



Dipartimento Politecnico di Ingegneria e Architettura (DPIA)
Corso di Visione Artificiale

Progetto: 3d Printer Defects Classification

Studente: Lorenzo Zaccomer

Professore: Andrea Fusiello

Date: April 22, 2022

Estratto

Progetto svolto per il corso di Visione Artificiale tenuto dal Professore Andrea Fusiello nell'anno accademico 2021/2022.

Lo scopo di questo progetto è quello di generare un modello in linguaggio Python affinché possa determinare la difettosità o meno di un pezzo stampato mediante una stampante 3d da un'immagine passata come parametro.

Indice

1	Introduzione	1
1.1	La stampante 3d	1
1.2	Descrizione del problema	1
1.3	Nascita dell'idea per questo progetto	2
2	Descrizione del progetto	2
2.1	Introduzione	2
2.2	Introduzione	3
3	Altro	3
3.1	Tables	3
3.2	Figures	3
3.3	Equations	4
3.4	Listings	5
4	Ringraziamenti	6
5	Conclusioni	6

1 Introduzione

1.1 La stampante 3d

La stampante 3D è un dispositivo attorno al quale vi è un grosso interesse al giorno d'oggi, questo perché è impiegabile in molti campi applicativi, sia in ambito industriale, sia in ambito *user level*, questo perché permette un grosso risparmio in termini di costi, in aggiunta alla facilità di manutenzione che essa offre.

Con una spesa iniziale non troppo elevate per la stampante in sé (200€ per un modello base), permette a chi si interfaccia per la prima volta a questo *mondo*, di *prodursi* i pezzi desiderati, inoltre il suo utilizzo è facilitato dalla presenza di svariati tool di disegno grafico sul mercato, come per esempio *OpenSCAD*, il quale è open-source e scaricabile gratuitamente, e dalla comunità di *Thingiverse*, dove gli utenti possono condividere i propri progetti, e con *UltimakerCura*, è possibile modificare le impostazioni di stampa.

Si pensi che, ad esclusione ovviamente delle componenti elettroniche, come i servo motori, di crearsi una propria stampante 3d, stampando i pezzi impiegando un'altra stampante 3d.

Il materiale maggiormente impiegato per le stampe è il PLA, la quale è una plastica.

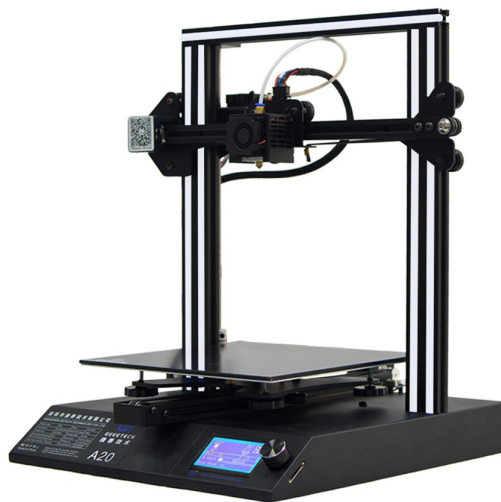


Figure 1.1: Esempio di stampante 3d, modello Geeetech A20

1.2 Descrizione del problema

Essa però presenta una serie di problemi durante l'utilizzo, il più tangibile per esempio è il rumore emesso dall'elettronica e dalle componenti meccaniche, per cui è necessario posizionarla in un'altra stanza il più lontano possibile, anche nel caso di una box dedicata, così facendo il problema si attenua ma non scompare, questo rende impossibile un controllo *a vista* dell'esecuzione di stampa, per cui in caso di un problema di stampa, si rischia di far proseguire la stampa per ore, con uno spreco di materiale plastico e di

tempo.

Inoltre un parametro che può rendere difettoso il pezzo è la **temperatura**, in particolare di più elementi:

- Della *stanza* stessa cui si trova la stampante
- Del *piano di adesione* della stampa
- Dell'*ugello* (nozzle) della stampante

Questo perché se la temperatura di questi 3 elementi non è adeguata, si rischia una ritrazione del pezzo, con conseguente difettosità della stampa.

Nel primo caso è maggiormente risolvibile, questo riscaldando la stanza con un classico scaldabagno, o al più impiegare un contenitore chiuso dove posizionare la stampante stessa, in modo da generare un *microclima*.

Nel secondo caso l'ugello deve lavorare alla temperatura di fusione del materiale in uso, ovvero intorno ai 200° per il PLA, e questo parametro è settabile dalla stampante stessa mediante lo schermo, se la temperatura è troppo bassa, il filamento non si scalda e la stampa non viene eseguita, se la temperatura è troppo alta, il filamento rischia di fare quello che si chiama in gergo *effetto spaghetti*, CANTIERE

Per quanto riguarda il piano di adesione, le temperature operative si aggirano dai 30° agli 80°, però se esse non sono adeguate si rischia che in un caso la base del pezzo si solidifichi troppo, facendo sì che il pezzo si *stacchi completamente* dal piano, o in effetto minore il sollevamento di una sua parte, in questo caso è possibile, mediante il programma di stampa, preimpostare un layer aggiuntivo attorno al pezzo durante le fasi iniziali, il quale andrà a "subire" eventualmente la ritrazione, senza coinvolgere il pezzo stesso. Infine se le temperature del piano sono troppo alte il pezzo non si raffredda nella maniera giusta, rischiando di produrre delle pieghe sulle superficie esterne del pezzo.

