and other manufacturing sectors. There seems to be a number of reasons why, including:

- 🌀 The information is important compliance information which has to be readable
- 🌀 My customer fines me if it is missing or not legible
- 🌀 I am increasing automation and have less people around to detect print quality problems

Whatever the motivation, this has become something of a hot topic and there are a range of technologies and products out there claiming to provide an automated solution to this inspection problem. We will be taking a look at some of these technologies, the problems everybody encounters when applying vision technology in this application area and our unique 4Sight technology which takes a very different approach to everything that has gone before.

However, first we are going to take a look at why we have a problem in the first place. Why do these sophisticated machines need verification systems, shouldn't they just work – like technology is supposed to do? To understand the answers to these questions is to understand some of the fundamental challenges that face automated inspection of these codes.

Looking back

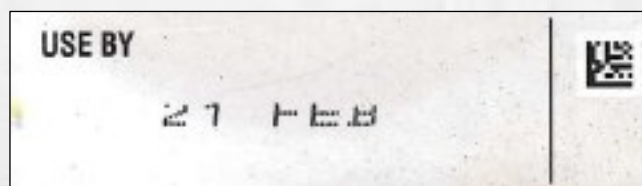
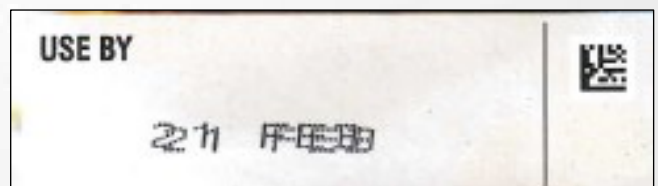
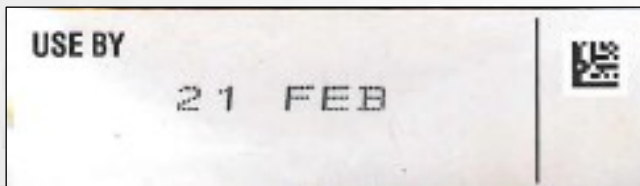
Digital coding machines have been around a while, with the first commercial continuous inkjet printers emerging into mainstream packaging operations probably around the 1970's – the era of flared trousers, T-Rex, no PCs, No Smartphones, No internet, and no email or social media. That was a while ago, and the emphasis of innovation has changed enormously in that period. Back in those heady days we were just frantically inventing stuff, trying to make the technology work well enough to be viable. The benefits of this new technology to meet the needs of new legislation were so great that customers tolerated less than optimum reliability and quality just so they could get a code on a pack and legally sell it.

Fast forward around 50 years and coding innovation has been focussed for some while on maximising reliability and print quality whilst minimising cost of ownership – all standard stuff we see as any technology matures. So with all this effort, why do we still get bad codes on our packaging?



External influences

Well, coding equipment IS very reliable and will print great quality codes for hours/days on end. Occasionally something goes wrong; nobody really knows why, but the coder gets the blame and most of the time, this is not necessarily justified. What has often happened is that the coding machine has, in some way or other, had a close encounter with a human being or the environment that hasn't ended well. The user has done something – or not done something they should have done and as a consequence sooner or later the result is a bad quality or missing code.



Possible causes

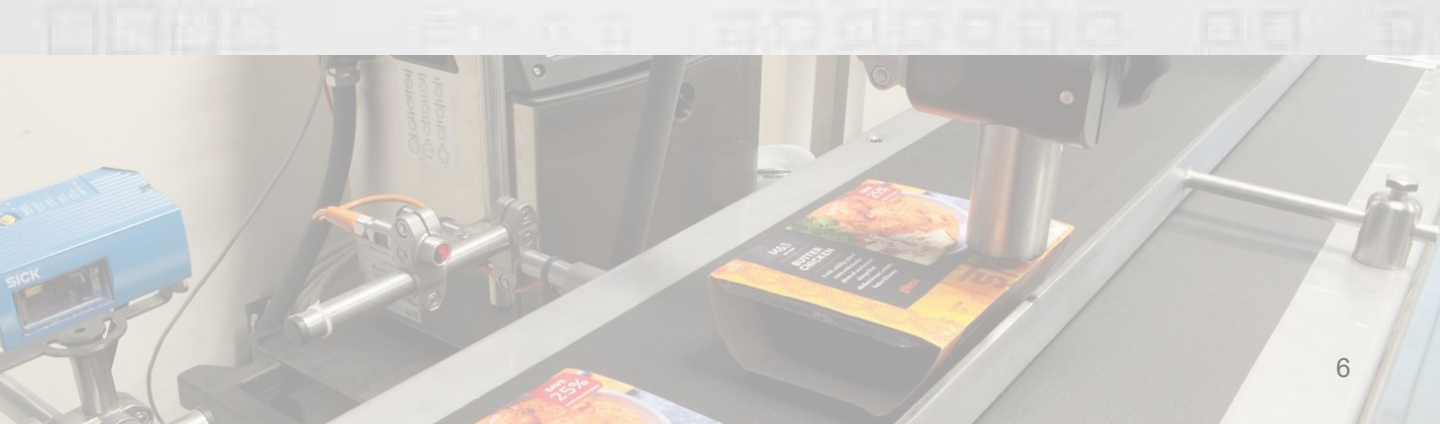
Here are some examples of what we are talking about :

Irregular routine maintenance - Every coder has a process of cleaning or adjustment that is needed to keep it printing well. If this is not done, sooner or later print quality will suffer. When and how is the bit that is hard to predict.

Environmental contamination - Dust, packaging debris or spilled product can all find their way into the wrong place. If that is the print head or print nozzles of the coder, print quality suffers in a pretty random way.

Poor coder setup - This is a bit more systemic, but it happens all the time. The printhead is too far away, too close or at the wrong angle. Perhaps the bracketry isn't great and it's moved due to vibration. If the triggering photocell is in the wrong place or badly adjusted and not detecting the product reliably, this will produce random coding results.

Incorrect line setup - Many coding technologies need the target to move past the printer in a controlled way. Many coding operations are done on product as it travels down a conveyor. Set a guard rail too tight or too loose, interrupt the flow of product in some way and product handling irregularities happen – cue bad quality codes! Of course, we all know that very often things work fine most of the time and occasionally go wrong – in engineering talk, this is a 'borderline condition' which sounds very wise, but still means every now and then you get a bad code.



Needle in a haystack

Maybe you are starting to see a pattern here – it's all a bit random isn't it ? Coding sometimes goes from 100% good to 100% bad, but often it is more like 99.9% good, 0.1% bad as one of these 'borderline conditions' has its moment in the sun.

Charles Bryan, UK National Accounts Manager for Domino says

"We often get calls to instances where code quality has been adversely affected, only to find it's totally unrelated to the performance of the coding equipment itself.

Inconsistent presentation of the product can be overlooked when degraded codes are noticed and something as simple as poorly guided products can create significant volumes of rework or, worse still, waste, in today's FMCG production environment.

All too often periodic quality checks can miss these intermittent types of issue, identifying when the problem started and which products are affected is a waste of valuable resource".



Now logically you might conclude the solution is to simply get people to do all the right things and the problems will pretty much go away. Yes they would, if you could, but is that a realistic expectation in a high pressured manufacturing environment. Probably not. Can QC checks by humans detect the errors perhaps? Have you ever tried looking for a needle in a haystack continuously for an 8 hour shift ?

Conclusion

So, the need for an automated inspection system to identify and prevent bad quality codes from getting to the customer is perhaps now clearer, as will be some of the key parameters of its operation

- 100% inspection is needed because we are often looking for a problem that appears once in a proverbial blue moon
- The faults and consequences are very often random in nature and therefore the effect on print quality is also random and unpredictable
- Nobody wants a quality system that negatively impacts line output by making questionable decisions about whether to stop a line or reject product. It has to make reliably good decisions all of the time

Sounds like a tall order? Yes it is, and technically it is much harder to achieve than you might think. Just like in the 1970's when coding technology challenged the fundamental science of the day, coding print quality verification is challenging the limits of what camera and software technology can do today.

More of that in our next blog –



With over 19 years' experience and a product platform which is the result of over 40 man years' development, AutoCoding Systems is a leading supplier of packaging and coding control systems for the food and drink manufacturing industry. With around 2000 lines under our control in the UK, USA, and Australia, we are proud to name many of the world's largest food companies among our customers.

With a full product installation and aftersales care team, AutoCoding Systems improves efficiency on the packaging line and manages the compliance of coding and packaging on millions of products every single day. We offer a completely vendor-agnostic solution, providing our customers with complete flexibility and freedom of choice to use coding and printing technologies from all the leading suppliers thanks to our extensive driver library and vendor partnerships.

Ready to find out what our solution could do for your manufacturing business? Reach out to us today to request a free demo.

+44 (0)1928 790444

acs.info@jbt.com

www.autocodingsystems.com

**See how AutoCoding Systems can
help your company**

[Request a demo](#)