Perciò l'addetto alla stampa, deve tenere conto di una moltitudine di possibili eventi, soprattutto nel caso in cui un pezzo richieda delle ore per essere generato.

1.3 Nascita dell'idea per questo progetto

L'idea nasce quando è stata regalata una stampante 3d (vedi figura 1.1) e ho iniziato ad utilizzarla per i primi esperimenti, per cui le problematiche descritte precedentemente le ho affrontate in prima persona, ho pensato per mesi a come minimizzare il problema, e CANTIERE

2 Descrizione del progetto

2.1 Introduzione

In questa sezione si andrà a descrivere in maniera "generale" il codice, andando ad osservare le parti salienti che lo caratterizzano.

2.2 Introduzione

3 Altro

3.1 Tables

Tables can be built and labeled like this, adding a new row requires adding new line of entries, separated by `&` and ends with `\\`, while adding new column requires adding 1 in the `\begin{tabular}{l1111}` part.

Table 1: Table Caption

Title 1	Title 2	Title 3	Title 4
Entry 1	Entry 2	Entry 3	Entry 4
Entry 5	Entry 6	Entry 7	Entry 8

3.2 Figures

you can add figures with captions, also you may label them and reference them again with `\ref{fig:fig1}`.



Figure 3.1: Birzeit Logo (PNG)



Figure 3.2: Birzeit Logo (SVG)

Or you can generate figures using TikZ, I've provided some styles to generate block diagrams as in Figure 3.3, but I won't be able to explain TikZ syntax for you so please try figure it out by yourself (it isn't hard to understand). Also you can add subfigures using `subfigure` environment ¹.

¹Environment means a block surrounded by `\begin {envname}` and `\end {envname}`

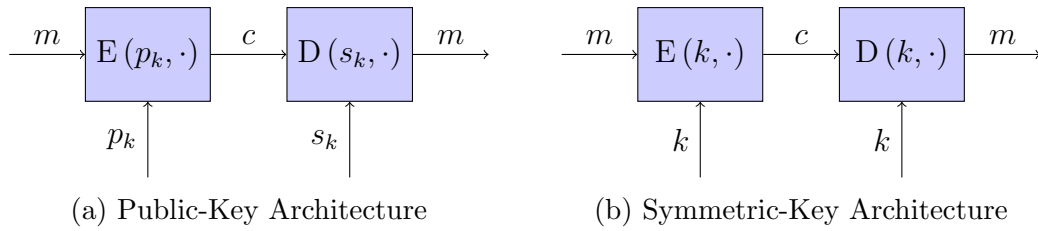


Figure 3.3: Public-Key vs. Symmetric-Key: Architecture

3.3 Equations

you can use `\begin{align} \end{align}` to insert aligned equations like this

$$\tau = 28 \text{ s} \tag{1}$$

$$\omega = 2\pi \frac{2}{60} = 0.209 \text{ rad/s} \tag{2}$$

$$\begin{aligned} lag &= \frac{\theta(\omega)}{\omega} = \frac{\arctan(\omega\tau)}{\omega} \\ &= \frac{\arctan(0.209 \cdot 28)}{0.209} \\ &= 6.69 \text{ s} \end{aligned} \tag{3}$$

or you can insert `\notag` at the end of each line to prevent it from being enumerated.

$$\begin{aligned} \tau &= 28 \text{ s} \\ \omega &= 2\pi \frac{2}{60} = 0.209 \text{ rad/s} \\ lag &= \frac{\theta(\omega)}{\omega} = \frac{\arctan(\omega\tau)}{\omega} \\ &= \frac{\arctan(0.209 \cdot 28)}{0.209} \\ &= 6.69 \text{ s} \end{aligned}$$

And you can insert one line equation with `\begin{equation} \end{equation}`

$$A = 15 \cdot B \tag{4}$$

and `\begin{equation*} \end{equation*}` to prevent the equation from being enumerated.

$$A = 15 \cdot B$$

And inline math if you want with `$Expression$` like this $A = 15$.

3.4 Listings

Listing 3.1 shows a sample listing for C++ code.

```
#include <iostream>
#include <algorithm>

using namespace std;

int main() {
    int n; cin >> n;
    int a[n];
    for (int i = 0; i < n; ++i)
        cin >> a[i];
    sort(a, a + n);
    for (int i = 0; i < n; ++i)
        cout << a[i];
    return 0;
}
```

Listing 3.1: Sample Code

Listing 3.2 shows a sample listing for shell session.

```
$ g++ main.cpp

$ ./a.out
4
3 2 4 1
1 2 3 4
```

Listing 3.2: Sample Output

4 Ringraziamenti

Feel free to use this template for your reports.

5 Conclusioni

Feel free to use this template for your reports.

References

List of Figures

1.1	Esempio di stampante 3d, modello Geeetech A20	1
3.1	Birzeit Logo (PNG)	3
3.2	Birzeit Logo (SVG)	3
3.3	Public-Key vs. Symmetric-Key: Architecture	4

List of Tables

1	Table Caption	3
---	-------------------------	---

List of Listings

3.1	Sample Code	5
3.2	Sample Output	5